An evaluation framework for writing assistance systems: Application to VITIPI

Philippe Boissière, Daniel Dours
Institut de **R**echerches en **I**nformatique de **T**oulouse – University Toulouse III – CNRS,
118 Route de Narbonne, 31 062 TOULOUSE Cedex France
{boissier, dours}@irit.fr

Abstract : In this paper we propose an evaluation framework for writing assistance systems. The variety of such systems makes it difficult for potential users to select the most convenient one according to their particular needs. First of all, we wish list the main features of existing writing assistance systems. Then, we will explain the evaluation framework which consists of two parts: The first one deals with systems features, listing all the abilities offered by such systems. The second one tries to gauge the systems efficiency. Going further, the evaluation framework will be used to evaluate the VITIPI system, the evaluation results will be exposed. Finally, the evaluation framework limitations will be pointed out and different ways for improving the evaluation framework quality will be presented.

Résumé : L'objectif de ce papier est de proposer un modèle d'évaluation pour les systèmes d'assistances à l'écriture. De tels systèmes sont très nombreux de par le monde et les utilisateurs potentiels sont très désorientés pour en choisir un qui soit le plus adapté possible à leurs besoins particuliers. Nous dresserons tout d'abord la liste des caractéristiques principales des systèmes existants. Nous expliquerons ensuite le modèle d'évaluation qui est composé de deux parties. La première traite des caractéristiques des systèmes en dressant la liste des fonctionnalités que l'on peut trouver. La seconde essaye de mesurer l'efficacité des systèmes. Le modèle d'évaluation sera explicité en prenant pour exemple le système VITIPI dont nous exposerons les résultats. Nous discuterons sur les limitations de notre modèle d'évaluation. Pour conclure nous verrons comment améliorer notre modèle d'évaluation.

Key Words : Writing assistance systems, Evaluation, Assisted and Augmented Communication (AAC),

Introduction

Writing becomes very important in our society with the use of new coming media like e-mail, SMS,… [1]. For disabled people who are speech troubled, writing is almost the only way of communication. So, it has to be improved. On one hand, a lot of writing assistance systems can be found all around the world [2], with a lot of features and various implementations. On the other hand, each disabled people is a single one with particular needs. So there is a need to make those various features match the user's needs ? Unfortunately, there is no evaluation framework to help the user to select the system, which most satisfies his particular needs.

First of all, we wish list the main features of existing writing assistance systems. Then, we will explain the evaluation framework which consists of two parts : systems features and systems efficiency. Going further, the evaluation framework will be used to evaluate the VITIPI system, the evaluation results will be exposed. Finally, the evaluation framework limitations will be pointed out and different ways for improving the evaluation framework quality will be presented.

Features of existing systems

The goal of all writing assistance systems is to reduce the number of letters the user should type. In other words, text typed by user has to be as short as possible. That is the coding principle [3]. Two kinds of coding can be distinguished : *static* and *interactive* coding.

When a system deals with abbreviation [4], the user knows a list of abbreviated words and rewriting rules. They are defined from the beginning and never change. This is the *static* and *explicit* coding methods. By the opposite, when user doesn't know coding rules, he could use *interactive* and *implicit* coding. When a keystroke is typed, system provides an output and user reacts with another input. There is an interaction between the user and the system. Two kinds of *interactive* coding can be distinguished. In the first one, the most common way is to display a list of words [5], [6]. Whatever the word mode selection (pointing devices, typing a commands list or the number of the word), coding has been done depending on the word list displayed. In the second one, no list of words is displayed. When the user types one letter, the system displays either the ending of the word, or a part of it. As soon as the word remains incomplete, user provides letters to the system depending on the context. This is VITIPI [7] methodology. Those are an interactive, dynamic and implicit coding methods.

Coding principle is not the only feature of writing assistance systems. At the very beginning, systems dealt with *isolated words*. Previous words were not taken into account and syntactic or semantic rules were not integrated to the system. Over the years, syntax and semantic were progressively incorporated into systems, thanks to stochastic methodologies used in speech recognition [8] (n-grams [9], Markov modeling [2], [6]) or syntactical approaches [10] (A.I.,

rewriting rules). Nevertheless, most of the commercial systems are still running with isolated words. In opposition to isolated words, we talk about part of sentences, it means a succession or sequence of words (like n-grams) [9].

Another feature with writing assistance systems is the ability to deal with new entities that have never been found by the system. New entities means new words (words that don't belong to system database, misspelled words with orthographic and/or typing errors), or sequences of words that have never been encountered by the system. At the very beginning, when systems were faced with new entities, they stopped. Nowadays, new entities are integrated in systems, but for most of them predictions cannot be done for these new entities. New entities integration into database could be done on-line or off-line. If the integration is on-line the database is automatically updated and new entities are immediately usable, whereas with an off-line integration, the database needs sometimes user's intervention to be updated because syntactical attributes should be added to new entities. It is not very convenient for the user.

One could encounter systems with standard vocabulary automatically integrated in its database. It could be very fine, but as we will see later on, efficiency could sometimes be lower with a such standard vocabulary. So, it is very useful if user could set or unset the standard vocabulary and automatically generate its own vocabulary.

Another feature of such system, is the ability to generate a database regardless of the original language. For isolated words systems, language could be changed by swapping vocabulary. For systems that run with part of sentences, it is almost impossible except when grammatical rules and syntactical attributes are not explicitly coded into the system.

Finally, the last useful feature is the aptitude for working into a separated window. Some systems, especially those built for disabled people, have their own editor and can not be used with a general commercial editor or other common software (spreadsheet, databases, e-mails,…).

The evaluation framework

It is difficult to make an objective evaluation for various reasons: published results aren't made with the same formula [11]. The rank of the word could be used, it means the number of typed letters before the word is completely displayed. Giving the rank average of all the words is a system efficiency. Some results underlines the time saved with the system, others emphasize the number of keystrokes saved by the system. Since nobody has the same typing rate (particularly with disabled), results are not comparable, furthermore they are not obtained with the same data (or corpora).

According to ergonomic criteria, evaluation is made up with five components: utility, usability, efficiency, no dangerousness and user's satisfaction [12]. Not all criteria will be examined

in the framework. Some of them (utility, dangerousness) are trivial, others (usability, satisfaction) could be evaluated according to the user's or therapist's opinion. Ideally, evaluation shouldn't focus on system, but it should evaluate the couple system and user. A qualitative evaluation [13] outlined the cognitive load for writing assistance systems.

The proposed framework is divided in two parts. The first one deals with system features, it indicates if the feature is present or not in the system. The second one, is focuses on efficiency evaluation with a definition sent forward..

System features

- Coding principles
    - Explicit coding (with coding abbreviations)
    - Implicit and interactive coding
        - With list of words
        - Without list of words
- Running system
    - With isolated words
    - With parts of sentence
- Typing adaptability
    - Typing mistakes
    - Orthographic mistakes
    - New entities (new words or sentences)
- Database update
    - On-line update
    - Off-line update
- Standard lexicon optionally set
- Automatic database building
    - Only for the original language system
    - Regardless of the original language system
- Separated window editor

System efficiency

We told that there are various ways to determine system efficiency. When systems use interactive coding displaying lists of words, authors often publish the average number of keystrokes typed to select one word. But words size is context depending. For some typical corpora like medical, technical, scientific, or very inflected languages such as German [15], words are larger than ordinary corpus or language. To our opinion publishing the system displayed letters percentage is better.

The time saving ratio evaluation is computed as follow: The ratio of input keys typed by the user compared to the ratio of output characters provided by the system. To evaluate it, we have to compute the number of keys typed (or **I**nput) by the user with system assistance :

Let I = *Input_symbols* + *key_functions*

With:

- *Input_symbols*: Number of letters typed by user. It includes also digits or numbers contained in the text and spacing (just one for each word)
- *key_functions*: Number of keys functions typed to select word in list or to correct word prediction.

Without system assistance, the user should type the entire text (or **T**otal letters).

Let T = *Input_symbols* + *Output_symbols*

With :

- *Output_symbols*: Number of symbols automatically provided by the system.

Finally, the Ratio Time Saving (*R.T.S)* is obtained by the following formula which is similar to Zagler Keystroke-Saving-Rate [15].

$$\boldsymbol{R.T.S} = \left(1 - \frac{(\boldsymbol{I})}{(\boldsymbol{T})}\right)$$

For a given system, the R.T.S. depends on system features and on the test text (i.e. the text used to compute the R.T.S) The more the test text is near to the database, the more the R.T.S is high. If a lot of test text sentences belong to the database, few uncorrected predictions are made, thus few key functions are typed to correct them. When all test text sentences are in the database, the R.T.S. is maximized.

Once the evaluation framework has been defined, we are going to take VITIPI system as an instance of the evaluation framework.

VITIPI Evaluation

VITIPI System is a writing assistance system based on an interactive coding without words displaying list. Thus, we are going to take it to instance the evaluation Framework. More details can be found [7] showing VITIPI running and the obtained results [16]. Following the evaluation framework, we will first give the system features, in a second part system efficiency will be sent with various results.

VITIPI features

VITIPI features are summarized in the following Table 1 according to the system's features list fed to the evaluation framework.

System features	Name of the systems			
	VITIPI	Other systems		
		S1	…	Sn
Explicit coding (with coding *abbreviations*)	No			
Implicit coding with list of words	Yes (option)			
Implicit coding without list of words	Yes			
Running with isolated words	Yes (option)			
Running with part of sentences	Yes			
Typing adaptability (typing mistakes)	Yes			
Typing adaptability (orthographic mistakes)	Yes (option)			
Typing adaptability (new entities i.e. words/sentences)	Yes			
On-line/Off-line update database	On-line			
Standard lexicon optionally set	Yes			
Automatic database building for a single language	Yes			
Automatic database building for several languages	Yes			
Editor with separated windows	Yes			

Table 1 : Systems features instanced by VITIPI system

VITIPI efficiency

Various tests were made for VITIPI efficiency evaluation [16], [14]. In a first step, we are going to describe methodological aspects, in a second one, results will be given and analyzed.

Methodological aspects

The aims of these experiments were to test various parameters. We wanted to know the standard lexicon influence upon the R.T.S results, then we wished to evaluate VITIPI efficiency with unknown system sentences. Thus, the system has been tested, with known and unknown sentences. We have tried to evaluate the amount of sentences that system needs to yield an good result. Finally, optimal length of words sequence was studied.

Evaluations were made with 3 different corpora. The first one deals with weather forecasting French sentences collected during 24 days. It consists of 439 sentences, 7,830 words 985 vocabulary size. The second is composed of 20 e-mails exchanges concerning a workshop organization meeting. The whole text consist of 159 sentences made up with 2,539 words from a 912 words vocabulary. The third one contains 30 letters wrote by a French disabled association (for entertainment meetings). The text consist of 499 sentences, 7,579 words (from a 1,275 words vocabulary) and was divided into 3 sub corpora. Standard lexicon was made up of 5,930 useful French words.

Experiments and results

First experiment were made with system known sentences in order to evaluate the maximum value of the R.T.S. and the standard lexicon influence. Two different systems were built, both contained all corpus sentences. On one hand, standard lexicon was added to the system, on the other, there was no additional word. Once systems were built, they were faced with all known

sentences contained in corpus, and they were tested with 0 previous words (i.e. isolated words), 2 and 9 previous words. Results are shown in the following tables 2 and 3.

	Weather forecasting corpus (RTS in %) and known sentences					
	With standard lexicon			Without standard lexicon		
	0 previous word	*2 previous words*	*9 previous words*	*0 previous word*	*2 previous words*	*9 previous words*
Day 1	34.3	77.9	78.2	55.6	82.3	82.7
Day 1 to 24	30.3	71.4	74.2	38.0	75.5	78.9

Table 2 : Standard lexicon influences on R.T.S. with weather forecasting corpus and known sentences.

	e-mail corpus messages (RTS in %) and known sentences					
	With standard lexicon			Without standard lexicon		
	0 previous word	*2 previous words*	*9 previous words*	*0 previous word*	*2 previous words*	*9 previous words*
Message 1	42.1	82.9	83.1	53.9	84.8	85.0
Mess 1 to 20	34.6	79.8	80.4	43.1	81.9	82.7

Table 3 : Standard lexicon influences on R.T.S. with e-mails corpus messages and known sentences.

It could be noticed that standard lexicon is not useful for writing assistance, R.T.S. is lower with standard lexicon than without it. It is particularly emphasized with 0 previous word (isolated words). One could explain this lower results because the more the vocabulary size grows, the more it will be faced with ambiguities, and the ratio prediction goes down. The system will have to wait for the user's letters to clear up ambiguities. Conversely, the more the vocabulary size decreases the more the ratio prediction goes up. The same phenomenon cold be observed with sentences. When the number of sentences grows, the R.T.S is going down.

Standard lexicon is not useful with known sentences, will it be helpful with unknown sentences ? In order to test it, the following experiment has been made. As in the above experiment, two systems were built (with and without lexicon), but unlike the first experiment, all corpus sentences were not incorporated into the system. Some sentences were left out of the system in order to test it. For example, once the first day weather forecasting was entered in the system, the system proceeds by testing the second day weather forecasting considering the information of the first day. The same experiment was made with 23 days weather forecasting tested with the 24^{th} day. Results are shown in Table 4.

	Weather forecasting corpus (RTS in %) and unknown sentences					
	With standard lexicon			Without standard lexicon		
	0 previous word	*2 previous words*	*9 previous words*	*0 previous word*	*2 previous words*	*9 previous words*
Day 2	18,0	21,0	25,0	21,0	26,0	27,0
Day 24	20,0	45,0	42,0	27,0	47,0	45,0

Table 4 : Standard lexicon influences on R.T.S. with weather forecasting corpus and unknown sentences.

It could be noticed that standard lexicon is not useful for unknown sentences. We also want to outline that R.T.S with 2 previous words is higher than with 9. Why ? If 9 previous words are taken into account, the number of words that are liable to be written is reduced to one. If it isn't the desired words, user has got to refuse the proposed word by typing a command. By the opposite, with only 2 previous words, the number of words that are liable to be written is highest, so there is a strong likelihood that that the user's desired word will be found. User should not have to refused, and system may help to write the ending of the word.

Now we are going to examine system reaction with an empty database in order to test learning ability. The third corpus has been divided into three sub-corpora containing about 10 letters. In a first step, the empty system was tested with first sub-corpus sentences. In a second step, system was built with first sub-corpus, and tested with the second sub-corpus. Finally, the first and second sub-corpora were introduced into the system and was tested with the third sub-corpora. All tests were carried out without standard lexicon. Results are shown in table 5.

	Disabled association corpus letters (R.T.S. in %). With unknown sentences without standard lexicon		
	0 Previous word	2 Previous words	5 Previous words
1st sub-corpus	21.5	28.2	28.1
2nd sub-corpus	27.6	38.8	38.5
3rd sub-corpus	30.9	48.3	48.2

Table 5 : Learning ability and prediction. R.T.S. results with a disabled association corpus letters (unknown sentences without standard lexicon).

It could be noticed that even with an empty database, system is able (after 4 or 5 letters) to predict the ending of a word or a part of it. As we have explained above, 5 previous words is not better than 2 previous words.

Discussion

The presented evaluation framework is yet usable, but should be improved and emended. One may add others relevant features in system features part. For the efficiency part, we have only gauged the system efficiency with R.T.S. indicator. Following Sperandio's opinion [12], other items should be very important to gauge, but very hard to evaluate. User's satisfaction is very relevant and

it will be a good challenge to define an indicator, regardless of user's particularities. Furthermore, there is no way to know if the selected system comes up to the user's expectations.

There is another problem which has never been tackled : does writing assistance influence users with the produced text? In others words, what is the part played by proposed words in user's writings? Do they improve user's vocabulary, or by the opposite, do they restrict user's thoughts ? It should be very attractive and convenient to only select the proposed words and syntactical patterns displayed by the system.

Conclusion

This is a first step for an evaluation framework. To our opinion, users must be able to evaluate writing assistance systems, in order to select the best one according to their particular needs. In the other hand, an evaluation framework should be very helpful for designers. It could help them to adapt their parameters (n-grams lengths, stochastic Markov models, grammatical rules, syntactical and/or semantic attributes, ….). Once a common learning corpus and testing text will be chosen by the working community, we hope that this evaluation framework will be emended and used.

References

1. Boissière, Ph., Dours, D. *From a specialised writing interface created for the disabled, to a predictive interface for all: the VITIPI System*. in 1st International UAHCI 2001 Conference, News-Orleans, pp. 895-899, 5-10 August 2001.
2. Hunnicutt, S., Carlberger, J., *Improving Word Prediction Using Markov Models and Heuristic Methods*. in Augmentative and Alternative Communication, 17, PP 255-264, 2001.
3. Cantegrit, B. Toulotte, J.M. *Réflexions sur l'aide à la communication des personnes présentant un handicap moteur*. in Atelier Thématique TALN 2001 PP 193 - 202, Tours, 2-5 Juillet 2001.
4. Ricco, X. Dutoit, T. *Vers un logiciel multilingue et gratuit pour l'aide aux personnes handicapées de la parole : HOOK*. in Atelier Thématique TALN 2001 PP 223 - 232, Tours, 2-5 Juillet 2001.
5. Maurel, D. Rossi, N. Thibault, R. *Handias : un système multilingue pour l'aide à la communication de personnes handicapées.* in Atelier Thématique TALN 2001 PP 203 - 212, Tours, 2-5 Juillet 2001.
6. Menier, G. Poirier, F. *Système adaptatif de prédiction de texte*. in Atelier Thématique TALN 2001 PP 213 - 222, Tours, 2-5 Juillet 2001.

7. Boissière, Ph., Dours, D. *VITIPI : Versatile Interpretation of Text Input by Persons with Impairments*. in 5th national Conference on Computers for Handicapped Persons, pp.165-172, Linz July 1996,

8. Jelinek, F. *Self-organized language modeling for speech recognition*. Readings in Speech Recognition, Waibel and Lee (Editors). Morgan Kaufmann. 1989

9. Lesher, G.W. Moulton, B.J. Higghinbotham, D.J. *Effects of ngram order and training text size on word prediction.* in Proceedings of the RESNA '99 Annual Conference, (pp. 52-54), Arlington, VA: RESNA Press 1999.

10. Pasero, R. Sabatier, P. *The Access Project : Development platform for unified access to enabling environments.* in Natural Language Understanding and computational Logic, Lecture Notes in Computer Science, Springer 1998.

11. Carlberger, J. *Design and Implementation of a Probabilistic Word Prediction Program*. in Master's Thesis Dept. of Speech, Music and Hearing, KTH, SE-100 44 Stockholm, Sweden (1997).

12. Sperandio, J.C., *Critères Ergonomiques d'Assistance Technologique aux Opérateurs*. in JIM'2001 : Interaction Homme-Machine & Assistance pour Grand Public in Systèmes Complexes / Handicap, p. 30-37 Metz, 4-6 juillet 2001.

13. Klund, J. Novak, M. *If Word Prediction Can Help, Which Program Do You Choose?*. in Technology Special Interest Section Quarterly, Vol 7, N°1, March 1997, American Occupational Therapy Association. 1997.

14. Boissière, Ph., Dours, D. *A Proposal of an Evaluation Framework for Writing Assistance Systems: Application to VITIPI*. in 8th Conference of the ICCHP'2002, Linz (Austria) 15-20 July 2002.

15. Zagler, W.L Seisenbacher, G. *German Language Predictive Typing : Results from a Feasibilty Investigation*. in Proceedings of the ICCHP'2000, pp.771-779, Karlsruhe, 17-21 July 2000.

16. Dubus, N. *Evaluation de l'interface intelligente d'aide à la saisie informatique, VITIPI au lycée "Le Parc Saint-Agne".* in Journal d'Ergothérapie PP 95 - 100 MASSON Mars 1996.