# SIBYLLE, an Assistive Communication System Adapting to the Context and its User[*]

TONIO WANDMACHER AND JEAN-YVES ANTOINE
Laboratoire d'Informatique (LI), Université François Rabelais de Tours – IUP Blois, 3 place J. Jaurès, F-41000 Blois, France
and
FRANCK POIRIER
Université Européenne de Bretagne UBS, CERYC, Campus de Tohannic, F-56000 Vannes, France
and
JEAN-PAUL DEPARTE
Centre Mutualiste de Rééducation et de réadaptation fonctionnelle de Kerpape, F-56275 Ploemeur, France

In this paper, we describe the latest version of SIBYLLE, an AAC system that permits persons who have severe physical disabilities to enter text with any computer application, as well as to compose messages to be read out through speech synthesis. The system consists of a virtual keyboard comprising a set of keypads which allow for the entering of characters or full words by a single-switch selection process. It also includes 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 with every text the user enters. 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 a rehabilitation center (Kerpape, France).

Categories and Subject Descriptors: J.3.3 [Compute applications]: Life and medical sciences
General Terms: Human Factors, Experimentation, Performance.
Additional Key Words and Phrases: Augmentative and Alternative Communication; Virtual keyboard; Word prediction; Latent Semantic Analysis; User adaptation; Keystroke saving rate

## 1. INTRODUCTION

This paper presents SIBYLLE, an AAC (Augmentative and Alternative Communication) system for persons with severe speech and motion impairments (cerebrally and physically disabled persons, *Locked-in* syndrome, cerebral palsy etc.). Whatever the disease or impairment considered, oral communication is impossible for these persons who also have serious difficulties in physically controlling their environment. In particular, they are not able to use the standard input devices of a computer. Like other AAC systems, such as *FASTY* [Trost et al., 2005] or *Dasher* [Ward et al., 2000] SIBYLLE aims at restoring the communicative abilities of these persons.

[*] This article represents an extended version of a paper originally published in the proceedings of the 9th *ACM SIGACCES Conference on Computers and Accessibility* (*ASSETS*), 2007, cf. [Wandmacher et al., 2007].

Many AAC systems have a very similar architecture, consisting of 4 components (Figure 1). At first, one finds a physical input interface connected to the computer. This device is adapted to the motion capacities of the user. When the latter must be restricted to a single switch (eye glimpse or breath sensor, for instance), the control of the environment is reduced to a mere Yes/No command.

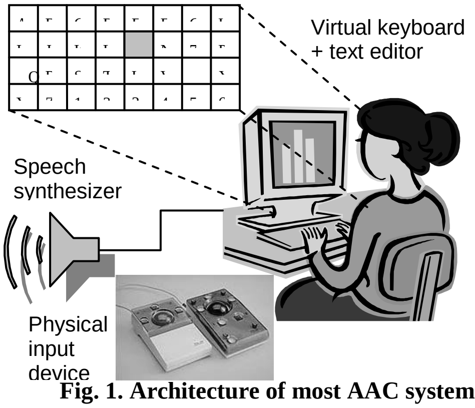


**Fig. 1. Architecture of most AAC systems**

Secondly, a virtual keyboard replaces the physical keyboard by displaying a table of symbols (words, letters or icons) on screen. It allows the user to select successively the symbols that will compose the intended message. In SIBYLLE, symbol selection is achieved by a linear scan procedure: a cursor successively highlights each key of the virtual keyboard which can then be selected by the user.

The last two components are a text editor and a text-to-speech synthesizer, which is used to read out the typed message for spoken communication. The latest version of SIBYLLE works for French, German and English, and it is usable with any *Windows*™ application (text editor, web browser, mailer...).

The main challenge of AAC systems results from the slowness of message composition. Whereas people can produce up to 200 words per minute in oral communication, persons using an AAC device cannot type more than 1 to 15 words per minute, depending on their abilities and the configuration of the system [Alm et al., 1992]; moreover, this task is very tiring.

We thus investigate two complementary approaches intended to speed up text input: fast key selection and keystroke reduction. These improvements are based on two prediction modules which will be described in this paper. At first, we present the user interface of our system. Sections 3 and 4 describe in detail the prediction modules which have been developed for SIBYLLE.

In sections 5 and 6 we describe and evaluate the adaptation capacities of the word prediction component, which takes into account the user's way of speaking, as well as the current semantic context. Finally, we present some first results from user feedback and give a brief outlook on the following steps we plan to take in the development of our system.

## 2. SIBYLLE: THE USER INTERFACE

### 2.1. Interface design of an AAC system

Although text entry methods can significantly improve the efficiency of an AAC system, its usability mainly depends on its user interface. A large diversity of interfaces can be found in the literature. This heterogeneity results from the variety of human factors that directly affect the usability of an interface:

1. *Physical or motor control abilities of the user:* if the user is still able to control to a certain extent his/her gestures, devices such as a *finger guide* or a *grid keyboard* could be added onto the physical keyboard to avoid erroneous selections. When the motor control abilities of the user are more restricted, the use of a virtual keyboard becomes indispensable. Although one can imagine a large variety of arrangements, most AAC systems emulate the functions of a standard physical keyboard. Some enable however the user to define his/her own keyboard [cf. Vella et al., 2005]. In some cases, the user is still able to control a mouse, but most of the time, he/she can only use a single input device. Then, key selection is usually achieved by a scanning procedure. Alternatives to this standard solution also exist. For instance, the *Dasher* system proposes an inventive procedure of dynamic browsing between plausible letters, controlled by eye movements [Ward et al., 2000].
2. *Cognitive abilities of the user:* text entry methods are only useful when the user has sufficient linguistic knowledge or at least a certain phonetic ability. Young children or people who have additional language disorders (aphasia, dyslexia) will therefore employ a virtual keyboard with iconic keys. Several systems have developed such an interface [cf. Baker, 1982; Abraham, 2002].
3. *Perceptive abilities of the user:* some diseases or disabilities can include additional perceptive difficulties (e.g. cerebral palsy). For instance, people whose vision is disturbed by rapid movements could have difficulties using a dynamic keyboard like in the *Dasher* system. A static keyboard with standard scanning is better adapted to this type of case, although its theoretic communication rate is lower.
4. *Keystroke saving method*: most AAC systems include some method to speed up communication by trying to save the number of keystrokes needed for the

composition of a message. Some systems have investigated abbreviation expansion for text input [McCoy and Demasco, 1995; Willis et al., 2002; Shieber and Baker, 2003]: here the user types an abbreviated form that will be automatically expanded by the system. Other AAC systems use prediction techniques to reduce keystrokes. Two main strategies can be found here: on the one hand, the system only considers the most probable word, which is directly inserted in the message [cf. Boissière and Dours, 2001], an approach often referred to as 'word completion'. On the other hand, a list of words is provided [Trost et al., 2005; Berard and Neimeijer, 2004], from which the user can then select the intended word.

As demonstrated so far, many factors influence the usability of an AAC system. It is therefore of first importance in the design of such a system to keep the interface as adaptable as possible to the user's needs.

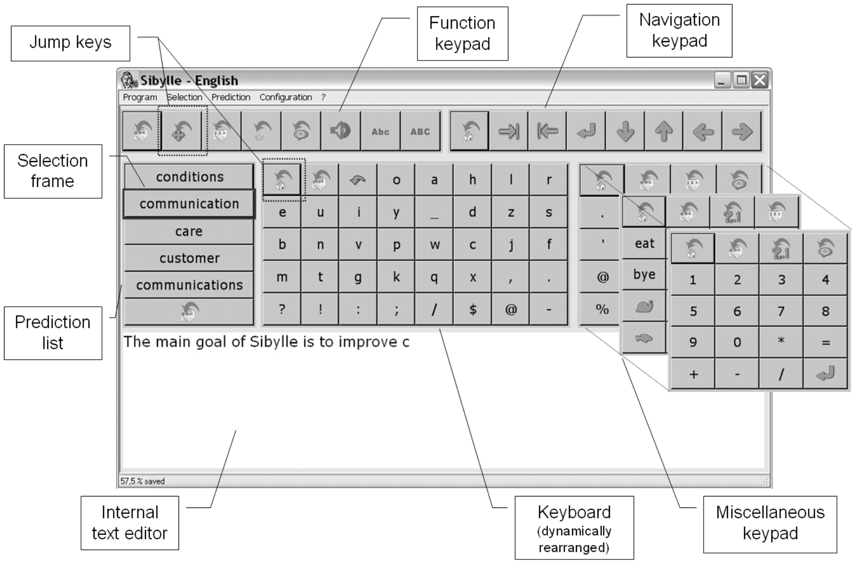
### 2.2 User interface of SIBYLLE

The development of our system has been oriented along a few major design decisions:

- SIBYLLE is above all designed for single switch input devices (users with severe motor impairments),
- Keystroke reduction is based on a word prediction model, even if SIBYLLE also includes a basic module for abbreviation expansion.
- For users who are not visually impaired, a fast key selection technique is proposed by means of a dynamic reorganization of the keyboard (see below).

Figure 2 shows the latest version of the user interface of SIBYLLE. 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 hungry, I want to drink*"). Jump keys provide fast moves between these sub-keypads: they are usually the first keys on each keypad. The different keypads of the interface are displayed in Figure 2.

- Letter keypad: it is used to compose messages, character after character. When the user activates the letter prediction component of SIBYLLE (s. section 3) with the linear scan mode, the keys are dynamically rearranged in order to present the most probable letters first. Since punctuation signs and numbers are hardly predictable, they are displayed in a separate keypad. Thus, the letter keypad only comprises alphabetic characters, as well as the space symbol.
- *Prediction list*: when the user selects one of these predicted words, it is automatically inserted in the message. The user can choose between a horizontal and a vertical layout of the list.



The main goal of Sibylle is to improve c

**Fig. 2. The user interface of SIBYLLE (version 3.6, English)**

Previous works have suggested that a vertical arrangement of the word list is better accepted than a horizontal one [cf. Garay-Vitoria and Abascal, 2006].

- *Function keypad*: this keypad is displayed by default on the upper area of the user interface. In former versions, SIBYLLE only comprised an integrated text editor connected with a text-to-speech synthesizing application. But since users also want to compose e-mails, use a real word processor or a search engine on the web, we decided to make SIBYLLE more flexible. By interfacing the *Microsoft Windows* API, our system is now able to enter text in any kind of *Windows* application. Furthermore, configurable function keys enable direct action such as *Save As*, *Open* or *speech synthesis.*
- *Navigation keypad*: like in an ordinary physical keyboard, this keypad enables the user to move the text cursor without operating a mouse. It should be used when composing messages, but is above all useful to move the pointer between the menus of any *Windows* application.
- *Miscellaneous keypad*: this keypad can be used in several modes. One can use it to select numbers, but also punctuation marks, and finally to select predefined sentences or messages. These messages can be adapted for the user. Figure 3 shows the different default layouts of the keypad according to the selected mode. Pre-recorded messages are represented by a little icon. We plan to allow the user to define her or his own icon sets in the short term.

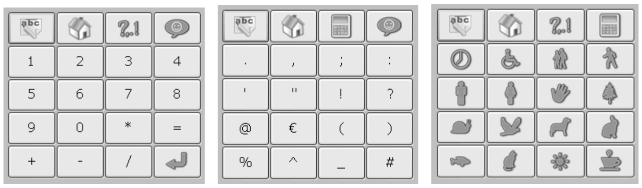


**Fig. 3. The three-fold layout of the miscellaneous keypad according to the selected mode (left: number selection, middle: punctuation, right: pre-recorded messages)**

When the user is not able to control a mouse, key selection is performed by a scanning strategy: a selection frame successively highlights each key, which can then be selected. Experiments with our system have shown that the users are often disturbed by the abrupt shifts of the selection frame. When the cursor approaches the desired key, they have difficulties to temporally prepare their action. As a result, we observed a significant rate of selection errors. For this reason we have added a timing line, which gently glides from the top to the bottom of the frame (Figure 4) and shows the time remaining until the next shift by its position. This temporal feedback has proven useful to many users.

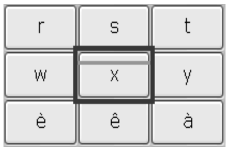
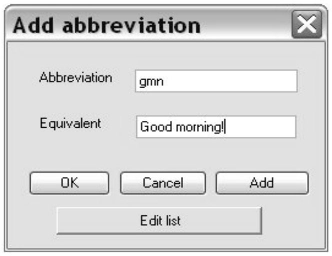


**Fig. 4. Selection frame with the timing line**

For users who are still able to control the keystroke duration, we have implemented a *click timer* to which specific functions (such as erasing, capitalizing or jumping to other windows) can be assigned. This timer distinguishes up to three durations (short/long/very long click).


Add abbreviation

Abbreviation gmn

Equivalent Good morning!

**Fig. 5. User interface for the definition of abbreviations**

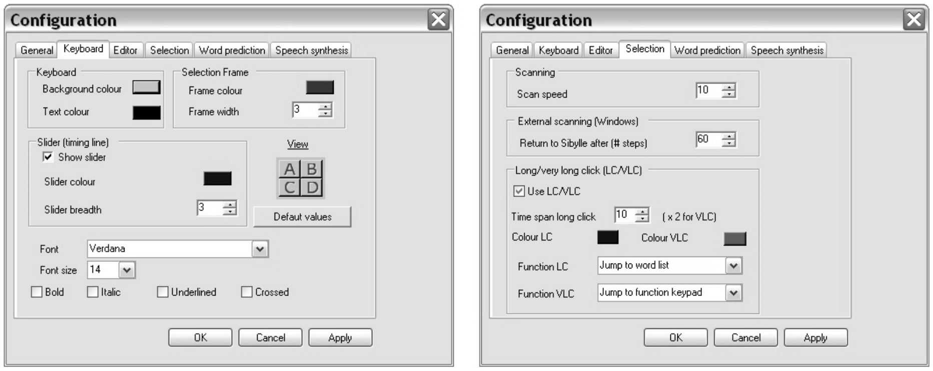
SIBYLLE also includes a module for abbreviation expansion. This component enables the user to define his/her own abbreviations that are directly expanded during the composition of a message (Figure 5).

### 2.3 Interface configurability

One of the most distinguishing features of SIBYLLE is the high adaptability of its user interface: In order to best meet the individual requirements of every user, we designed all interactive elements as modifiable as possible. This extends to the whole layout of the interface, for example:

- *Keyboard rendering*: Colors, fonts and font size of all keyboards as well as the keys themselves can be modified and rearranged.
- *Selection parameters*: Scanning mode, scanning delays, time spans and long-click functionality (such as direct jumps to other windows or capitalizing) are adjustable.
- *Interface layout*: The size and position of every sub-keypad within the application window can be set individually.

Figure 6 shows some of the configuration panels of SIBYLLE.



Keyboard

Background colour ▭

Text colour ■

Selection Frame

Frame colour ■

Frame width 3

Slider (timing line)

☑ Show slider

Slider colour ■

Slider breadth 3

Font Verdana

Font size 14

□ Bold □ Italic □ Underlined □ Crossed

Scanning

Scan speed 10

External scanning (Windows)

Return to Sibylle after (# steps) 60

Long/very long click (LC/VLC)

☑ Use LC/VLC

Time span long click 10 ( x 2 for VLC)

Colour LC ■ Colour VLC ■

Function LC Jump to word list

Function VLC Jump to function keypad

**Fig. 6. Two of the configuration panels of** SIBYLLE **(v.3.6)**

An optimal adaptation of the interface however can of course only be achieved by close interaction between the user and the medical staff (cf. also section 7).

## 3. FAST KEY SELECTION: SIBYLETTER

In many 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. We learned from user feedback that this kind of selection is rapidly tiring. For this reason we implemented in SIBYLLE a linear scanning mode: the cursor highlights all the keys one after the other, and only one keystroke is needed to select the desired key when highlighted.

In order to speed up communication with a linear scan procedure, a basic idea is to order the letters according to their frequency in the considered language. A frequency-ordered keyboard has a significantly lower access time than a standard QWERTY layout. An even faster access can still be achieved if the letter ordering is dynamically rearranged, according to the probability of each character in the current context. An ‘*x*’ for example is much more probable after an ‘*e*’ than after a ‘*t*’ or a ‘*j*’. The letter keypad of SIBYLLE is updated after every selection, so that the most probable letters according to those already composed are presented first. The dynamic reordering of the keypad is directed by *SibyLetter*, a letter prediction module based on a 5-gram letter model. This statistic model estimates at every given moment the conditional probability of each character given the four previously typed symbols [cf. Schadle, 2004]:

$$P(c_i) \approx P(c_i \mid c_{i-1}, \ldots, c_{i-4}). \tag{1}$$

Spaces between words and all punctuation signs are considered in the context of these last four characters. Three models were trained through large corpora for French, German and English. Data sparseness is managed with a simple back-off technique [Allen, 1997]: if a specific n-gram is not observed in the training corpus, its probability is estimated from the (*n*-1)-gram. As an illustration, Figure 6 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.

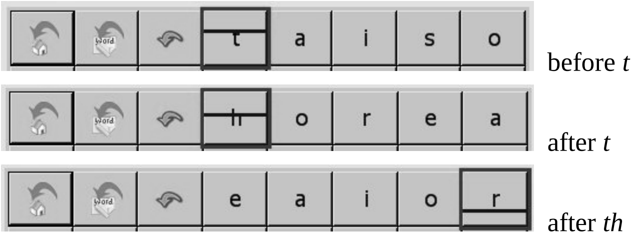


**Fig. 7. Reorganization of the dynamic letter keypad during the composition of the first two letters of ‘three’**

This dynamic behavior can only be activated when the user has chosen the linear scan for selection. In this mode, his/her attention is focused to the selection frame and its immediate environment. For this reason the user is not disturbed by the reorganization of the keypad. This certainly explains why the dynamic aspect of the interface does not seem to increase significantly the cognitive load on the users, in comparison to a static line/column keyboard (cf. section 7).

To assess the performance of *SibyLetter*, we have conducted an experiment on two newspaper corpora of app. 50,000 words (*Humanité* (French) and *Tageszeitung* (German)). Our results show that in the dynamic mode the wanted character appears on the average at the 3rd position of the keyboard (Table I, cf. also [Schadle et al., 2001]). This result is remarkable, compared to a standard line/column scan, which requires approximately 9 shifts.

Obviously, equivalent results can be reached with a dynamic reordering based on the consideration of a frequency-based dictionary. One interest of our approach is precisely that we are able to achieve fast key selection without the use of a dictionary. This is an important point considering that out-of-vocabulary words (OOV) and especially typing or spelling errors affect the robustness of AAC systems.

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

**Table I. Comparison of different selection modes with a 64 char. set (upper and lower case)**

## 4. SAVING KEYSTROKES: SIBYWORD

Another major strategy to accelerate communication is to reduce the number of characters that have to be typed. In our system, keystroke reduction is mainly 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 [Trnka et al., 2007]. The latest version of our word predictor, *SibyWord*, is not only sensitive to the context, it also adapts to the user’s way of communicating as well as to the current topic of discourse. In the following we will present the underlying statistical model, making this adaptation possible.

### 4.1 Theoretical basis of word prediction

Word prediction is only possible because natural language is highly redundant. Every word prediction method eventually exploits this redundancy, be it of syntactic or semantic nature.

Syntactic redundancy results from the implicit knowledge that every speaker has on the statistical properties of his/her language, while semantic redundancy results from the world knowledge of a communicating person, who is able to interpret each message within a meaningful situational context.

The aspect of redundancy in language is closely related to its information content. This has been already observed by Claude Shannon in his seminal work on information theory [1948] and his later works. In his article from 1951, Shannon measured the amount of redundancy by an approximation of the probability estimate of a symbol at position *n*, given the *n-1* symbols to the left. By augmenting the size of *n*, he could estimate upper and lower bounds for the entropy of the language signal. He thus calculated for English a lower bound of about 1 bit per character, which comes up to a redundancy of app. 75%.

From a theoretical point of view, a word predictor basically plays a (word-based) '*Shannon game*', as defined in [Shannon, 1951]: given a left context of *n-1* symbols, it tries to determine the most probable symbol at position *n*; if it is right, it enters one of the symbols, if not, the user has to provide the following symbol and the game continues.

## 4.2 The base model of *SibyWord*

Our word prediction component, *SibyWord,* predicts the most appropriate words considering the context of the words already written. These words are then displayed in the prediction list (s. Figure 2). When the user selects one of these words, it is automatically inserted (or completed) in the current text.

*SibyWord* is based on a stochastic language model (LM), which estimates the probability of occurrence for any word in the lexicon, according to the 3 previously inserted words (4-gram). We trained three models for French, German and English on newspaper corpora (cf. Table II). Using the *SRI* toolkit [Stolcke, 2002][1] we computed a 4-gram LM over a controlled vocabulary of app. 140,000 words. To deal with unseen word combinations, we used *modified Kneser-Ney discounting* [Goodman, 2001] as a smoothing method, and we applied *Stolcke* pruning [Stolcke, 1998] to reduce the model to a manageable size (threshold $\theta = 10^{-7}$).

| **Corpus** | **Language** | **Training size**<br>(number. of words) | **Vocabulary**<br>(number of words) |
|---|---|---|---|
| *Le Monde*<br>(1998-1999) | French | 44,000,000 | 141,078 |
| *Tageszeitung*<br>(1997-1999) | German | 37,000,000 | 141,242 |
| *The Guardian*<br>(1997-1998) | English | 49,000,000 | 133,558 |

**Table II. Word prediction model: training data**

[1] SRI Toolkit: www.speech.sri.com.

Suppose that the user wants to compose the following sentence: “*Most children like ice cream*”. Once the first two words are composed, SIBYLLE proposes the following prediction list: *in*, *and*, *are*, *who* and *with* (s. Figure 7). All of these proposals are syntactically correct, but none of them corresponds to the intended word.

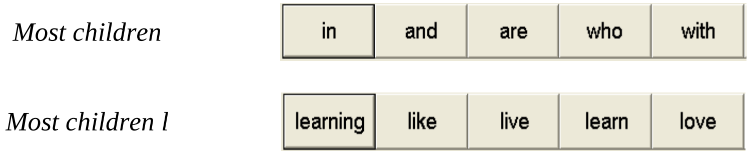


**Fig. 8. Successive prediction lists of SibyWord (5-word prediction list in horizontal mode)**

Then, the user selects the first letter of *like*. The system now filters out all words not starting with ‘*l*’, the intended word *like* appears in second position in the list, and it can be selected by the user.

It can be assumed that a word appearing in the list and not being selected right away is not intended; even though it still matches the given beginning after insertion of another character, it can be filtered out to leave place for other words. This filtering strategy enhances keystroke savings (s. section 6.1), but it is of course risky, since the user might have missed the intended word in the list and then has to insert all its characters. The degree of helpfulness of this strategy depends on the user’s cognitive abilities. For example, persons with visual or memory impairments might often miss a word in the prediction list. For this reason the filtering of already shown words can be switched off.

First, experiments have shown that our baseline model is able to save more than half of the keystrokes (based on user emulation). However, language models are highly dependent on their training resources. The performance of a language model, trained on newspaper text, will significantly decrease in real usage (normally users do not speak the way newspaper journalists write). We have conducted several experiments to assess this potential degradation [Wandmacher and Antoine, 2006]; similar observations have been made by Trnka and McCoy [2007]. They show that the performance of a word predictor decreases up to 30%, when the language style of the test corpus is very different from the training data. However, large training corpora being similar to the language style of AAC users do not (yet) exist. Newspaper corpora, which are easily available in large quantities, represent here a (surely non-optimal) compromise between language generality and data abundance.

Besides, 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. Trost et al.

2005; Trnka et al. 2006, Trnka and McCoy, 2007]. Whereas these works only consider user adaptation, we have now investigated two kinds of adaptation:

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

In the following section we present the different adaptation techniques that we have implemented in *SibyWord*.

## 5. ADAPTATION TECHNIQUES

### 5.1 User Adaptation: Dynamic User Model

User adaptation is achieved by the integration of two language models: a large base model (4-gram), trained on a *newspaper* corpus and a dynamic user model (DUM), a trigram model which is trained on every text composed by the user; words which are not yet in the general vocabulary (out-of-vocabulary words, OOV) are integrated to the model, as well as user-defined abbreviations (s. section 2.2). The abbreviations can then be directly selected from the word prediction list and are expanded in the text without any further effort.

The base LM reflects the general language, while the DUM adapts the latter to the specific style and vocabulary of the user. The global probability $P'(w_i)$ for a word $w_i$ is estimated by linear interpolation of the two models:

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

where $\lambda_1$, $\lambda_2$ are weighting coefficients ($\lambda_1 + \lambda_2 = 1$). 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. Jelinek, 1990]. It is obvious that this kind of user-sensitive training does not lead to immediate improvements of the predictor; it is long-term adaptation.

From a practical point of view, three alternative strategies are proposed to the user:

- *No adaptation*: the DUM is not activated. There is no learning on the messages composed by the user.
- *Implicit adaptation*: the messages composed by the user are systematically used for training the DUM
- *Explicit adaptation*: every time the user wants to exit the system, he or she is asked whether the text of the current session is to be used for training.

Such a configurable strategy of adaptation is of first importance, since users may have additional cognitive disorders or simply be young children with restricted linguistic knowledge. Their messages can therefore present a high rate of spelling or grammatical errors. It is not clear whether the DUM should integrate these erroneous productions or not: some users favor communication speed and do not care whether the messages are grammatically correct, provided their errors do not disturb the speech synthesis. On the contrary, in an educational setting, teachers and mainly practitioners (e.g. speech therapists) may insist on the correctness of the user's productions.

### 5.2 Semantic adaptation

Several works have already investigated the question of short-term adaptation of language models. In particular, *cache* (or *recency promotion*) models [Kuhn and De Mori, 1990] provide a simple adaptation to the currently typed text by enhancing the probability of recently inserted words. The underlying idea here is that words that have already occurred in a text are more likely to occur another time. Therefore their probability is raised by a constant or an exponentially decaying factor [Clarkson and Robinson, 1997], depending on the position of the element in the cache. The idea of a decaying cache function is that the probability of reoccurrence depends on the distance between the word in the cache and the word to be predicted. The highest probability of reoccurrence is usually after 15 to 20 words and it decreases to normal after app. 1,000 words.

Cache-based models have shown to bring slight but constant gains for keystroke savings [Wandmacher and Antoine, 2006]. Here, we investigate a more sophisticated solution for achieving a context-sensitive adaptation: it can normally be assumed that a text to be entered focuses on some topic; therefore words that are semantically related to this topic should have a higher probability of occurrence, whereas unrelated words should receive a lower probability.

Explicit topic adaptation tries to determine the current topic of conversation and then to select the most adequate model for prediction. This strategy however has hardly proved to be efficient, mainly due to the difficulty in achieving robust topic detection [Bigi et al., 2001]. Our aim is, therefore, to achieve an implicit topic adaptation by taking the semantic similarity between a word and the semantic field of the context into account. The semantic field is not rigidly coupled to a certain topic, it evolves smoothly with the development of the context; therefore explicit topic detection is not needed.

### *5.2.1 Latent Semantic Analysis*

Several works have suggested the use of *Latent Semantic Analysis* (*LSA*) in order to integrate semantic similarity to a language model [Bellegarda, 1997; Coccaro and Jurafsky, 1998]. *Latent Semantic Analysis* [Deerwester et al., 1990] is a technique that models semantic similarity based on co-occurrence distributions of words. *LSA*, which is founded on cognitive motivations [Landauer et al., 1997], 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. Since it does not take word order into account ("bag-of-words" model), it is however very poor at predicting their actual position within the sentence, and it is completely useless for the prediction of grammatical words (e.g. '*of*', '*the*', '*to*').

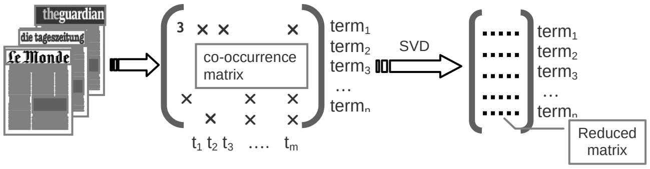


**Fig. 9. Schematic process of LSA training**

From a formal point of view, *LSA* is based on the vector space model of information retrieval [Salton and McGill, 1983]. At first, a given training corpus is transformed into a term × context matrix, displaying the occurrences of each word in each context (Figure 8). A context can be a word window, a sentence, a paragraph or a full text. For *LSA*, a paragraph window is normally assumed. The decisive step in the *LSA* process is then a *Singular Value Decomposition* (*SVD*) of the weighted matrix. Thereby an original matrix *A* is decomposed as follows:

$$\mathrm{SVD}(A) = U \cdot \sum \cdot V^{\mathrm{T}} \tag{3}$$

Matrices *U* and *V* consist of the eigenvectors of the columns and rows of *A*. $\sum$ is a diagonal matrix containing the singular values of *A* in descending order. By only keeping the *k* strongest (*k* usually being around 300) singular values and multiplying $\sum_{\mathrm{k}}$ with either *U* or *V*, one can construct a so-called semantic space for the terms or the contexts, respectively. Each word $w_i$ is then represented as a vector of *k* dimensions, whose distance to others can be compared by a standard vector distance measure. In most *LSA* approaches the cosine measure is used. By calculating the cosine of the angle between one term vector and all the others, a ranked list of next neighbors can be obtained for a given word. From the *LSA* point of view, these neighbors should be semantically related to the word.

### *5.2.2 Building an LSA-based language model*

How can the semantic information as provided by LSA be used for prediction purposes? As already explained, LSA offers a convenient way to calculate the semantic distance between words being represented as vectors in a high-dimensional space. This also extends to phrases, paragraphs or even full documents: every textual element can be represented and compared within the same vector space, simply by calculating the sum of the vectors of the words it contains. We can, therefore, represent our given current context or history $h$ (= $w_1$, ... , $w_m$) by the (normalized) sum of the vectors corresponding to the words the history contains [Landauer et al., 1997]:

$$\vec{h} = \sum_{i=1}^{m} \vec{w}_i \tag{4}$$

This context vector has the same dimensionality as the term vectors. It can be compared to the term vectors by any vector similarity measure. For our AAC application, we make the assumption that an utterance or a text to be entered by the user is usually semantically cohesive. We then expect all word vectors to be close to the current context vector. This forms the basis for a simple (pseudo-) probabilistic model based on *LSA*: after calculating the 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 $\gamma$ is applied [Coccaro and Jurafsky, 1998], which raises the cosine to some power ($\gamma$ is normally between 3 and 8; we got best results with $\gamma = 4$). After normalization we obtain a probability-like distribution, which can be used for prediction purposes. It is calculated as follows:

$$P_{LSA}(w_i|h) = \frac{\left(\cos(\vec{w}_i, \vec{h}) - \cos_{\min}(\vec{h})\right)^{\gamma}}{\sum_k \left(\cos(\vec{w}_k, \vec{h}) - \cos_{\min}(\vec{h})\right)^{\gamma}} \tag{5}$$

where $w_i$ is a word in the vocabulary, $h$ is the current context (history) and $\cos_{\min}(\vec{h})$ returns the lowest cosine value measured for $\vec{h}$ with all present word vectors). The denominator then normalizes the similarity values to ensure that they sum up to 1.

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

Ex. 1 *The game was nearly over when the ball*

Table III shows the ten words that present the highest LSA probabilities with the context vector corresponding to Example 1: all ten predicted words are semantically related to the context, they should, therefore, be given a higher probability.

| **Rank** | **Term** | $\mathbf{P_{LSA}}$ | **Rank** | **Term** | $\mathbf{P_{LSA}}$ |
|---|---|---|---|---|---|
| 1 | game | 0,0191 | 6 | upfield | 0,0081 |
| 2 | kick | 0,0189 | 7 | volley | 0,0076 |
| 3 | offside | 0,0113 | 8 | touchline | 0,0045 |
| 4 | pass | 0,0112 | 9 | referee | 0,0036 |
| 5 | tackles | 0,0081 | 10 | pitch | 0,0033 |

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

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 ($0 \le \lambda_1 \le 1$).

$$P'(w_i) = \frac{P_b(w_i)^{\lambda_1} \cdot P_s(w_i)^{(1-\lambda_1)}}{\sum_{j=1}^{n} P_b(w_j)^{\lambda_1} \cdot P_s(w_j)^{(1-\lambda_1)}} \qquad (6)$$

In our case, geometric interpolation gives better results [cf. Wandmacher and Antoine, 2007a], 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 be 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 [1998] proposed an entropy-related confidence measure, based on the assumption that words occurring in many different contexts (i.e. have a high entropy), cannot be well predicted by *LSA*. Measuring relation quality in an *LSA* space, Wandmacher [2005] showed, however, 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). Instead he found 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 and thus are more probable to be correctly predicted. We define the density of a term $w_i$ as follows:

$$D_m(w_i) = \frac{1}{m} \cdot \sum_{j=1}^{m} \cos(\vec{w}_i, NN_j(\vec{w}_i)) \qquad (7)$$

In the following we will use this measure (with *m*=100) as a confidence metric to estimate the reliability of a word being predicted by the LSA component. To be used as an interpolation coefficient, $D_m$ $(w_i)$ is modified in the following way:

$$\lambda_i = \beta \cdot D(w_i), \text{ iff } D(w_i) > 0; \text{ 0 otherwise} \qquad (8)$$

with $\beta$ being a weighting constant to control the influence of the LSA predictor. For all experiments, we set $\beta$ to 0.4 (i.e. $0 \leq \lambda_i \leq 0.4$), which proved to be optimal here.

For calculating the LSA space, we used the *Infomap* toolkit[2] and generated a term × term co-occurrence matrix for an 80,000 word vocabulary (matrix size = 80,000 × 3,000 keywords), grammatical words were excluded (Table IV). We set the size of the co-occurrence window to ±100, and the matrix was then reduced by singular value decomposition to 150 columns. Table IV lists the training corpora that we used for the calculation of the LSA space.

| **Corpus** | **Language** | **Training size**<br>(nb. of words) |
|---|---|---|
| *Le Monde* (1989-98) | French | 100,000,000 |
| *Die Tageszeitung* (1989-1999) | German | 101,000,000 |
| *The Times* & *The Guardian* (93-98) | English | 108,000,000 |

**Table IV. LSA model: training data**

### *5.2.3 Semantic adaptation: related work*

A number of approaches have tried to adapt a word predictor to the current semantic context. On the one hand, there are methods like the ones described in [Trost et al., 2005] and [Li and Hirst, 2005] that make use of the *trigger* model, as presented by Rosenfeld [1996].

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

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 "*penalty*" are much more likely to appear. The trigger-target pairs are usually calculated by collocation measures (such as *Point-Wise Mutual Information*, cf. Church and Hanks, 1989) from large corpora. Trost et al. [2005] have evaluated such a model for German, however their gains remained modest.

On the other hand, approaches like the one by Trnka et al. [2005] 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 the highest weight to the LM whose topic is closest to the current topic of discourse. This model seems to yield rather good results. However, one of its drawbacks is the need for topically assigned corpora. Such corpora exist for English (e.g. the *Switchboard* corpus), but they are not (yet) available for other languages such as German or French.

### 5.3 Treatment of compound words

When it comes to prediction purposes, German is a rather difficult language. It has a complex morphology comprising three genders (masculine, feminine, neuter) and four noun cases (nominative, genitive, accusative, dative), which multiplies the number of possible inflected word forms. The principal problem, however, is the treatment of German compound words, which are realized as a single orthographic unit. The formation process is productive (i.e. the number of possible words formed in this way is infinite), and it can lead to words of sometimes astonishing length. This characteristic can also be found in other Germanic languages such as Dutch or Swedish. The examples in Table V illustrate the productivity of the formation. The frequencies for these compounds have been determined on a corpus containing more than 120 million words, and they clearly show the problem: even though the last three words represent well-formed compounds, they are very unlikely to be found in a corpus of any size.

| **Word / Compound** | **Frequency** |
|---|---:|
| *Wort* 'word' | 39,241 |
| *Wortvorhersage* 'word prediction' | 0 |
| *Wortvorhersagemodul* 'word prediction module' | 0 |
| *Wortvorhersagemodulentwicklung* 'development of word prediction modules' | 0 |

**Table V. Formation of German compound words and their frequencies in a 120 million German newspaper corpus**

Baroni et al. [2002] have analyzed a large German newswire corpus (APA), and they found that nearly half of the unique words (types) in that corpus were compounds. Most of them had a very low frequency, a big part actually occurred only once. Since even large predefined lexicons normally do not cover such words, they cannot be predicted. Moreover, since they are usually rather long, their negative impact on prediction performance is rather significant, when no further means is given.

To deal with this problem for word prediction, Baroni et al. present an adapted prediction model (*split compound model*). It considers the internal morphological structure of compounds, which are analyzed into a head and a modifier part. For example, a noun-noun compound like *Polizeikontrolle* (‘police control’) is split into *Polizei* (modifier) and *Kontrolle* (head), and each part is then predicted separately. The gains of this complex model remain however very low.

We have, therefore, opted for a different strategy, which is very simple: our *partial selection* (PS) method allows for the selecting of each part of a compound and agglutinating it to the former part by entering a backspace after selection. This alone however would not be sufficient, because sometimes two parts are joined by a so-called *joint morpheme* (e.g. *Hund***e***nase* ‘dog-*e*-nose’ or *Verein***s***sitzung* ‘club-*s*-reunion’). Therefore, our method allows a person to enter one of these morphemes (‘-s-‘, ‘-e-‘, ‘-en-‘, ‘-es-‘, ‘-er-‘) after a compound part has been selected.

## 6. RESULTS

### 6.1 Objective evaluation: keystroke saving rate

It is difficult to assess objectively how a word predictor can really speed up communication rates. Indeed, the observed improvements strongly depend on the user, and on the interaction between the prediction component and the user interface. As a result, the evaluation of an AAC system should be considered along several perspectives, such as:

- usability of the user interface,
- performance of the word prediction component,
- environmental evaluation of the complete system, in order to assess how the different components of the system interact.

In this section, we will only concentrate on the performances of the text prediction component. Two kinds of objective evaluation related to prediction are reported in the literature [Garay-Vitoria and Abascal, 2006):

- *Empirical evaluation* (*human testing*): based on the observations and the typing speed of several users

- *User emulation*: an emulation module enters a test corpus using the word predictor and thereby calculates a standardized evaluation measure.

The pros and cons of these two approaches are well-known. On the one hand, human testing provides results that include the influence of human factors like writing errors, fatigue, learning time etc. But these observations strongly depend on the recruited users which restricts the evaluation to individual case studies.

On the other hand, while emulation is fast and yields a reproducible objective evaluation measure, it completely ignores human factors. It produces only theoretical results which have to be carefully interpreted. In particular, contradictory experiments have clearly shown there is not a direct correlation between objective metrics and speed rate improvement [Anson et al. 2006; Koester and Levine 1994; Koester et Simpson 2000].

Several objective metrics have been proposed to assess the ability of a prediction component to speed up a communication aid. Some of them are directly related to human testing. Soukoreff and MacKenzie [2003] use for instance a *KSPC* (*KeyStroke Per Character*) measure which is a good indicator for the rate of typing errors. Likewise, measures of communication speed [Koester and Levine, 1994] are strongly related to the motor and cognitive abilities of the recruited users. For assessment by emulation, text predictors are traditionally evaluated by an objective measure called *Keystroke Saving Rate* (*ksr*) which is defined as follows:

$$ksr_n = \left(1 - \frac{k_p}{k_a}\right) \cdot 100 \qquad (9)$$

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).

Other metrics can be found in the literature. The *hit rate* (HR) is the percentage of times that the intended word appears in the prediction list. It gives a clear idea of how much the system is able to aid the user. Some experiments have; however, shown that hit rates are correlated to keystroke saving rates [Fazly and Hirst, 2003]. The perplexity measure, which is frequently used to assess statistical language models, proved to be less accurate in this context, particularly when new words were added during the prediction process [Wandmacher and Antoine, 2006]. Thus, we have decided to adopt the *ksr* measure to assess our system.

It is obvious that this evaluation measure directly correlates with the size of the prediction list. The more words presented, the better the *ksr* will be, however the cognitive load on the user rises as well. The following figure shows the *ksr* for prediction list sizes from 1 to 20, tested on a French newspaper corpus (*news-fr*, s. section 6.2). The results are based on the combined model, as described in section 6.7.

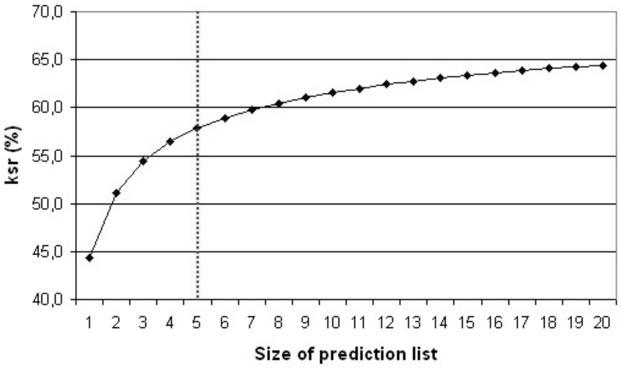


**Fig. 10. Keystroke savings compared to prediction list size**

As the curve in Figure 9 clearly shows, the dependency between *ksr* and list size is non-linear: whereas we gain more than 13% from $ksr_1$ to $ksr_5$, ($ksr_1$=44.4%; $ksr_5$=57.9%) the gain between $ksr_5$ and $ksr_{10}$ is only 3.6% ($ksr_{10}$=61.5%). For this reason, an *n* between 3 and 7 seems a reasonable trade-off between saving rate and cognitive load. This result is coherent with prior works showing that typing speed rates reach a plateau at a word list length of about five words [Swiffin, Arnott, Newell 1987]. In the following section, keystroke savings are estimated with a word prediction list of 5 items ($ksr_5$). This corresponds to the list size that is used in practice in the SIBYLLE system.

As in [Trost et al. 2005] and [Trnka et al. 2007], the keystroke saving rates presented in the paper are based on the assumption that one additional keystroke is required to jump to the word selection list and that a space is automatically inserted afterwards. By default, the computation of the *ksr* supposes that the system follows the strategy of dynamic filtering, presented in section 4: words, which have already occurred in the list and that were not selected by the user, will not reappear after the next character has been inserted. This has a slight but stable effect on the *ksr*. Without filtering of already shown words we measured a $ksr_5$ of 56.9% (-1%) for the above corpus (*news-fr*).

### 6.2 Ecological evaluation: corpora from multiple language registers

*SibyWord* has been assessed for each language (French, German and English) separately. To bring our evaluation closer to real usage, we have conducted experiments on various corpora that correspond to different language registers and topics of communication

| Register | Corpus (name) | Nbr of words |
|---|---|---|
| News | From *L'Humanité* (*news-fr*)<br>From *Süddeutsche Zeitung* (*news-de*)<br>From *The Guardian* (*news-en*) | 58,457<br>56,031<br>53,070 |
| Literature | *Germinal*, by Emile Zola (*lit-fr*)<br>*Effi Briest*, by Theodor Fontane (*lit-de*)<br>*The Picture of Dorian Gray*, by Oscar Wilde (*lit-en*) | 50,251<br>54,844<br>53,640 |
| Transcribed speech | *OTG*[3] (*speech-fr*)<br>*German Verbmobil*[4] (*speech-de*)<br>*English Verbmobil* (*speech-en*) | 15,435<br>20,729<br>20,788 |
| E-mail | French personal e-mails (*email-fr*)<br>German personal e-mails (*email-de*)<br>Mails from the *Enron* e-mail dataset[5] (*email-en*) | 44,946<br>15,774<br>22,151 |

**Table VI. Evaluation corpora used (w. number of 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
- For German only: 4-gram + partial selection (PS)
- 4-gram + DUM + LSA

### 6.3 Baseline prediction model

Table VII presents the performances of the baseline model for the different corpora. Whatever the language considered, the model was trained on news corpora (Table II, section 4). In this control situation (same register as training corpus), the resulting $ksr_5$ varies from 51.6% (German) to 57.8% (French). The lower $ksr_5$ observed for German can be explained by its more complex morphology, as well as the presence of compound words, which are not predictable by the base model (s. section 5.3).

[3] *OTG*: the corpus collects the transcription of spontaneous spoken dialogs between French tourist agents and customers at the tourism office of Grenoble, France (Nicolas et al., 2002).

[4] *Verbmobil* corpus: http://www.phonetik.uni- muenchen.de/ Forschung/Verbmobil/VerbTRL.html

[5] *Enron* e-mail dataset: http://www.cs.cmu.edu/~enron/

| Corpus | French | German | English |
|---|---|---|---|
| News | 57.8% | 51.6% | 55.5% |
| Literature | 46.0% | 44.9% | 49.8% |
| Transcribed speech | 48.3% | 49.1% | 48.5% |
| E-mail | 48.6% | 48.0% | 49.4% |

**Table VII. Performances ($ksr_5$) of the baseline model (4-gram) on different communication situations**

As already mentioned, language models are dependent on their training resources; for this reason it is not astonishing to observe the highest savings for the corpora which are most similar to the training data (newspaper). Corpora of other registers however yield significantly worse results; the literature corpus shows a performance loss of more than 20%. In a real usage situation, even worse results can be expected with this baseline model. Two causes can be invoked to explain this degradation:

- Out-Of-Vocabulary (OOV) words cannot be predicted by the system. (Table VIII presents the percentage of OOV in the French test corpora.)
- Even though all text data can be considered grammatical, every language register exhibits its own particular way of sentence formation and diction. In this respect, newspaper data differs very much from literature of the 19th century and even more from conversational speech, where repetitions and phrasal disruptions are very common.

|  | News | Literature | Speech | E-mail |
|---|---|---|---|---|
| **% of OOV** | 2.2 % | 2.4% | 1.3 % | 5.1% |

**Table VIII. Percentage of out-of-vocabulary words (OOV) in the French test corpora**

### 6.4 Dynamic User Model

Table IX displays the overall results of the combined model and the gains with respect to the baseline results. In the beginning the DUM was always empty. As Table IX shows, we get an important increase of *ksr* for all test corpora. We even get a slight improvement for the test corpus that belongs to the same register (newspaper) as the training data. For the other registers we obtain gains of up to 9.4%. Whichever test corpus considered, the keystroke saving rate remains higher than 50%. Interestingly, the speech corpora get the highest gains for all three languages.

| Corpus | French | German | English |
|---|---|---|---|
| Newspaper<br>*news*-(*fr*,*de*,*en*) | 58.5%<br>(*+0.7%*) | 54.6%<br>(*+3.0%*) | 56.3%<br>(*+0.8%*) |
| Literature<br>*lit*-(*fr*,*de*,*en*) | 50.6%<br>(*+4.6%*) | 50.0%<br>(*+5.1%*) | 53.0%<br>(*+3.2%*) |
| Transcr. speech<br>*speech*-(*fr*,*de*,*en*) | 57.7%<br>(*+9.4%*) | 57.5%<br>(*+8.4%*) | 56.9%<br>(*+8.4%*) |
| E-mail<br>*email*-(*fr*,*de*,*en*) | 53.0%<br>(*+4.4%*) | 51.6%<br>(*+3.6%*) | 54.1%<br>(*+4.7%*) |

**Table IX. Performances ($ksr_5$) of the dynamic user model. Improvement over the baseline**

This is probably due to the strong difference of the language style and also to the high number of repetitive phrases (e.g. "*See you later*", "*I don't know*"), which can be predicted very easily, once they have been integrated one time.

To assess the temporal flexibility of the DUM, we also observed the *ksr* development during the prediction of the test data. As the learning curves in Figure 10 show (for *lit-fr* and *email-fr*), the DUM-based model already performs 2% better than the baseline after only 2,000 words, and it reaches a plateau of +5-6% after approximately 20,000 words. This implies that the training time needed to exploit the advantages of the DUM-based model is not very long (3 to 6 hours of typing for a skilled user).

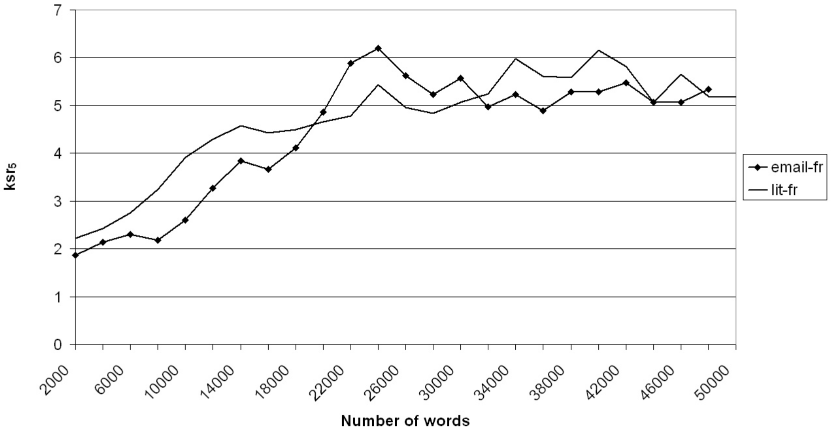


**Fig. 11. Influence of the Dynamic User Model: $ksr_5$ increase according to the amount of training data**

## 6.5 Semantic adaptation

We also evaluated our LSA-based semantic component against our 4-gram baseline model. Table X shows the gains achieved with a combined model, using confidence-weighted geometric interpolation (as described in section 5.2). The semantic adaptation that is achieved by the LSA model leads to a less important increase of *ksr* (+1.0% to +1.7%) than the user model. However, we can conclude that the LSA-based model is beneficial for all test corpora and languages, and the gain is on average five times higher than that of a cache model [cf. Wandmacher and Antoine, 2006]. Moreover, it performs far better than the trigger model, as used by Trost et al. [2005] (+0.3% in $ksr_5$) or the topic model by Trnka [2006] (+0.4% with respect to a trigram baseline).

| **Corpus** | **French** | **German** | **English** |
|---|---|---|---|
| Newspaper<br>*news*-(*fr,de,en*) | 58.9%<br>(+*1.1%*) | 52.6%<br>(+*1.0%*) | 56.7%<br>(+*1.2%*) |
| Literature<br>*lit*-(*fr,de,en*) | 47.7%<br>(+*1.7%*) | 46.1%<br>(+*1.2%*) | 51.0 %<br>(+*1.2%*) |
| Transcr. speech<br>*speech*-(*fr,de,en*) | 49.5%<br>(+*1.2%*) | 50.4%<br>(+*1.3%*) | 49.5%<br>(+*1.0%*) |
| E-mail<br>*email*-(*fr,de,en*) | 50.2%<br>(+*1.6%*) | 49.1%<br>(+*1.1%*) | 50.7%<br>(+*1.3%*) |

**Table X. Performances ($ksr_5$) of the LSA-based model.**
**Improvement with the baseline model**

An aspect that the results here do not show is the subjective improvement for the users. Since the LSA-based model is able to semantically relate the words in the prediction list to the current context, our LSA-based model also serves as a sort of thesaurus and helps the user to find the appropriate word. This cognitive support can turn out to be much more important than a gain in *ksr*.

### 6.6 Partial selection (Compound treatment)

The emulation of the partial selection method is not as easy to achieve as that of the other methods. It presumes that the user applies an optimal selection strategy which in practice is more difficult than simply scanning a prediction list to see if a word matches. Saving rates can even decrease when simply every word onset is matched, because it then takes two more selection steps to choose the following element (1 back step + 1 selection). The results in Table XI display the optimal gains, i.e. PS was only applied when it could decrease the number of keystrokes to be typed. Since partial selection is mostly useful in handling the insertion of German compound words, we only display the results for German here. Interestingly, partial selection also seems to have a slightly beneficial effect for French and English corpora (+0.1 to +0.3%).

For the partial selection method we can observe stable gains of 0.8 to 1.5% for all corpora. This is somewhat less than the results of Trost et al. [2005] who report higher gains of app. 3% for an equivalent strategy, but still a significant improvement. We can, therefore, conclude that, even though this approach is very simple, it has a beneficial effect on the problem of compound words, and it performs significantly better than the complex model proposed by Baroni et al. [2002] who report an improvement of +0.3%.

| Corpus | PS off | PS on |
|---|---|---|
| *news-de* | 51.6% | 53.1% (*+1.5%*) |
| *lit-de* | 44.9% | 46.1% (*+1.2%*) |
| *speech-de* | 49.1% | 50.0% (*+0.9%*) |
| *email-de* | 48.0% | 48.8% (*+0.8%*) |

**Table XI. Performances ($ksr_5$) of the Partial Selection (PS) strategy. Improvement with the baseline model**

## 6.7 Combining strategies

So far we have presented the results for each adaptation method separately, and we have observed significant and mostly stable gains for all of them. However, this does not imply that these strategies work well together. Therefore, the following table shows the overall results with all strategies combined.

| Corpus | All off (Baseline) | All on (SibyWord) | Gain |
|---|---|---|---|
| *news-fr* | 57.8% | 59.4% | +1.6 |
| *lit-fr* | 46.0% | 52.2% | +6.2 |
| *speech-fr* | 48.3% | 57.9% | +9.6 |
| *email-fr* | 48.6% | 53.8% | +5.2 |
| *news-de* | 51.6% | 56.9% | +5.3 |
| *lit-de* | 44.9% | 51.8% | +6.9 |
| *speech-de* | 49.1% | 58.4% | +9.3 |
| *email-de* | 48.0% | 53.1% | +5.1 |
| *news-en* | 55.5% | 57.6% | +2.1 |
| *lit-en* | 49.8% | 54.4% | +4.6 |
| *speech-en* | 48.5% | 57.7% | +9.2 |
| *email-en* | 49.4% | 54.8% | +5.4 |

**Table XII. Performances ($ksr_5$) for all corpora and languages tested, with and without all adaptation strategies**

As the results in Table XII indicate, the gains from the different adaptation methods are nearly additive, and they remarkably improve the overall results. With the application of all adaptation methods the keystroke savings remain above 50% for all languages and registers. For the speech corpora especially, important gains of more than 9% can be seen; however, all the other language registers also benefit from the adaptation.

The smallest gains are observed for the newspaper corpora; this was expected due to the high similarity with the training data. In general, we can conclude that our adaptive word predictor considerably enhances keystroke savings with respect to an already well performing 4-gram baseline, and it is able to reduce the training dependency innate in statistical NLP approaches. It has theoretically proven high performance for a variety of rather different communication situations and language styles. The practical perspective should now be looked into.

## 7. USER ASSESSMENT

The SIBYLLE system benefits from the experience of seven years of daily use in the rehabilitation center of Kerpape (Brittany, France). This center receives adult patients and children requiring reeducation or rehabilitation care within the framework of a full-time hospital, a day hospital or an outpatient service. The multi-disciplinary team of professionals (physio- and ergotherapists, speech therapists, orthoptists, teachers and technicians) aims to optimize the independence as well as the social and professional reinsertion of its patients.

When a communication-impaired patient arrives at Kerpape, she or he meets all of the interacting staff, who try to determine her or his specific needs by carrying out a number of experiments. The speech therapists analyze the patient’s linguistic abilities and thereby find out which kind of AAC will be most suitable (e. g. use of an iconic, phonetic or alphabetic keyboard). The ergotherapists determine the functional and motor capacities of the patient in order to define the most appropriate input device as well as the selection modes of the AAC system. The orthoptists then analyze the patient’s visual abilities to ensure that all elements of the interface are clearly perceptible. When the basic parameter settings are found, the technical staff then configures the AAC system accordingly; this step is of course performed in close collaboration with the patient.

Such an adaptation process can take a considerable amount of time; especially in the case of visual disability it involves several months of intense work with the patient until the optimal configuration can be found; yet according to the practitioners at Kerpape, it will eventually be found with the SIBYLLE system, due to its far-reaching configurability (cf. section 2.3).

Its successive versions have been used by more than twenty patients[6]. Some of them are adults, but the majority are children and adolescents from the school integrated in the center (s. Table XIII presenting the users from 2005 to 2007).

| User | Age | Disease | Clinical pattern |
|---|---|---|---|
| *Q* | 19 | cerebral palsy | dystonic quadriplegia, anarthria |
| *H* | 15 | encephalitis | dyskinetic quadriplegia, dysarthria + visual impairment |
| *P* | 15 | cerebral palsy | dystonic quadriplegia, anarthria |
| *M* | 15 | cerebral palsy | spastic quadriplegia + amblyopia |
| *E* | 14 | cerebral palsy | dystonic quadriplegia, anarthria |
| *G* | 19 | cerebral palsy | dystonic quadriplegia, anarthria |
| *S* | 23 | cerebral palsy | dystonic quadriplegia, anarthria |
| *Y* | 21 | cerebral palsy | dystonic quadriplegia, anarthria |

**Table XIII. Clinical description of SIBYLLE users in the Kerpape rehabilitation center during the years 2005-2007**

The system was highly appreciated by most users[7]; only two of them, who are visually strongly impaired, felt uncomfortable with the dynamic rearrangement of the keyboard. But even in these severe cases the practitioners could configure the system in a way (i.e. by selecting a static keyboard layout, appropriate colors and font size, and by optimizing the placement of the keypads) that would benefit the users.

This is the particular strength of SIBYLLE. The linguistic facilities of the system are able to evolve with the user's capacities and needs (and not the other way around, as is often the case). A user can start with a very simple static configuration and then successively use more advanced features in order to speed up his/her communication rate without changing the interface. And indeed the teachers of the Kerpape school could observe a significant acceleration of the text insertion process after their students had started to use SIBYLLE. They also observed that the children accept 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. The reduction of the physical fatigue of the users is certainly as important as the improvement of the communication speed [Berard and Neimeijer, 2004].

[6] Note that the results reported in the following are based on earlier versions of SIBYLLE, incorporating a non-adaptive word predictor. The interface properties however (dynamic key selection, configurability etc.) were already part of the initial system.

[7] This statement results from years of work with the users, we did however not yet perform a standardized user inquiry.

Finally, we have also noticed a significant decrease of orthographic and grammatical errors when the patients are using the system. A comparable result has already been observed with users of other AAC systems [Morris et al. 1992 ; Carlberger et al., 1997]. This observation applies in particular when the user has additional language impairments.

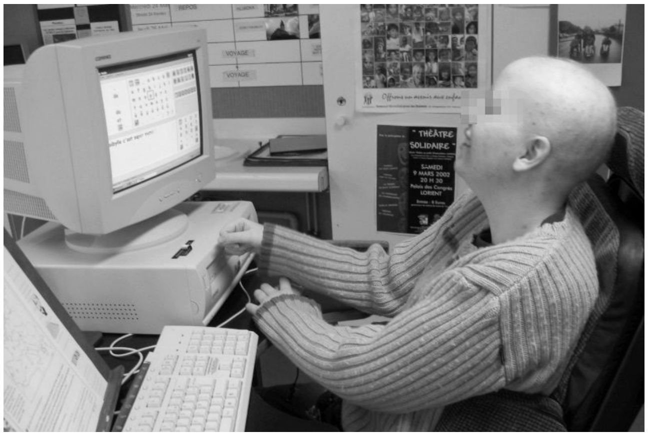


**Fig. 12.** SIBYLLE **(v. 1.5.2) used by an athetosic child from the integrated school of the Kerpape rehabilitation center**

Despite these encouraging user experiences, a disturbing observation is that, frequently, some users do not select the intended word even though it is clearly present in the prediction list. In an experiment conducted with the commercial DIALO system, Biard *et al*. [2006] observed that their patients selected only 2,300 word hypotheses during the composition of text summing up to 80,000 letters (app. 16,500 words). Our discussions with the users and the practitioners tend to show that this situation, which obviously limits the keystroke savings and likewise the communication speed, is due to an already quoted cognitive problem [Koester, Levine, 1994; Horstmann, Levine 1991]: the users have difficulties writing a message and reading the list simultaneously, due to an increase of the cognitive load.

A possible solution to this problem could be to implement direct completion like in the VITIPI system [Boissière and Dours, 2001]: 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. However, this type of immediate display may not be sufficient enough to limit the conflict between input (reading the prediction) and output (writing the message) activities.

Another solution is to directly include word predictions 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, some preliminary experiments suggest that this strategy could be useful when only one or two suggestions are included.

Moreover, one must consider that this selection mode (and direct completion as well) requires a single keystroke, while two successive steps are needed to jump to the word list and to select a word in the ‘‘standard’’ strategy. It should compensate for the fact that fewer hypotheses are proposed to the user. But as we have already pointed out, due to the differing physical preconditions, each user has her or his own preferences and needs; therefore there is no single optimal solution for the interface of an AAC device; only offering a multitude of possible configurations can respond to the various demands of AAC users. For this reason, we are currently implementing the two selection modes mentioned above: direct completion and word selection from the letter keypad.

## 8. CONCLUSION AND PERSPECTIVES

We presented the user interface, as well as the letter and word prediction modules of our AAC system SIBYLLE, which has been in use since 2001. While earlier versions of the system already comprised the interface and letter prediction components, recent development concentrated on improving the word predictor. Therefore, the new features of this module were the main focus of the present work. We have described in detail how the word predictor adapts to the user’s language style and to the current semantic context, and we have presented the results of an extended evaluation (by emulation) of each of the adaptation methods. In the last part, we also reported first results from a real-use evaluation including users from the Kerpape rehabilitation center.

This user-centered evaluation must now be extended. 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 have severe cognitive impairments. This implies the development of sophisticated evaluation measures that are able to consider the individuality of each user while assuring transparency and reproducibility.

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

- all actions of the user (keystrokes, selected items, time stamps),
- all replies/actions of the system, contents of the prediction lists etc.

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 disability. The recordings of these log files are now in progress in the rehabilitation center of Kerpape.

## ACKNOWLEDGMENTS

The authors would like to thank the developers of the *SRI* and the *Infomap* toolkits for making their programs available. This research is partially funded by the UFA (Université Franco-Allemande) and *La Fondation Motrice* (*ESAC_IMC* project).

## REFERENCES

ABRAHAM, M. 1997. Palliation of sensorial and cognitive handicaps of communication: profiling different pictograms for different users. *Assertive Technology Research Series*. Vol. 12. 12-32. IOS Press. 2002.

ALLEN, J. 1992. *Natural Language Understanding*. Benjamins Cummings. Chapter VII.

ALM, N., ARNOTT, J.L., NEWELL, J.F. 1992. Prediction and conversational momentum in an augmentative communication system. Comm. ACM, 35(5). 46-57.

ANSON, D., MOIST, P., PRZYWARA, M., WELLS, H., SAYLOR, H., and MAXIME, H. 2006. The effects of word completion and word prediction on typing rates using on-screen keyboards. *Assistive Technology,* 18(2), 146-154.

BAKER, B. 1982. Minspeak. *Byte*. 9. 186-202.

BARONI, M., MATIASEK, J., and TROST, H. 2002. Wordform- and class-based prediction of the components of German nominal compounds in an AAC system. *Proceedings of the 19th COLING*, Taipei, Taiwan.

BELLEGARDA, J. 1997. A Latent Semantic Analysis Framework for Large-Span Language Modeling. *Proceedings of Eurospeech'97*. Rhodes, Greece.

BERARD, C., and NEIMEIJER, D. 2004. Evaluating effort reduction through different word prediction systems. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, La Haye NL. Vol. 3, 2658-2663.

BIARD, N., DUMAS, C., BOUTEILLE, J., POZZI, D., LOFASO, F., and LAFFONT, I. 2006. 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.

BIGI, B., BRUN, A., HATON, J.-P., SMAILI, K., and ZITOUNI, I. 2001. Dynamic Topic Identification: Towards Combination of Methods, *Proceedings of the RANLP workshop*,.

BOISSIÈRE, P., and DOURS, D. 2001. 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, 895-899.

CARLBERGER, A., CARLBERGER, J., MAGNUSON, T., HUNNICUTT, M.S., PALAZUELOS-CAGIGAS, S., and NAVARRO, S. A. 1997. Profet, a new generation of word prediction : an evaluation study. *Proceedings of NLPCA'97*. Madrid, Spain. 23-28.

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

CLARKSON, P. R., and ROBINSON, A. J. 1997. Language Model Adaptation using Mixtures and an Exponentially Decaying Cache. *Proceedings of the IEEE ICASSP'97*, Munich, Germany.

COCCARO, N. and JURAFSKY, D. 1998. Towards better integration of semantic predictors in statistical language modeling, *Proc. of the Intl. Conf. Spoken Language Processing, ICSLP-98*. Sydney, Australia.

DEERWESTER, S. C., DUMAIS, S., LANDAUER, T., FURNAS, G. and HARSHMAN, R. 1990. Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science, JASIS* 41(6), 391-407.

FAZLY, A., and HIRST, G. 2003. Testing the efficacy of Part-Of-Speech information in word completion. *Proceeding of the Workshop on Language Modeling for Text Entry Methods, 11th EACL*. Budapest, Hungary.

GARAY-VITORIA, N., and ABASCAL, J. 2006. Text prediction systems : a survey. *Univ. Access. Inf. Society*. Vol. 4. 188-203.

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

HORSTMANN, H.M., and LEVINE, S.P. 1991. The Effectiveness of Word Prediction. *Proceedings of 14th RESNA Conference*., 100-102.

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

KOESTER, H., LEVINE, P. 1994. Modelling the speed of text entry with a word prediction interface. *IEEE Trans. Rehab. Eng*. 2(3). 177-187.

KOESTER, H., and SIMPSON, R. C. 1990. Effect of system configuration on user performance with word prediction: results for users with disabilities, *Proceedings of the RESNA 2000 Annual Conference*, Orlando, FL.

KUHN, R. and DE MORI, R. 1990. A Cache-Based Natural Language Model for Speech Reproduction''. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 (6), 570-583.

LANDAUER, T. K., LAHAM, D., REHDER, B., and SCHREINER, M. E. 1997. 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.

LI, J., and HIRST, G. 2005. Semantic knowledge in a word completion task. *Proceedings of the 7 Int. ACM SIGACCESS conference on Computers and Accessibility*, Baltimore, MD, USA. 121-128.

MCCOY, K.F., and DEMASCO, P. 1995. Some applications of natural language processing to the field of augmentative and alternative communication. *Proceedings of the IJCAI'95 Workshop on Developing AI Applications for Disabled People.* Montreal, Canada. 97-112.

MORRIS, C., NEWELL, A., BOOTH, L., RICKETTS, I., and ARNOTT, J. 1992. Syntax PAL: A System to Improve the Written Syntax of Language-Impaired Users. *Assistive Technology*, vol. 4, no. 2, 51-59.

NICOLAS, P., LETELLIER-ZARSHENAS, S., SCHADLE, I., ANTOINE, J.-Y., and CAELEN, J. 2002. Towards a large corpus of spoken dialogue in French that will be freely available: the Parole Publique project. *Proceedings of the 3rd International Conference on Language Resources & Evaluation, LREC'02*, Las Palmas de Gran Canaria, Spain. 649-655.

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

SALTON G., and MCGILL M. 1983.*Introduction to Modern Information Retrieval*. McGraw-Hill, New-York.

SCHADLE, I., Le Pévédic, B., Antoine, J.-Y., and Poirier, F. 2001. SibyLettre, Système de prédiction de lettre pour l'aide à la saisie de texte. In: *Proceedings of TALN'01*, Tours, France.

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

SHANNON, C. E. 1948. A mathematical theory of communication, *Bell System Technical Journal,* vol. 27, 379-423 and 623-656.

SHANNON, C. E. 1951. Prediction and Entropy of Printed English, *Bell System Techn. Journal,* vol. 30, 50-64.

SHIEBER, S.M., and BAKER, E., 2003. Abbreviated Text Input. In: *Proceedings of the International Conference on Intelligent User Interfaces*, Miami, Florida.

SOUKOREFF, R. W., and MACKENZIE, I. S. 2003. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. *Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2003*. New-York, NJ. 113-120.

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

STOLCKE, A. 2002. SRILM - An Extensible Language Modeling Toolkit, *Proceedings of the Intl. Conf. Spoken Language Processing, ICSLP'2002*, Denver, Colorado. vol. 2, 901-904.

SWIFFIN, A.L., ARNOTT, J.L., and NEWELL, A.F. 1987. Adaptive and Predictive Techniques in a Communication Prosthesis, *Augmentative Alternative Communication*. vol. 3, no. 4, 181-191.

TRNKA, K. and MCCOY, K. 2007. Corpus Studies in Word Prediction, Proc. of *ASSETS'07*, Tempe, Arizona.

TRNKA, K., YARRINGTON, D. MCCOY, K. F., and PENNINGTON, C. 2006. Topic Modeling in Fringe Word Prediction for AAC. In *Proceedings of the 2006 International Conference on Intelligent User Interfaces*., Sydney, Australia, 276-278.

TRNKA, K., YARRINGTON, D. MCCOY, K. F. and PENNINGTON, C. 2007. The Effects of Word Prediction on Communication Rate for AAC. *Proceedings of NAACL HLT'2007*, Rochester, 173-176.

TROST, H., MATIASEK, J., and BARONI, M. 2005. The Language Component of the FASTY Text Prediction System, *Applied Artificial Intelligence*, 19(8), 743-781.

VELLA, F., VIGOUROUX, N., and TRUILLET, P. 2005. SOKEYTO: a design and simulation environment of software keyboards. *Proceedings of AAAT'2005*, Lille, France. 723-727.

VENTAGIRI, H.S. 1993. Efficient keyboard layout for sequential access in augmentative and alternative communication *Augmentative Alternative Comm*. 9. 161-167.

WANDMACHER, T. 2005. How semantic is Latent Semantic Analysis?, *Proc. RECITAL'05*, Dourdan, France.

WANDMACHER, T., and ANTOINE, J.-Y. 2006. 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.

WANDMACHER, T., and ANTOINE, J.-Y. 2007. Methods to integrate a language model with semantic information for a word prediction component. In: *Proceedings of EMNLP'07,* Prague, Czech Republic.

WANDMACHER, T., ANTOINE, J.-Y. and POIRIER, F. 2007. SIBYLLE: A system for alternative communication adapting to the context and its user, In: *Proc. of the 9th ACM SIGACCESS Conference on Computers and Accessibility (ASSETS),* Oct 15-17, Tempe, Arizona, USA.

WARD, D., BLACKWELL, A., and MCKAY, D. 2000. 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. 129-137.

WILLIS T., PAIN H., TREWIN S., and CLARK, S. 2002. Informing Flexible Abbreviation Expression for User with Motor Disabilities *Proceedings of ICCHP'02.*