SIBYLLE: A system for alternative communication adapting to the context and its user

Tonio Wandmacher
Université François Rabelais
IUP Blois, 3 place J. Jaurès
F-41000 Blois, France
(+33) 2 54 55 21 11
tonio.wandmacher@univ-tours.fr

Jean-Yves Antoine
Université François Rabelais
IUP Blois, 3 place J. Jaurès
F-41000 Blois, France
(+33) 2 54 55 21 11
jean-yves.antoine@univ-tours.fr

Franck Poirier
Université Européenne de Bretagne
UBS, CERYC, Campus de Tohannic
F-56000 Vannes, France
(+33) 02 97 01 72 29
franck.poirier@univ-ubs.fr

ABSTRACT
In this paper, we describe the latest version of *SIBYLLE*, an AAC system that permits persons suffering from severe physical disabilities to enter text with any computer application and also to compose messages to be read out by a speech synthesis module. The system consists of a virtual keyboard comprising a set of keypads which allow entering characters or full words by a single-switch selection process. It also comprises a sophisticated word prediction component which dynamically calculates the most appropriate words for a given context. This component is auto-adaptive, i.e. it learns on every text the user has entered. It thus adapts its predictions to the user's language and the current topic of communication as well. So far the system works for French, German and English. Earlier versions of *SIBYLLE* have been used since 2001 in the Kerpape[1] rehabilitation center (Brittany, France).

Categories and Subject Descriptors
J.3.3 [**Compute applications**]: Life and medical sciences

General Terms
Human Factors, Experimentation, Performance.

Keywords
Augmentative and Alternative Communication ; Virtual keyboard; Word prediction ; Latent Semantic Analysis ; User adaptation ; Keystroke saving rate ;

1. INTRODUCTION
This paper will present *SIBYLLE*, an AAC (Augmentative and Alternative Communication) system for persons with severe speech and motion impairments (cerebrally and physically handicapped persons, locked-in syndrome etc.). Like other AAC systems, such as FASTY [1] or *Dasher* [2], *SIBYLLE* aims at restoring communicative abilities. Currently, *SIBYLLE* is available for French, German and English.

SIBYLLE is composed of four modules. The first one is the physical input interface (e.g. an eye glimpse or a breath sensor) replacing standard computer devices, which are unsuited for these users. Secondly, the on-screen virtual keyboard that replaces the physical keyboard and allows the user to select textual items (letters or words) in order to compose messages: a selection frame successively highlights each of the items, which can then be selected by the user. Finally, the last two components are a text editor and a text-to-speech synthesis (TTS) module to read out the typed message.

Since the user has to wait until the selection frame highlights the desired item on the keyboard, one of the main difficulties of such a system results from the extreme slowness of message composition. We thus investigate two complementary approaches to speed up text input: fast key selection and keystroke reduction. These improvements are based on two prediction devices, taking the already typed part of the message (left context) into account: the system presents the most probable words on screen and the letters on the virtual keyboard are ordered according to their probability of occurrence. This prediction is achieved by means of a statistical language model which is automatically adapted to the user and the current semantic context of communication.

2. SIBYLLE: THE USER INTERFACE
Since it concerns above all people suffering from severe impairments, *SIBYLLE* is designed for single switch input devices. The virtual keyboard combines a set of sub-keypads offering to insert letters, numbers, words and also predefined sentences for "*emergency*" uses (e. g. "*I am thirsty, I want to drink*"). Jump keys provide fast moves between these sub-keypads: they are usually the first keys on each keypad. Figure 1 presents the latest version of the user interface. One can distinguish the main sub-keypads of the virtual keyboard:

- *Letter keypad*: Is used to compose messages letter after letter. Its organization is dynamic when necessary (see § 3.).
- *Prediction list*: When the user selects one of these predicted words, the system automatically completes the message, thereby avoiding time consuming letter selections.
- *Function keypad*: In former versions, *SIBYLLE* only comprised an integrated text editor. Since users of our system wanted to write not only simple text documents but also compose e-mails, use a real word processor or a search engine on the web, we decided to make *SIBYLLE* more flexible.

[1] http://www.kerpape.mutualite56.fr/

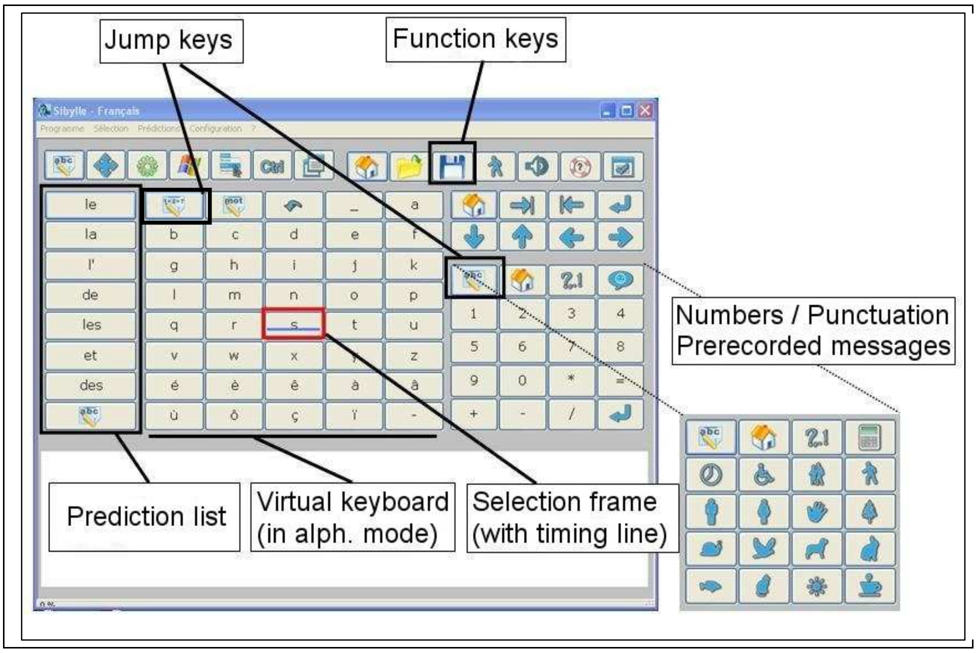

Figure 1. The user interface of *SIBYLLE* (French version 3.1)

By interfacing the *Microsoft Windows* API, our system is now able to enter text to any kind of *Windows* application. Furthermore, configurable function keys enable direct actions such as *Save As*, *Open* or *TTS synthesis*

Figure 2. The different aspects of the numpad according to the selected mode (left: number selection, middle: punctuation, right: pre-recorded messages)

- *Numeric keypad (numpad)*: This sub-keypad can be used in several modes: one can use it to select numbers - but also punctuation marks, and finally to compose rapidly predefined sentences. These pre-recorded messages can be defined by the user. The display of this sub key-pad changes according to the selected mode (figure 2).

The user interface is highly configurable. For instance, the user can choose between three selection modes:

- mouse selection (for users who are still able to control such a device)
- line/column scan
- linear scan

In a scanning selection mode, the users are often disturbed by the "jerky" shifts of the selection frame from one item to the following one: when the cursor approaches the desired key, they meet difficulties to temporally prepare their action. As a result, we observed a significant rate of erroneous selections with the previous versions of *SIBYLLE*. For this reason we added a timing line which glides gently from the top to the bottom of the frame (figure 3) when it stops on each item. This dynamic feedback is very helpful for the users, which can estimate in this way the time remaining until the next shift of the selection frame occurs.

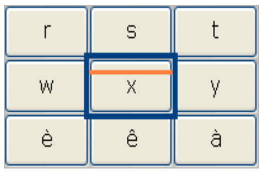

Figure 3. Selection frame with the timing line

The user can modify the scanning speed of the selection frame, (and consequently the gliding speed of the timing line), so that these parameters fit her/his control abilities. Several other parameters can be adapted to the user's needs. For instance, one can modify the minimal and maximal keystroke durations. For users who are still able to control the duration of their keystrokes, we have implemented long and "very long" keystrokes, to which specific functions, such as *erase*, *capitalize*, *new line*, *speech synthesis*, can be attached (according to the user's preferences).

3. FAST KEY SELECTION: SIBYLETTER

In standard AAC systems, key selection is achieved via a line/column scan which significantly reduces the average number of cursor shifts needed to reach the intended key. However, this selection mode requires two keystrokes per item selection (line/column). We learned from user feedback that this kind of selection is rapidly tiring; for this reason we implemented a *dynamic linear scanning mode* which only requires one keystroke per item selection: the cursor here highlights all the keys of the virtual keyboard successively. In order to speed up communication, the keyboard is dynamically rearranged after every selection, in order to first present the most probable letters, according to the letters already typed. This letter prediction is based on a 5-gram letter model, which calculates at every given moment the probability of each character for the given context [3]. Figure 4 shows the dynamic reorganization of the letter keypad, when the user composes the first letters of the word '*three'* on the English version of *SIBYLLE*.

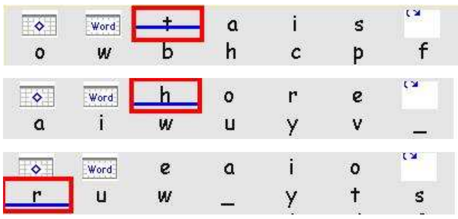

Figure 4. Reorganization of the dynamic letter sub-keypad during the composition of the first two letters of *three*

This dynamic behavior concerns only the linear scan. When the user chooses this selection mode, his/her attention is focused on the selection frame and its immediate neighboring. For this reason the user is not disturbed by the reorganization of the keypad, which certainly explains why the dynamic aspect of the interface did not increase the cognitive load of our users significantly.

Furthermore, the letter prediction component (*SibyLetter*) yields very good results; experiments conducted on a large test corpus show that the wanted character appears on the average at the 3rd position (table 1). This result is remarkable compared to a standard line/column scan, which requires 9 shifts on the average.

Table 1. Comparison of different selection modes (test on newspaper corpus with an item set of 64 characters)

Selection mode	Language	Avg. number of shifts per character	Nb. of keystrokes per char.
static line / column scan	French / German	9,0	2
static linear scan (QWERTY)	French	33,0	1
SIBYLLE : dynamic linear scan (5-gram)	French	2,7	1
	German	3,0	1

4. SAVING KEYSTROKES: SIBYWORD

Considering the satisfactory behaviour of *SibyLetter*, further significant improvements should mainly rely on keystroke reduction. This is achieved by word prediction, a technique that has been shown to speed up communication rates considerably in an AAC system, especially, when it is context sensitive (cf. [4]) In our system this is achieved by the *SibyWord* component, which tries to predict the most appropriate words considering the current context. The predicted word is then either directly displayed after the end of the inserted text (a method referred to as "word completion", cf. [5]), or a list of N-best (typically 3 to 7) predictions is provided on the virtual keyboard. In *SIBYLLE*, a specific sub-keypad displays the most probable words according to *SibyWord* predictions. When the user selects one of these words, it is automatically inserted (or completed) in the current text (see figure 1, showing a list of 7 predicted French words: *le*, *la*, *l'*, *de*, *les*, *et*, *des*).

In our system, word prediction is achieved by means of a stochastic language model, which estimates the probability of occurrence for any word in the lexicon, according to the 3 previously inserted words (4-gram). Normally, this model is calculated on a large corpus of text (usually newspaper).

The first experiments with this 4-gram word predictor have shown that our system is able to save more than half of the keystrokes on a newspaper test corpus. However, language models are highly dependent on their training resources; therefore, the performance of a language model – trained on newspaper text – decreases strongly in real usage with disabled people, who normally do not speak the way newspaper editors write.

Furthermore, since the users respond to very varied clinical patterns and will use AAC systems for varied purposes, we face multi-factorial requests for adaptation. Previous works already emphasized the importance of adaptation for AAC systems (cf. [1], [6]). Whereas these works only consider user adaptation, we have now investigated two kinds of adaptation:

- *User adaptation*, which aims at adapting (in the long term) the word predictor to the user's language style.
- *Semantic adaptation*, which aims at dynamically favouring words that belong semantically to the current topic of communication (short-term adaptation).

5. ADAPTATION TECHNIQUES

5.1 User Adaptation: Dynamic User Model

The user adaptation is achieved through the integration of two language models: a base (4-gram) model, trained on a newspaper corpus and a dynamic user model (DUM), a trigram model which is trained on every text composed by the user; new words are integrated to the model as well. The base LM reflects the general language while the DUM adapts the latter to the specific style and vocabulary of the user. The global common probability $P'(w_i)$ for a word w_i is estimated by linear interpolation of the two models:

$$P'(w_i) = \lambda_1 \cdot P_{Base}(w_i \mid w_{i-1}\, w_{i-2}\, w_{i-3}) + \lambda_2 \cdot P_{DUM}(w_i \mid w_{i-1}\, w_{i-2})$$

where λ_1, λ_2 ($\lambda_1 + \lambda_2 = 1$) are weighting coefficients. They are dynamically adapted, depending on the average success of each of the models in previous predictions. To calculate these parameters, we apply an EM-style algorithm, (cf. [7]).

5.2 Semantic Adaptation: LSA

Latent Semantic Analysis [8] is a technique that models semantic similarity based on co-occurrence distributions of words. LSA, which is founded on cognitive motivations [9], is able to relate coherent contexts to specific content words, and it is good at predicting the occurrence of a content word in the presence of other thematically related terms.

However, since it does not take word order into account ("bag-of-words" model), it is very poor at predicting their actual position within the sentence, and it is completely useless for the prediction of function words. Some attempts have been made to integrate the information coming from an LSA-based model with standard language models of the n-gram type (e.g. [10]).

In the LSA model, a word w_i is represented as a high-dimensional vector, derived by *Singular Value Decomposition* (SVD) from a term × document (or a term × term) co-occurrence matrix of a training corpus. In this framework, a context or history h (= w_1, ..., w_m) can be represented by the (normalized) sum of the vectors, corresponding to the words it contains [9]. This vector reflects the meaning of the preceding (already typed) section, and it has the same dimensionality as the term vectors. It can thus be compared to the term vectors by well-known similarity measures (scalar product, cosine).

In our AAC application, we make the assumption that an utterance or a text to be entered is usually semantically cohesive. We then expect all word vectors to be close to the current context vector, whose corresponding words belong to the semantic field of the context. This forms the basis for a simple (pseudo-) probabilistic model based on LSA: after calculating the cosine similarity for each word vector $\vec{w}_i$ with the vector $\vec{h}$ of the current context, we could use the normalized distances as probability values. This probability distribution however is usually rather flat (i.e. the dynamic range is low). For this reason a contrasting (or temperature) factor γ is normally applied [10], which raises the cosine to some power (γ is normally between 3 and 8; we got best results with $\gamma = 4$). After normalization we obtain a probability distribution, which can be used for prediction purposes. It is calculated as follows:

$$P_{LSA}(w_i|h) = \frac{\left(\cos(\vec{w}_i, \vec{h}) - \cos_{\min}(\vec{h})\right)^{\gamma}}{\sum_k \left(\cos(\vec{w}_k, \vec{h}) - \cos_{\min}(\vec{h})\right)^{\gamma}}$$

w_i is a word in the vocabulary, h is the current context (history), $\cos_{\min}(\vec{h})$ returns the lowest cosine value measured for $\vec{h}$). The numerator then normalizes each similarity value to ensure that the probabilities sums to 1.

Let us illustrate the capacities of this model by giving a short example from the French version of our own LSA predictor. Suppose that the user has already typed the following sentence:

(1) *Mon père a été professeur de mathématiques et je pense que..*

[My dad has been a professor in mathematics and I think that…].

Table 2 shows the ten words that are presenting the highest LSA probabilities: all ten predicted words are semantically related to the context, they should therefore be given a higher probability of occurrence.

Table 2. Most probable words provided by LSA for the sentence (1) as a given context

Rank	Word	P_{LSA}
1	professeur ('professor')	0,0117
2	mathématiques ("mathematics")	0,0109
3	enseigné (participle of 'taught')	0,0083
4	enseignait ('taught')	0,0053
5	mathématicien ('mathematician')	0,0049
6	père ('father')	0,0046
7	mathématique ('mathematics')	0,0045
8	Grand-père ('grand-father')	0,0043
9	Sciences ('sciences')	0,0036
10	enseignant ('teacher')	0,0032

However, this example also shows the drawbacks of the LSA model: it totally neglects the presence of function words as well as the syntactic structure of the current phrase. We therefore need to integrate the information coming from a standard n-gram model and the LSA approach.

Interpolation is the usual way to integrate information from heterogeneous resources. While for a linear combination we simply add the weighted probabilities of two (or more) models, geometric interpolation multiplies the probabilities, which are weighted by an exponential coefficient ($0 \le \lambda_1 \le 1$).

$$P'(w_i) = \frac{P_b(w_i)^{\lambda_1} \cdot P_s(w_i)^{(1-\lambda_1)}}{\sum_{j=1}^{n} P_b(w_j)^{\lambda_1} \cdot P_s(w_j)^{(1-\lambda_1)}}$$

Geometric interpolation gives better results, since it takes the agreement of two models into account. Only if each of the single models assigns a high probability to a given event, the combined probability will high. If one of the models assigns a high value and the other does not, the resulting probability will be lower than the linear average.

Finally, whereas in standard settings the interpolation coefficients are stable for all probabilities, we use confidence-weighted coefficients that are adapted for each probability. Coccaro & Jurafsky [10] proposed an entropy-related confidence measure for LSA, based on the assumption that words occurring in many different contexts (i.e. have a high entropy), cannot well be predicted by LSA. Measuring relation quality in an LSA space, One of the authors [11] showed that the entropy of a term does not correlate with relation quality (i.e. the number of semantically related terms in an LSA-generated term cluster), but he found a medium correlation between the number of semantically related terms and the average distance of the *m* nearest neighbors (density). The closer the nearest neighbors of a term vector are, the more probable it is to find semantically related terms for the given word. In turn, terms having a high density are more likely to be semantically related to a given context. We therefore use a density-based measure to achieve this confidence weighted interpolation of the LM and the LSA models. These aspects have been described in more detail in [12].

5.3 Semantic adaptation: related work

There are a number of approaches that tried to adapt a word predictor to the current semantic context. On the one hand we find approaches like [1] and [15] that make use of the trigger model, presented in [13]. This model is based on the idea that the appearance of a word *x* (the trigger) makes the appearance of another, semantically related word *y* (the target) more likely. For example, if a word like "*foul*" has already occurred in the text, "*referee*" or "*yellow card*" are much more likely to appear. The trigger-target pairs are usually calculated by collocation measures (such as *Point-Wise Mutual Information*, cf. [14]) from large corpora. Trost et al. [1] have evaluated such a model for German, however their gains remained modest (+0.25% *ksr* with respect to the baseline).

On the other hand, approaches like [6] make use of topically assigned corpora, from each of which a separate language model is calculated. These single topic-related LMs are then dynamically interpolated, so that the overall LM gives highest weight to the LM whose topic is closest to the current topic of discourse. In [6], this model shows a *ksr* advantage of 1.4% over a 3-gram baseline. However, a drawback of this model is the need for topically assigned corpora. Such corpora do exist for English (e.g. the *Switchboard* corpus), but they are not (yet) available for other languages such as German or French.

6. RESULTS (FRENCH AND GERMAN VERSIONS)

For our experiments, we calculated our baseline 4-gram model on a 44 million word corpus from the French daily *Le Monde* (1998-1999). Using the *SRI* toolkit [17] we computed a 4-gram LM over a controlled 141,000 word vocabulary, using *modified Kneser-Ney* discounting [18], and we applied *Stolcke* pruning [16] to reduce the model to a manageable size ($\theta = 10^{-7}$). The LSA space was calculated on a 99.7 million word corpus from *Le Monde* (1996 – 2002). Using the *Infomap* toolkit[2], we generated a term × term co-occurrence matrix for an 80,000 word vocabulary (matrix size = 80,000 × 3,000), stopwords were excluded. After several pre-tests, we set the size of the co-occurrence window to ±100. The matrix was then reduced by singular value decomposition to 150 columns.

The German language model was calculated on a 37 million word corpus from the newspaper *Tageszeitung* (1997-1999), for the calculation, we used the same model parameters as above. For the LSA space we used 90.1 million words, also from the *Tageszeitung* corpus (1989-1998). Again, we calculated a 80,000 × 3,000 co-occurrence matrix, which was reduced to 150 dimensions.

6.1 Keystroke saving rate

It is difficult to assess objectively how a word predictor can really speed up communication rates. Indeed, the observed improvements depend strongly on the user, and on the interaction between the prediction component and the user interface as well. In this paper, we will study the behaviour of the word predictor separately, measuring its theoretical ability to save keystrokes. Classically, word predictors are evaluated by an objective metric called *Keystroke Saving Rate* (*ksr*):

$$ksr_n = \left(1 - \frac{k_p}{k_a}\right) \cdot 100$$

with k_p, k_a being the number of keystrokes needed on the input device when typing a message with (k_p) and without prediction (k_a = number of characters in the text that has been entered, n = length of the prediction list, usually $n = 5$). As [1] and [6], we assume that one additional keystroke is required for the selection of a word from the list and that a space is automatically inserted afterwards. Note also that words, which have already occurred in the list, will not reappear after the next character has been inserted.

The perplexity measure, which is frequently used to assess statistical language models, proved to be less accurate in this context, particularly when new words are added during the prediction process.

In order to study the adaptation of our system, we assessed *SIBYLLE* on several test corpora that correspond to various communication situations (cf. also [19]):

A) *Newspaper*: Extracts from French (*Humanité*, 58,457 words) or German (*Süddeutsche Zeitung*, 56,031 words) newspapers.

B) *Scientific*: A scientific article (unpublished) of one of the authors, from the domain of NLP; 8,766 words.

C) *Prose*: 1st chapter from *Germinal* from Zola; 20,928 words

D) *Speech*: Transcription of spontaneous dialog between French tourist agents and customers; 15,435 words.

E) *E-mail*: Personal e-mails of one of the authors, where headers, replies and hyperlinks were removed; 8,874 words.

For each test-set we then calculated the keystroke saving rate based on a 5-word list (ksr_5) for the following settings:

- 4-gram LM only (*Baseline model*)
- 4-gram interpolated with a *Dynamic User Model* (DUM).
- 4-gram + LSA model (with geometric interpolation and confidence weighting).
- 4-gram + DUM + LSA

[2] *Infomap Project*: http://infomap-nlp.sourceforge.net/

Tables 3 and 4 sum up the overall performances of these models on different French and German evaluation corpora. At first glance, one can see that the overall performances of *SIBYWORD* are very satisfactory: whichever test corpus was considered, the *ksr* remains higher than 50% (right column on table 3), meaning that the user can save one keystroke over two in every situation.

Table 3. Performances (ksr_5) of the French version of *SIBYWORD* on different communication situations

Corpus	4-gram (baseline)	4-gram + DUM	4gram + LSA	SIBYWORD (4-gram + DUM + LSA)
Newspaper	57.8%	58.5%	58.9%	59.4%
Scientific	44.2%	52.4%	45.6%	52.9%
Prose	46.2%	50.1%	48.0%	51.4%
Speech	48.3%	57.7%	49.9%	57.9%
E-mail	50.1%	52.5%	51.7%	53.2%

Table 4. Performances (ksr_5) of the German version of *SIBYWORD* (newspaper corpus)

Corpus	4-gram (baseline)	4-gram + DUM	4gram + LSA	SYBIWORD (4-gram + LSA + DUM)
Newspaper	51.6%	54.6%	52.6%	55.4%

Furthermore, comparison with the 4-gram LM baseline shows the benefits of our adaptation techniques. For the dynamic user model, we get an important increase of *ksr* for all test corpora. Even for the test corpus that belongs to the same register (newspaper) as the training data, we get a slight improvement of performances. A detailed analysis of our results shows that the *Dynamic User Model* is able:

- To reduce the number of unknown words (Out-Of-Vocabulary words, OOV): this is particularly important in case of the scientific corpus, which presents a high rate of OOV (16,6%) for the baseline model.
- To adapt the model to the communication style: this observation applies particularly to the spoken dialogue corpus, which shows many phrasal disruptions.

One should note that the influence of the *Dynamic User Model* rapidly gains significance. Figure 6 summarizes an experiment we conducted on two of our corpora (*prose* and *e-mail*). It shows that a *ksr* increase of 2% is already observed with only 2,000 words of user training data. This represents on the average 10 hours of continuous typing for an experienced user.

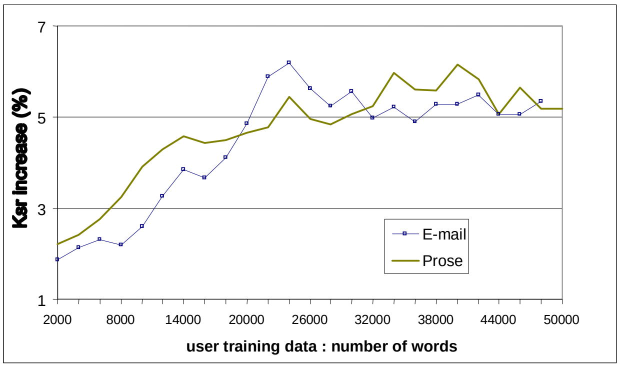

Figure 5. Influence of the *Dynamic User Model*: ksr_5 increase according to the amount of training data

The semantic adaptation that is achieved by the LSA model leads to a less important but nearly constant increase of *ksr* (+1.0% to +1.7%). This benefit is cumulative with the DUM, the combined model always yields the best results. In particular, additional experiments [11] have shown that the improvements due to the LSA adaptation is on average five times higher than that of a standard decaying cache model (cf. [20]).

Finally, the performances of the German version of *SIBYLLE* are 4% to 6% lower than those of the French one. Since the test corpora are obviously different, it is difficult to interpret these differences. Nevertheless, it is probable that this weakness results from the high frequency of compound words in German, which are hardly predictable with a standard model. This is why we need to develop a specific component to handle these complex words. In particular, we are presently studying the influence of the "partial selection" approach implemented in the *FASTY* AAC system [1].

6.2 User assessment: Kerpape center

The *SIBYLLE* system benefits from the experience of seven years of daily use in the rehabilitation center of Kerpape (Brittany, France). Its successive versions have been used by more than twenty patients of the center. Some of them are adults, but the majority of the users are children from the school integrated in the center. The system was very appreciated by most of the patients. Only one of them, who is suffering from severe visual impairments, felt uncomfortable with the dynamic virtual keyboard.

The other patients and the practitioners noticed a significant acceleration of the text insertion process. We have also observed that the children which are studying in the Kerpape school accept to make longer working sessions. This indicates that the use of *SIBYLLE* implies less physical fatigue, compared to the AAC systems that were previously used in the center. This reduction of the physical fatigue of the users is certainly as important as the improvement of the communication speed [21].

Finally, we also noticed a significant reduction of orthographic and grammatical errors when the patients are using the system. A comparable result has already been observed with others AAC systems (see for example the *Profet* system [22]). This observation is particularly when the user suffers from additional language impairments.

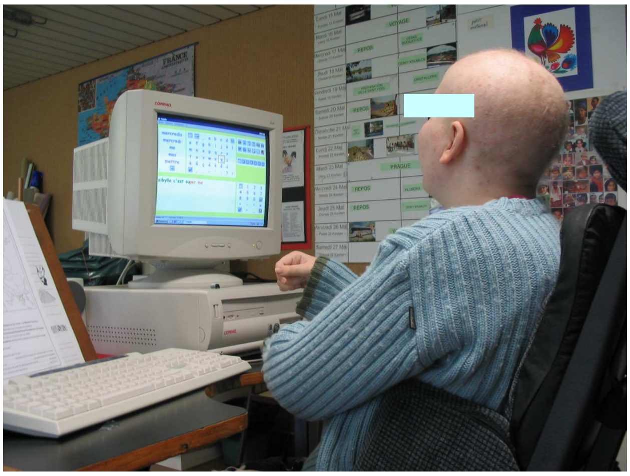

Figure 6. The *SIBYLLE* system (version 1.5.2) used by an athetosic child from the integrated school of the Kerpape rehabilitation center

A disturbing observation is that, frequently, our users do not select the intended word although the latter is clearly presented in the prediction list. In an experiment conducted with the commercial DIALLO system, Biard *et al*. [23] observed that their patients selected only 2,300 word hypotheses during the composition of text summing up to 80,000 letters. Our discussions with the users and the practitioners tend to show that this situation, which limits obviously the *ksr* and the communication speed, is due to a cognitive problem: the users meet clear difficulties to write a message and read the list simultaneously, due to an increase of the cognitive load.

A possible solution to this problem should be to implement a direct completion like in the VITIPI system [5]: instead of presenting a list of several word hypotheses on a specific sub key-pad, one can propose the most probable termination of the current word immediately after the latest typed letter. It is however not sure that this immediate display is sufficient to limit the conflict between input (read the prediction) and output (write the message) activities.

Table 5. *Ksr* of the French version of *SIBYWORD* (newspaper corpus) according to the size of the prediction list

Size	Ksr_1(1 word)	Ksr_2(2 words)	Ksr_5(5 words)
Newspaper	44.4%	51.1%	57.9%

Another solution is to insert word predictions directly in the letter keypad: the first keys will display these words and the following ones the predicted letters. Then the user will only have to focus his/her attention on the selection frame. The scanning of these additional keys obviously increases the time needed to reach a letter. Nevertheless, our experiments on the *Le Monde* corpus suggest that this strategy could be useful. Studying the influence of the size of the prediction list (table 5), we have found that a satisfactory *ksr* can be reached with only one (44.4%) or two (51.6%) lexical hypotheses. This means that the insertion of one or two words in the letter keypad is generally sufficient to benefit from the advantages of the word prediction component.

Moreover, one must consider that this selection mode (and direct completion as well) requires a unique keystroke, while two successive keystrokes are needed to jump to the word list and to select a word in the "standard" strategy. This point is important as well when considering keystroke saving. It should compensate the fact that fewer hypotheses are proposed to the user. Obviously, different users will prefer different selection modes if they are, above all, considering communication speed or on the contrary physical fatigue, or cognitive load. For this reason we are currently implementing two additional selection modes for *SIBYLLE*: direct completion and word selection from the letter keypad.

7. CONCLUSION AND PERSPECTIVES

Despite these encouraging results, we still need more information about real uses of AAC systems with patients presenting a large variety of clinical characteristics. In particular, a significant part of motion and speech disabled users also suffer from severe cognitive impairments. As a result, the messages they compose are highly ungrammatical, which disturbs our word predictor.

We are thus involved in the ESAC_IMC project (*Fondation Motrice*), whose aim is to collect and analyze a large corpus of real-use sessions on three AAC systems for French. The participants (Kerpape, LI, IRIT and VALORIA) have defined a common XML format for the log files that are being recorded. These log files keep track of the following events:

- all actions of the user (keystroke, corresponding item of the virtual keyboard, time indication)
- all replies/actions of the system, among which the list of word or letter predictions

Furthermore, we keep the clinical description of all the recorded users. This information will be very useful to characterize real needs for AAC according to different kinds of handicap. The recordings of these log files are now in progress in the rehabilitation center of Kerpape.

8. ACKNOWLEDGMENTS

The authors thank Jean-Paul Départe for his helpful assistance in the experiments with disabled people from the Kerpape rehabilitation center. We also want to thank the developers of the *SRI* and the *Infomap* toolkits for making their programs available.

This research is partially founded by the UFA (Université Franco-Allemande) and two foundations: AFM (VOLTAIRE project) and *La Fondation Motrice* (*ESAC_IMC* project).

9. REFERENCES

[1] Trost, H., Matiasek, J. and Baroni, M. The Language Component of the FASTY Text Prediction System, *Applied Artificial Intelligence*, 19(8), pp. 743-781. 2005

[2] Ward, D. , Blackwell, A., McKay, D. Dasher: A Data Entry Interface Using Continuous Gestures and Language Models". *Proceedings of the 13th Annual ACM symposium on User Interface Software and Technology, UIST*'2000, San Diego, CA. pp. 129-137. 2000.

[3] Schadle, I. Sibyl: AAC system using NLP Techniques. *Proceeding of the 9th International Conference on Computers Helping People with Special Needs, ICCHP'04*. Paris, France, pp. 1109-1015. 2004.

[4] Trnka, K., Yarrington, D. McCoy, K. F. and Pennington, C. The Effects of Word Prediction on Communication Rate for AAC. *Proceedings of NAACL HLT'2007*, Rochester, pp. 173-176. 2007.

[5] Boissière Ph., Dours D. From a specialised writing interface created for the disabled, to a predictive interface for all: the VITIPI system. *Proceedings of 1st International Conference UAHCI*. New-Orleans, USA.Lawrence Erlbaum Associates, Publishers, Mahwah, NJ, p. 895-899, august 2001.

[6] Trnka, K., Yarrington, D. McCoy, K. F. and Pennington, C. Topic Modeling in Fringe Word Prediction for AAC. In *Proceedings of the 2006 International Conference on Intelligent User Interfaces*., Sydney, Australia. pp. 276-278. 2006.

[7] Jelinek, F. Self-organized Language Models for Speech Recognition. In: A. Waibel and K.-F. Lee (eds.) *Readings in Speech Recognition*, Morgan Kaufman Publishers, pp. 450-506. 1990.

[8] Deerwester, S. C., Dumais, S., Landauer, T., Furnas, G. and Harshman, R Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science, JASIS* 41(6), pp. 391-407. 1990.

[9] Landauer, T. K., Laham, D., Rehder, B. and Schreiner, M. E. How well can passage meaning be derived without using word order? A comparison of LSA and humans", *Proceedings of the 19th annual meeting of the Cognitive Science Society*. pp. 412-417, Erlbaum Mawhwah, NJ. 1997.

[10] Coccaro, N. and Jurafsky, D. Towards better integration of semantic predictors in statistical language modeling, *Proc. of the Intl. Conf. Spoken Language Processing*, *ICSLP-98*. Sydney, Australia. paper 0852. 1998.

[11] Wandmacher T. How semantic is Latent Semantic Analysis?, *Proc*. *TALN/RECITAL'2005*, Dourdan, France. 6-10 june 2005.

[12] Wandmacher, T. and Antoine, J.-Y. Methods to integrate a language model with semantic information for a word prediction component. To appear in: *Proceedings of EMNLP-2007*, Prague, Czech Republic, 2007.

[13] Rosenfeld, R. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech and Language*, 10 (1), pp. 187-228, 1996.

[14] Church, K. and Hanks, P. Word association norms, mutual information and lexicography. *Proc. 27th Annual Meeting of the Association for Computational Linguistics, ACL'89*. Vancouver, BC, Canada. pp. 76-83. 1989.

[15] Li, J. and Hirst, G. Semantic knowledge in a word completion task. *Proceedings of the 7 Int. ACM SIGACCESS conference on Computers and Accessibility*, Baltimore, MD, USA. pp. 121-128. 2005.

[16] Stolcke, A. Entropy-based pruning of backoff language models". *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA. pp. 270-274 1998

[17] Stolcke, Andreas SRILM - An Extensible Language Modeling Toolkit, *Proceedings of the Intl. Conf. Spoken Language Processing, ICSLP'2002*, Denver, Colorado. vol. 2, pp. 901-904. September 2002

[18] Goodman, J. A Bit of Progress in Language Modeling, Extended Version *Microsoft Research Technical Report MSR-TR-2001-72* . 2001.

[19] Wandmacher, T. and Antoine, J.-Y. Training language models without appropriate resources. Experiments with an AAC system for disabled people. In: *Proc. of the5th Language Resources and Evaluation Conference, LREC'06*, Genova, Italy, 2006.

[20] Clarkson, P. R. and Robinson, A.J. Language Model Adaptation using Mixtures and an Exponentially Decaying Cache. In: *Proc. of the IEEE ICASSP-97*, Munich, 1997.

[21] Berard C., Neimeijer D. Evaluating effort reduction through different word prediction systems. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, La Haye NL. Vol. 3, pp. 2658-2663. oct. 2004.

[22] Carlberger A., Carlberger J., Magnuson T., Hunnicutt M.S., Palazuelos-Cagigas S., Navarro S.A. Profet, a new generation of word prediction : an evaluation study. *Proc. Natural Language Processing for Communication Aids, NLPCA'1997*. Madrid, Spain. pp. 23-28. 1997

[23] Biard N., Dumas C., Bouteille J., Pozzi D., Lofaso F., Laffont I., Apports de l'évaluation en situation de vie à partir d'une étude sur l'intérêt de la prédiction de mots auprès d'utilisateurs de synthèse vocale, *Proceedings Handicap 2006*, Paris, France. 145-148. 2006.