User Linguistic Model Adaptivity for Prediction in AAC Message Composition

Nicola Gatti

Dip. di Elettronica e Informazione,
Politecnico di Milano.
Piazza Leonardo da Vinci 32,
Milano, I-20133
ngatti@elet.polimi.it

Matteo Matteucci

Dip. di Elettronica e Informazione,
Politecnico di Milano.
Piazza Leonardo da Vinci 32,
Milano, I-20133
matteucci@elet.polimi.it

Licia Sbattella

Dip. di Elettronica e Informazione,
Politecnico di Milano.
Piazza Leonardo da Vinci 32,
Milano, I-20133
sbattella@elet.polimi.it

Abstract

Efficient linguistic prediction for AAC aids is a fundamental issue to increase and improve the communication capabilities of verbal disable people. However, although a number of works describing requirements for prediction engines in AAC languages exists, their adaptability to the specific needs of the user is lacking in literature. In this paper, we describe preliminary techniques for the adaptation of the AAC language model to the peculiar characteristics of the user. More precisely, we describe an automatic procedure able to produce a semantic/statistic linguistic model of the user language behavior given the overall AAC language model, the user dictionary, and a *corpus* of sentences produced by the user.

1 Introduction

Nowadays, the improvement and the growth of software application functionalities allow the user to tackle a number of tasks that becomes larger with every new release. However, this increase in application capabilities produces deep repercussions to the development of the interfaces; in fact, the more the tasks have to be tackled deeper the more complex becomes the human-computer interaction.

In literature, a recognized approach to face this complexity in interface design is based on providing the software interface with a sort of "intelligence" (Bunt *et al.*, 2004). More specifically, the interfaces must be *adaptable* and *adaptive*: they have to be customizable for the specific user (i.e., adaptability) and they should autonomously adapt to the user needs at runtime (i.e., adaptation). Thus, the software interface requires a model of the user that must be *ad-hoc* customized and tuned at runtime. This requirements are particularly evident when an user suffers disabilities and the interaction with the computer is strongly corrupted by them. In this paper we focus mainly on user model adaptability.

In particular, our research interest regards human-computer interaction for people suffering verbal impairments – both physical and mental – and using communication software aids (Gatti *et al.*, 2004). Currently, verbal impaired people communicate adopting paper tables of symbols based on *ad-hoc* languages (namely, AAC – Augmented and Alternative Communication – languages (ISAAC, 2004)) on which they sequentially point the symbols to compose sentences (an example of AAC symbol table is showed in Fig. 1). Only recently, software aids have been developed to support and facilitate their communication; however, the design of the interface for symbols selection is currently a crucial aspect.

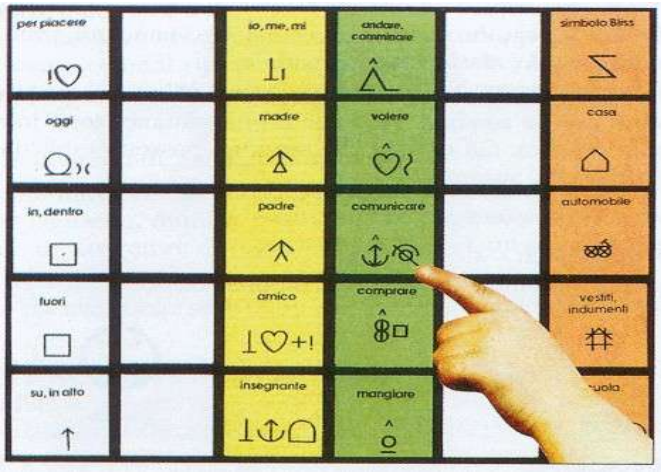

Figure 1: An example of AAC table

Disables usually exhibit mental impairments and/or physical disorders that oblige them to use very simple input devices requiring thus long time to compose sentences. By using traditional communication aids, the composition of a simple sentence can take up to ten minutes. Such a long time becomes frustrating for them and represents an obstacle for their social integration.

In the perspective of improving and simplifying the interaction of the disable with communication software (Higginbotham *et al.*, 1998), we have previously developed a composition assistant named CABA^{2}L (Composition Assistant for Bulk Augmentative and Alternative Language) to support the user in selecting the symbols with an automatic predictive scansion mechanism (Gatti, Matteucci, 2004). In CABA^{2}L, a few AAC symbols are presented in order of probability according to a Bayesian model of the user language behavior that considers both semantic (Quillian, 1968) and statistic properties (Ghahramani, 2001). As result, the disable is supported in his/her communication process reducing the time spent in message composition. This combination of semantic and statistic approaches to model user linguistic behavior improves prediction performance (Swiffin *et al.*, 1987) by using *ad-hoc* learning techniques (Gatti, Matteucci, 2004).

CABA^{2}L, currently, exploits an AAC language model that cannot be customized into an user specific linguistic model to be loaded as a runtime profile. In this paper we present an improvement of the prediction engine implemented in CABA^{2}L to address two adaptability aspects that we have not explored before: (*a*) the adaptability towards different AAC languages, (*b*) the on-line adaptability of the linguistic model to capture the user linguistic behavior. We note, in fact, that the use of different AAC languages is important to obtain an effective rehabilitation, in fact, while recovering capabilities the disable needs more complex languages.

In this new version of CABA^{2}L's prediction engine, the user linguistic model is generated given: (*a*) the model of a specific AAC language (summarily described as a generic set of symbols grouped in semantic categories connected by semantic networks), (*b*) the user dictionary as a sub-set of the previous one, (*c*) a *corpus* of sentences composed by the user, and based on the symbols belonging to the user dictionary. The prediction engine purges the semantic network describing the specific AAC language dictionary according to the symbols effectively adopted by the user, then it assigns a *residual probability* to symbols belonging to the user dictionary, but not reported in the *corpus*, and computes the probabilities of the user linguistic model as described in the next sessions. The determination of the optimal value of the residual probability governs the optimal trade-off between correct prediction and generalization of the user sentences.

In this paper, we discuss an experimental evaluation of learning error and generalization error for the prediction, and we conceive a few criteria to optimize the linguistic model learning procedure. We have experimentally evaluated the prediction performance according to the specific AAC languages and the changes between languages in collaboration with several Italian institutes for disability. Finally, we describe an on-line adaptivity technique based on the injection of new sentences in the dataset to on-line modify the linguistic model adding/removing semantic categories or symbols, and changing probability values. We discuss preliminary experimental results related to on-line adaptivity in order to identify the parameters that optimize the prediction performance.

The paper is structured as follows. The following section discusses the symbolic prediction for AAC languages and CABA^{2}L. Section 3 describes the training of the linguistic model embedded by CABA^{2}L. Section 4 describes several experimental results, and, finally, Section 5 concludes the paper.

2 Symbolic Prediction in AAC and the CABA^{2}L Model

In order to support vocal disable users suffering motor disorders in composing sentences, AAC communication software aids provide the users with an automatic scansion of the AAC symbol table. A generic scansion mechanism can be summarized as follows: an highlight moves autonomously on an AAC symbol table according to a *scansion strategy*, when the symbol, requested by the user, is highlighted the user selects such symbol activating an *input device* and the communication software adds the selected symbol in the sentence, then the highlight starts to move again to allow the selection of a new symbol. Each scansion mechanism is mainly characterized by a specific input device and a specific scansion strategy.

The input device is chosen according to the residual physical capabilities of the user, and, typically, it determines the speed with which the highlight moves on the table. For example, if the scanning speed is to fast, the user may have not enough time to select the requested symbol. Obviously, the more the residual capabilities of the user the faster the movement of the highlight on the table.

The scansion strategy is chosen in order to reduce the number of highlighted symbols before the user selects the symbol that he/she wants to add to the sentence. Lower the number of highlighted symbols before the selection of the requested symbol and faster the sentence composition. In literature several automatic scansion strategies for AAC communication aids have been developed (Higginbotham *et al.*, 1998). We cite the most employed scansion strategies: *sequential*, *row-column*, *at-subgroups*, and *predictive*. As underlined in (Gatti, Matteucci, 2004) the scansion strategy that offers the best performance, in terms of sentence composition speed, is the predictive one.

The idea, on which the predictive scansion is based, is that the scansion mechanism can exploit a linguistic model of the user and, thus, it can predict what will be the next most probable symbol that an user will request. However, although predictive scansion strategies could assure better performance (Koester, Levine, 1994), the *symbolic prediction* techniques do not find a satisfactory discussion in literature. In fact, numerous predictive techniques have been developed, but they have been applied mainly in alphabetical prediction and are not suitable for symbolic prediction. The main issue with alphabetical predictive techniques, that prevents their use for symbolic prediction, is related to the size of the dictionary of items to be predicted and their composition rules. In fact, alphabetical prediction operates on a limited number (about 20-24) of items, the alphabetic signs, that can be organized in words known *a-priori*. Conversely, symbolic prediction operates on a set of symbols that can be organized in different sequences according to the peculiar user linguistic capabilities.

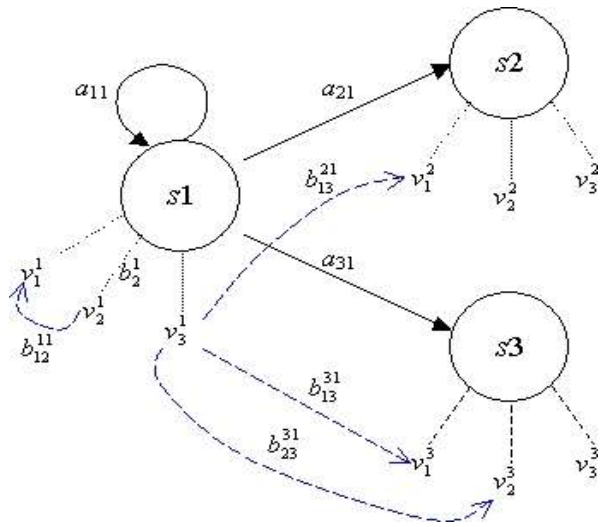

Figure 2: A simple example of DAR-HMM

In (Gatti, Matteucci, 2004) we developed an *ad-hoc* hybrid approach for symbolic prediction applied to a specific AAC language: Bliss (Bliss, 1966). More precisely, we have adopted a *syntactic/semantic approach* to model the AAC language model and we have used this model in order to predict the most probable symbols to be suggested by an automatic scansion system.

We combined the syntactic/semantic model with statistics techniques to take advantage of the AAC language symbols categorization in the generation of a probabilistic model that takes into account for uncertainties in the user language behavior. Symbols have been divided into six syntactic categories according to their grammatical role, and, later, each category has been divided into a number of subcategories adopting the semantic network formalism (Quillian, 1968) modeling the logic connection among two subcategories. This subcategories identification process has been accomplished in collaboration with experts in verbal rehabilitation to obtain a set of subcategories not excessively specific that would have complicated the model without any reason (e.g., we have a substantive subcategory "food" because it connects the verb subcategory "feeding", we have not a substantive subcategory "animal" because it does not connect a specific category). In (Gatti, Matteucci, 2004) subcategories and the number of symbols assigned to each subcategory (note that in the most general case a symbol could belong to more than one category) in the case of Bliss language are reported; however semantic categorization can be accomplished for any AAC languages even if it does not present a strict syntactical structure.

The uncertainties in the user language behavior was addressed developing an *ad-hoc* stochastic model called DAR-HMM, namely, Discrete Auto-Regressive Hidden Markov Model. (An example of DAR-HMM is showed in Fig. 2.) S_i is the *i*-th state and represents the *i*-th semantic sub-category and v^i_j is the *j*-th AAC symbol belonging to the *i*-th sub-category. States in this model are not directly observable; symbols represent the only information that can be observed by the prediction system, and this is the reason for the term "hidden" in the model name. a^i_j represents the probability to transit from the *j*-th state to the *i*-th state (e.g., in our case the probability of selecting a symbol belonging to category verbs after the selection of a symbol belonging to category nouns), b^i_j is the probability to emit the *j*-th symbol from the *i*-th state, and, finally, b^{ij}_{kw} is the probability of emitting the *w*-th symbol of the *j*-th state given the *k*-th symbol of the *i*-th state has been observed as the previous symbol. Finally, we call $\pi_j(t)$ the probability that S_j is the state at time *t*.

Once a DAR-HMM has been trained for a specific AAC language, its functioning is quite simple: given the last emitted symbol by the user and the hidden states probability distribution, it calculates the emission probabilities

of all the symbols keeping into account a^i_j, b^i_j, and b^{ij}_{kw}. Then, the prediction engine makes a list of the most probable symbols and suggest the most probable ones. Formally, called $O(t)$ the predicted symbol at time t, it is produced according to the following formula:

$$O(t) = \underset{v^i_k}{argmax}\left(b^{ij}_{kw} \sum_{j=1}^{N} \pi_j(t-1)\, a^i_j \right)$$

Beside the training aspect that will be described in the next section, the limitations of the model proposed are mainly due to the two following issues: (*a*) being an entire language model, the training process requires a huge *corpus* and cannot be done online (*b*) the model is static and no adaptation is provided to the model in order to follow the user cognitive development. In what follows we first describe the training procedure in the general case (i.e., a complete AAC language model) than we describe an approach to overcome the two limitations discussed above.

3 CABA²L Linguistic Model Identification

In CABA²L, the linguistic model is generated by using the dictionary of a specific AAC language (described as a generic set of symbols grouped in semantic or syntactic categories), and a *corpus* of sentences composed using the symbols in the dictionary.

The obtained DAR-HMM to be used for symbolic prediction can be described using a vector of parameters $\lambda = \langle \Pi°, A, B \rangle$, where $\Pi°[N]$ is the vector of initial subcategory probability $\pi_i(0)$ (i.e., the probability of seeing a certain symbol as the first symbol in the sentence), A[N][N] is the matrix with subcategory transition probabilities a^i_j, and B[N][M][M+1] is the emission matrix with symbol probabilities[1] b^i_j, and b^{ij}_{kw}. In CABA²L, λ vector is estimated using the *corpus* of sentences and a variation of the Baum-Welch algorithm (Rabiner, 1989), an iterative algorithm based on the Expectation-Maximization method (Bilmes, 1998), adapting this technique to the specific case of DAR-HMM (see Fig. 3).

Since the Baum-Welch algorithm is a greedy algorithm that can be trapped in local minima, the initialization estimate of λ parameter vector is a fundamental aspect. In literature a theoretical solution that addresses such issue does not exist; in practice, the adoption of a random or uniform distributed initialization for A and $\Pi°$ has been verified to be adequate. In particular we adopt an uniform distribution as initial estimate for $\Pi°$, and a distribution based on the knowledge about the phenomenon for A. Only arcs connecting subcategories in the semantic model of the language should have a probability $a^i_j \neq 0$. However, we have assigned to the arcs between symbols and states that are not connected in the semantic network a very low probability, not to preclude the training algorithm to eventually discover unforeseen correlations. The initial estimation for the B matrix is more critical so we have used the *Segmental k-Means* (Juang, Rabiner, 1990) technique to obtain a more confidential estimate. Such process considers a sub set of sentences composing the dataset, and, for each one, it looks for the best sequence of subcategories using the Viterbi algorithm (Rabiner, 1989) to upgrades the symbols emission probabilities.

---

[1] From an implementation point of view matrix B could represent the main issue of this model (i.e., with N=30 subcategories and M ~ 2000 symbols the cells number amount is of the order of 10^8, about 400MBytes). However B can be considered a sparse matrix since from each subcategories only a part of symbols can be emitted, so the cells number is, approximately, lower than 10^4 and *ad-hoc* data structure such as *heap* or *priority queue* with optimized algorithms can be used to overcame memory occupancy and speed access issues.

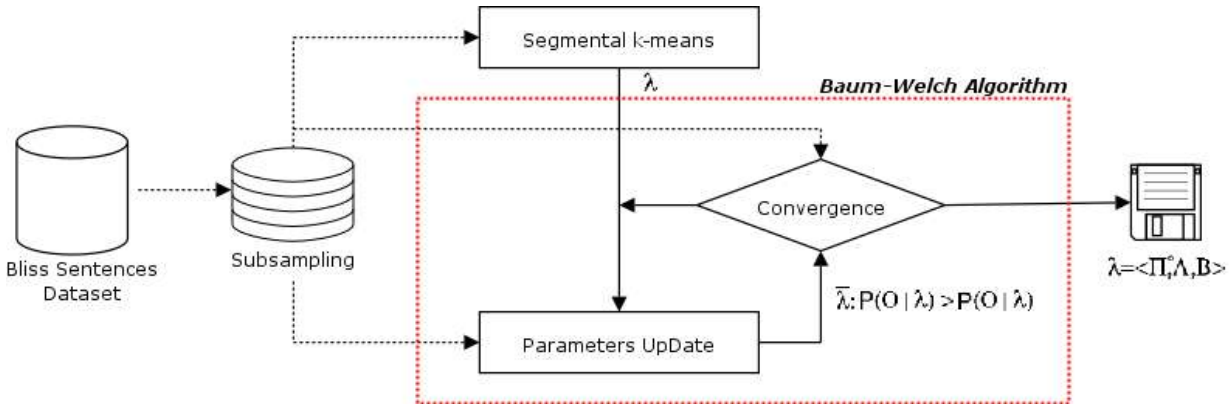

Figure 3: The training process

Given the initial values λ^0 for the model parameters, we use a modified Baum-Welch algorithm to estimate, from a real dataset, the model parameters through a sequence of temporary $\underline{\lambda}$ model parameters. As in any learning algorithm, the main issue is avoiding the *overfitting* phenomenon (Caruana *et al.*, 2001), so we stopped the training phase according to the generalization error (i.e., the error on new samples) and not just observing the training error (i.e., the error on the training set). In CABA²L , to estimate the generalization error we used the *K-fold cross-validation* technique (Amari *et al.*, 1995); it consists in dividing the whole set of sentences into *K* subsets of the same dimension and to use at each iteration *K*-1 subsets for parameter estimation and the remaining validation set is used to valuate the convergence of model generalization error. In other words, we calculate the errors of the model in predicting the sentences of the validation set it has never seen, and we analysed the validation error function during training iterations of the Baum-Welch algorithm until it reached its minimum.

As introduced in the previous section, the approach proposed up to now tries to build a language model for the specific AAC language and not for a specific user. In doing this we obtain good generalization for prediction, but we might miss the specific user language model incurring in an underfitting problem. Moreover the *corpus* needed to estimate the probabilities in the model has to be very large to reflect real correlations among the possibly many unused symbols in the language. To improve the specificity of the model and, at the same time, reduce the *corpus* required to train the predictor we use a dictionary from the user and using this we adapt the model learned from the language model to fit the specific user. In fact, the prediction engine purges the syntactic and semantic categories in the AAC language dictionary according to the symbols effectively adopted by the user in his/her dictionary, then it assigns a residual probability to symbols belonging to the user dictionary, but not reported in the *corpus* provided, and thus computes the probabilities of the linguistic model.

In order to make the DAR-HMM production procedure independent from any AAC language model, we employ the following protocol. We define an AAC language model in a XML file capturing: all the syntactic categories of the language, the semantic subcategories for each category, all the symbols belonging to the respective subcategories, and the sparse matrices A, B, and Π°. We define the user dictionary in a XML file as a list of symbols adopted by the user, and the sentence *corpus* of the user in a XML file as a list of sentences. The adoption of XML formalism is motivated to make the DAR-HMM independent from a specific AAC language. Summarily, given a AAC language model and the user dictionary, the procedure purges the symbols not adopted by user from the semantic network described in the AAC language model and scales the probability matrices. Then, the training algorithm is applied to make the DAR-HMM adapted to the user characteristics.

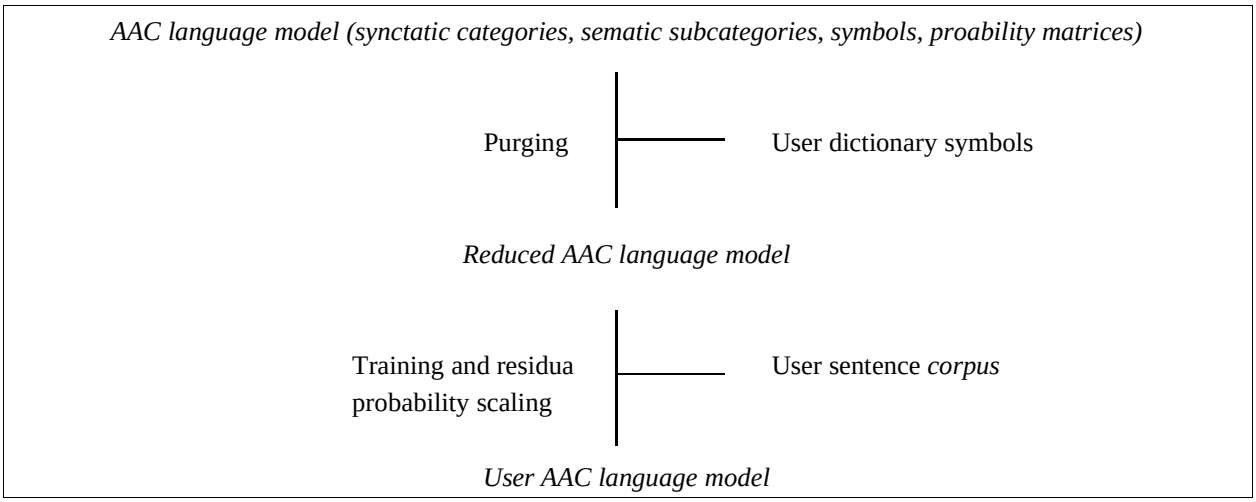

Figure 4: The language model production procedure

The λ parameters vector has to be estimated now by considering only the symbols remaining in the user model and not all the symbols in the language, moreover it could be possible to have symbols in the user model not present in the reduced *corpus*; to compensate for the latter problem, a small fraction of cumulative probability α is assigned to all the symbols that do not appear in the *corpus*. To initialize the A matrix we take for each of the symbol in the dataset its subcategory and we estimate the transition probability to that state (i.e., subcategory) by observing the subcategory of all the states preceding it in the *corpus*. This probability mass is computed for all the states for which exists a symbol in the *corpus* and rescaled by $(1-\alpha)$. Given each state S_i that appears into the user sentence *corpus* we connect that with all the residual states feasible for the reduced AAC language model and we assign a cumulative probability α to all the residual states. More precisely, called n_i the number of residual states for state S_i, the probability to transit from state S_i to each residual state is α/n_i and the transition probability computed by the procedure previously described are scaled by the factor $(1-\alpha)$. The same procedure for residual probability is applied to B and $\Pi°$.

Finally, the model produced by the procedure previously described may be affected by an eccessive purging, or, better, the user can improve his/her linguistic skill by extending his/her dictionary. This calls for on-line adaptability of the DAR-HMM structure and probabilities. In order to address model structure adaptability we repeat the training procedure each time a new symbol is added to the user dictionary (the overview of the model production procedure is showed in Fig. 4).

4 Experimental Evaluation

DAR-HMM has been implemented in CABA^{2}L and, finally, integrated in Bliss200, a communication software centered on Bliss language. CABA^{2}L receives from Bliss2003 the last selected symbol, calculates the most probable four symbols according to the established requirements, and scans them in an ad-hoc panel of the graphical interface before scanning the full table.

To validate DAR-HMM, we are interested in giving an estimated *training error* and *generalization error* in several user scenarios characterized by symbols, symbols subcategories, user residual linguistic capabilities, and user needs; we are also interested in evaluating the time required both for learning and prediction process. To accomplishing this validation, we have strictly collaborated with two Italian clinics for verbal impairments (PoloH

and SNPI of Crema) evaluating the prediction performance in different scenarios; in this paper we report just two scenarios as the most significant ones:

- a dataset of 20 sentences with 4 sub-categories and 7 symbols representing a verbal impaired person unskilled in Bliss utilization or suffering deep mental deficiency;
- a dataset of 80 sentences with 18 sub-categories and 120 symbols representing a verbal impaired person skilled in Bliss use and without deep mental deficiency.

We have shuffled the sentences of each dataset in order to achieve a homogeneous dataset not affected by time correlation. In addition we have divided each dataset into two parts, respectively 80% of sentences in the first part and 20% of sentences in the second one. We have adopted the first part to training the model computing the training error. We have adopted the second one to evaluate the generalization error.

The training error expresses the effectiveness of the learning and it is obtained comparing the suggestion proposed by CABA^{2}L during the composition of sentences it has learnt. To estimate the correct prediction's probability, we have carried out over 800 simulations where we compare the suggested symbols and the one chosen by the user. In Tab. 1 mean and standard deviation for both the two scenarios are showed, they evidence a training error of about 9.2% for the first scenario and 65.5% for the second one taking into account a number of proposed symbols equals to 4 as suggested by therapist.

Table 1: Training error: probability that the requested symbol is in the first four predicted symbols according to the datasets adopted to train the DAR-HMM

Predictions	Mean (Scenario 1)	Std. Dev. (Scenario 1)	Mean (Scenario 2)	Std. Dev. (Scenario 2)
1 symbol	0.34	0.05	0.25	0.02
2 symbols	0.56	0.07	0.3	0.03
3 symbols	0.78	0.07	0.32	0.03
4 symbols	0.91	0.05	0.34	0.04
Not suggested	0.09	0.06	0.65	0.04

The generalization error expresses the effectiveness of the prediction system, and it is obtained comparing the suggestions proposed by CABA^{2}L during the composition of sentences that exhibit the same probability distribution with respect to the sentences it has learnt, but were not presented to the system during the training phase. To estimate the correct prediction's probability, we have carried out over 200 simulations where we compare the suggested symbols and the one chosen by the user. In Tab. 2 mean and standard deviation for both the two scenarios are showed, they evidence a generalization error of about 11.3% for the first scenario and 64.3% for the second one taking again into account a number of proposed symbols equals to 4 before. The values of mean and standard deviation evaluated in generalization error are very close to the values evaluated in training error, thus DAR-HMM evidences high generalization ability. Although the training and generalization errors are in the second scenario high we are confident to get better result just having a bigger dataset.

Table 2: Estimated generalization error: probability that the requested symbol is in the first four predicted symbols according to the datasets not adopted to train the DAR-HMM

Predictions	Mean (Scenario 1)	Std. Dev. (Scenario 1)	Mean (Scenario 2)	Std. Dev. (Scenario 2)
1 symbol	0.2	0.08	0.18	0.09
2 symbols	0.44	0.15	0.25	0.07
3 symbols	0.67	0.18	0.3	0.07
4 symbols	0.89	0.07	0.36	0.08
Not suggested	0.11	0.07	0.64	0.08

5 Discussions and Conclusions

In this paper we have analyzed the AAC symbols scansion issues for motor disordered persons. In particular, we described an *ad-hoc* prediction model (DAR-HMM) to be used in AAC context and its adaptability to the user characteristics.

We have applied DAR-HMM to the case of Bliss language introducing semantic categories for Bliss symbols. In addition, we integrated CABA^{2}L into Bliss2003 an AAC communication software based on Bliss, and experimentally validated it with real data in collaboration with two Italian clinical centers for verbal impaired people proving its effectiveness for reduction of the time spent to compose Bliss messages. The definition of AAC language models based on a XML formalism and a general training procedure allowed the description of a generic linguistic model of the user. In addition, the adoption of the residual probability allowed a finer generalization of the prediction engine.

Time spent by verbal disables that collaborated with us in order to compose messages using Bliss2003 with respect to the time spent with adoption of a traditional scansion system has been reduced up to 60%. Tests have evidenced that the training phase requires few minutes depending on the size of the dataset and the number of symbols and subcategories, but this does not affect Bliss2003 performance, because it can be run on background. Conversely, these tests have proved that the symbols prediction is immediate (<1~second) and can be performed in real time.

In future the performance of the prediction will be improved refining the prediction model. Moreover we would like to achieve on-line adaptation of the DAR-HMM to the linguistic behavior of the user and to take into account the evolution of the user linguistic capabilities, and to support other AAC languages with respect to Bliss, particularly PCS. Finally we will analyze the learnt semantic/probabilistic model of the linguistic behavior of the user in order to study relationships between disabilities and verbal impairments.

References

Amari, C., Finke, M., Muller, K.R., Murata, N., and Yang, H. (1995). Asymptotic statistical theory of overtraining and cross-validation. *Technical report METR 95-06*. University of Tokio.

Bliss, C.K. (1966). Semantography. *Semantography Blissymbolic Communication*, Sindey, Australia.

Bilmes, J. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Markov Models. *Technical report*, University of California, Berkeley.

Bunt, A., Conati, C., McGrenere, J. (2004). What role can adaptive support play in adaptable system? In the *Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI)*, (pp. 117-124). Madeira, Portugal.

Caruana, R., Lawrence, S., Giles, C.L. (2001). Overfitting in neural network: backpropagation, conjugate gradient, and early stopping. In *Proceedings of Advances in Neural Information Processing Systems*.

Gatti, N., Matteucci, M. (2004). CABA2L a Bliss predictive composition assistant for AAC communication software. In the *Proceedings of the International Conference on Enterprise Information Systems (ICEIS)*, volume 5, (pp. 89-96). Porto, Portugal.

Gatti, N., Matteucci, M., Sbattella, L. (2004). An ICT Aid for Verbal Impaired People: Bliss2003. In J. Klaus, K. Miesenberger, W. L. Zagler and D. Burger (Ed.), *Proceedings of 9th International Conference on Computers Helping People with Special Needs*, (pp. 983-990). Paris, France.

Ghahramani, Z. (2001). An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 1, 1-38.

Higginbotham, D.J., Lesher, G.W., Moulton, B.J. (1998). Techniques for augmenting scanning communication. *Augmentative and Alternative Communication*, 14, 81-101.

ISAAC. Internet site of International Society for Augmentative and Alternative Communication. http://www.isaac-online.org. Last accessed September 2004.

Juang, B., Rabiner, L. (1990). The segmental k-means algorithm for estimating parameters of hidden markov models. *IEEE Transactions on Acoustic Speech and Signal Processing*, 38, 1639-1641.

Koester, B., Levine, S. (1994). Learning and performance of ablebodied individuals using scanning systems with and without word prediction. *Assistive Technology*, 14, 42-62.

Quillian, M. (1968). Semantic memory. In Minsky (Ed.), *Semantic Information Processing*; Cambridge, USA.

Rabiner, L.R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), (pp. 257-286).

Swiffin, A., Pickering, J., Newell, A.F. (1987). Adaptive and predictive techniques in a communication prosthesis. *Augmentative and Alternative Communication*, 3, 181-191.