USER INTERFACE DESIGN FOR VOICE CONTROL SYSTEMS

Wolfgang Tschirk
Siemens AG Austria, Speech Recognition Resort
Erdberger Lände 26, A-1031 Wien, Austria

Keywords: Voice control systems, speech recognition, user interfaces, man-machine interfaces.

Abstract: A voice control system converts spoken commands into control actions, a process which is always imperfect due to errors of the speech recognizer. Most speech recognition research is focused on decreasing the recognizers' error rates; comparatively little effort was spent to find interface designs that optimize the overall system, given a fixed speech recognizer performance. In order to evaluate such designs prior to their implementation and test, three components are required: 1) an appropriate set of performance figures of the speech recognizer, 2) suitable performance criteria for the user interface, and 3) a mathematical framework for estimating the interface performance from that of the speech recognizer. In this paper, we will identify four basic interface designs and propose an analytical approach for predicting their respective performance.

1 INTRODUCTION

James Glass *et al.* point out that "developing conversational interfaces is a classic chicken and egg problem. In order to develop system capabilities, one needs to have a large corpus of data [...]. In order to collect data that reflect actual usage, one needs to have a system that users can speak to." (Glass et al., 2000). Compared to the vast literature on traditional speech recognition aspects like signal analysis, feature extraction, pattern comparison techniques, and search strategies, for which excellent reports are available in (Rabiner and Juang, 1993), (IEEE, 1994), and (IEEE, 2002), little work was devoted to the development of analytical methods for user interface design. However, analytical approaches can considerably shorten trial and error loops in system development and raise robustness and user-friendlyness of voice control systems in a way hardly achievable by classical speech recognizer improvements.

A mathematical approach for dialogue control is proposed in (Niimi and Nishimoto, 1999), where the authors derive relations between speech recognizer performance and dialogue efficiency, and compare four strategies of confirming user inputs. They focus on keyword confusion and leave the problem most severe in real world voice control systems, namely to balance the acceptance of non-keywords with its counterpart, the keyword rejection, as future work. The computer-aided design and development approach for spoken dialogue systems given in (Lin and Lee, 2001) estimates performance figures by quantitative simulations, comparing different strategies of query and confirmation on a finite state machine model of the user interface.

In our paper, we consider all possible user interface errors for four basic interface designs. We show analytically how the error rates observed on the user interface depend on those of the speech recognizer, and how the former can be minimized while the latter remain unchanged. As an auxiliary result, we express the error rates for different vocabulary sizes in terms of the rates for a single vocabulary of arbitrary (however reasonable) size, an achievement which greatly reduces test effort. Throughout this paper, we concentrate on isolated word recognition, which is the dominant technology for a wide range of voice control systems. We take into account that, typically, users of such systems are not pleased with a period of uncertainty after having entered a command; therefore, we do not allow delayed decisions, although they are found to improve the performance of spoken dialogue systems (Souvignier et al., 2000).

Another crucial aspect of voice control system design lies beyond the scope of our paper: the question of selecting appropriate commands for a given set of

Tschirk W. (2004).
USER INTERFACE DESIGN FOR VOICE CONTROL SYSTEMS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 21-26
DOI: 10.5220/0002602100210026

control functions – words that are easily recognized automatically and, at the same time, correspond to the intuitive choice of most users.

2 SPEECH RECOGNIZER ERRORS

2.1 Error Types

In a speech recognizer's front end, the incoming audio signal is converted into parameters, commonly called *features*, that carry information relevant for the recognition act. The result is a pattern of features, which may be a vector composed of energy values over time and frequency or of other appropriate measures (Rabiner and Juang, 1993). Since here we do not deal with the feature extraction, we consider the pattern rather than the audio signal as the input to the speech recognizer.

Then, within the context of isolated word recognition, a speech recognizer is an algorithm that assigns one of $N+1$ classes $W_i, 0 \leq i \leq N$, to a pattern $\mathbf{p}$. We denote W_0 the garbage class, i.e. the class which shall be assigned to all patterns derived from background noise. The set of classes $\{W_i, 0 \leq i \leq N\}$ is the recognizer's vocabulary; N, the number of non-garbage classes, is the vocabulary size.

For classifying an incoming pattern $\mathbf{p}$, the recognizer calculates a score s_i for each class W_i. The higher s_i, the better $\mathbf{p}$ matches W_i, such that finally the class with the highest score is assigned to $\mathbf{p}$. (For recognizers that deliver a low score as an indicator for a good match, we reverse the sign of the score). The scores shall only depend on $\mathbf{p}$; we envisage a functional split where the recognizer is context-free and any context-dependency is located in an extra component of the user interface.

When a pattern is classified as garbage, we say that it is rejected; otherwise we say that it is accepted. Each recognition error falls into one of the following three categories: *confusion* (a non-garbage pattern is assigned the wrong non-garbage class), *false rejection* (a non-garbage pattern is classified as garbage), and *false acceptance* (a garbage pattern is assigned a non-garbage class).

A recognizer's performance can be described in terms of rates c of confusions, r of false rejections, and a of false acceptances. Such rates are usually estimated by feeding a set of patterns which were not used for training (the test set) into the recognizer and counting the incorrect outcomes appropriately. If the test set represents the patterns expected during operation, the rates measured can be viewed as a good guess for the underlying probabilities.

2.2 Error Rates and Vocabulary Size

In Section 6, we give an algorithm for predicting the error rates *observable on the user interface*. It involves the estimation of the speech recognizer's error rates on different vocabulary sizes. Ideally, they are obtained from testing; however, this might require an enormous test effort and the availability of sufficient data. As an alternative, we now derive a formalism for predicting the error rates c_M, r_M, and a_M on a vocabulary of size M from the rates estimated on a vocabulary of size N. It is particularly helpful in the design of voice control systems featuring large numbers of commands when data are available for only a small number of words, and relies on assumptions reasonable in the absence of other evidence:

If the recognizer receives a non-garbage pattern $\mathbf{p}$ of class W_k, we assume that 1) the probability that $\mathbf{p}$ is rejected rather than accepted as W_k does not depend on the presence of classes other than W_0 and W_k, and 2) the probability that $\mathbf{p}$ is classified as W_i rather than W_k is equal for all $i \notin \{k, 0\}$ and independent of the presence of classes other than W_i and W_k. If the recognizer receives a garbage pattern, we assume that 3) the probability that $\mathbf{p}$ is classified as W_i rather than garbage is equal for all $i \neq 0$ and independent of the presence of classes other than W_i and W_0.

With these assumptions and the trivial figure $c_1 = 0$, we obtain (see Appendix) for $M, N > 0$:

$$c_M = \frac{(M-1)c_N}{(N-1)+(M-N)c_N}, \tag{1}$$

$$r_M = \frac{(N-1)r_N}{(N-1)+(M-N)c_N}, \tag{2}$$

$$a_M = \frac{Ma_N}{N+(M-N)a_N}. \tag{3}$$

3 USER INTERFACE ERRORS

Also on the user interface, we observe confusions, false rejections, and false acceptances. How the rates of these errors derive from the speech recognizer's error rates, depends on the particular user interface design. For a number of design alternatives, we will give these relationships in the following sections.

In general, no design will be optimum with respect to all three types of errors. Therefore, each voice control system calls for its own optimization criterion. Some authors propose to use weighed sums of the error rates (Villarrubia and Acero, 1993) and to set the weights according to the requirements of the particular application. Others measure dialogue efficiency in

terms of the average number of exchanges taken (Niimi and Nishimoto, 1999) or the percentage of satisfied users (Lin and Lee, 2001).

In our performance estimations, we will stick to the rates of confusions, false rejections, and false acceptances, and, from the first two of them, which relate to incorrect system reactions on correct user inputs, we will estimate a lower bound for the rate of failed command sequences (a command sequence is a concatenation of keyword utterances required to make the voice control system perform a certain action; e.g. a command sequence consisting of *light* and *switch on* may cause the voice control switch on the light).

4 SUBVOCABULARY TYPES

4.1 Embedded Subvocabulary

In a voice control system designed to control a light, a heater, and a telephone, the word *warmer* makes sense in the context of the heater control but not when the light menu or the telephone menu are selected; the commands *switch on* and *switch off* may be allowed for both the light and the heater. In such a way, at each time instant only a subset of the speech recognizer's vocabulary is active, and these subvocabularies may have words in common.

If the system considers all classes, whether they belong to the active subvocabulary or not, we call the subvocabulary *embedded*. First, we analyze this strategy in its simplest form: an *observable confusion* occurs, if the recognizer commits a confusion and the result belongs to the active subvocabulary; an *observable false rejection* occurs, if 1) the recognizer commits a false rejection, or 2) the recognizer commits a confusion and the result does not belong to the active subvocabulary; an *observable false acceptance* occurs, if the recognizer commits a false acceptance and the result belongs to the active subvocabulary.

In the absence of user errors, which we assume throughout this analysis, a correct non-garbage result always belongs to the active subvocabulary. A wrong non-garbage result is assumed to fall into each of the remaining classes with equal probability. From this, we find the rates C_N^S of confusions, R_N^S of false rejections, and A_N^S of false acceptances observable on the user interface for an embedded subvocabulary of size $S > 0$ out of a total vocabulary of size $N \geq S$ as follows:

$$C_N^S = \frac{S-1}{N-1} c_N, \tag{4}$$

$$R_N^S = r_N + \frac{N-S}{N-1} c_N, \tag{5}$$

$$A_N^S = \frac{S}{N} a_N. \tag{6}$$

For large N with small S, a more sophisticated approach is favourable: if a non-garbage result falls out of the active subvocabulary, then the class which scored next is taken as a new hypothesis, and this process is repeated until either a hypothesis is garbage or falls into the active subvocabulary, or a predefined number H of hypotheses were examined. This strategy is implemented in the voice control devices described in (Tschirk, 2001). For $2 \leq H \leq N - S$, the observable error rates are given by:

$$C_N^{S,H} = C_N^S + \sum_{i=2}^{H} \frac{S-1}{N-i} c_{N-i+1} \cdot \prod_{j=1}^{i-1} \frac{N-S-j+1}{N-j} c_{N-j+1}, \tag{7}$$

$$R_N^{S,H} = R_N^S + \prod_{i=2}^{H} \frac{N-S-i+1}{N-i} c_{N-i+1} + \sum_{i=2}^{H} r_{N-i+1} \prod_{j=1}^{i-1} \frac{N-S-j+1}{N-j} c_{N-j+1}, \tag{8}$$

$$A_N^{S,H} = A_N^S + \sum_{i=2}^{H} \frac{S}{N-i+1} a_{N-i+1} \cdot \prod_{j=1}^{i-1} \frac{N-S-j+1}{N-j+1} a_{N-j+1}. \tag{9}$$

In Section 7, we mainly refer to the usage of embedded subvocabularies as given by Equations (4) to (6). The power of examining more than one hypothesis is shown in Section 7.4.

4.2 Separated Subvocabulary

If the system evaluates each incoming pattern with respect to only those classes represented in the active subvocabulary, disregarding the other ones, we call the subvocabulary *separated*. The observable error rates on a separated subvocabulary of size S are derived from Equations (4) to (6) by setting $N = S$.

5 MENU ARRANGEMENTS

5.1 Hierarchical Menus

On an voice interface featuring menus, the user has to select the appropriate menu prior to submitting a control command. In our example of Section 7.2, each controllable device has its own menu. We can place a device selection menu on the top level of the user interface, such that switching on the light requires either a sequence of 3 commands: *select* to go to the

selection menu, *light* to select the light, and *switch on* to switch it on, or, if the light menu was the last one selected, a sequence of length 1: *switch on*. We call this menu arrangement *hierarchical*.

5.2 Connected Menus

In order to support direct switching between menus, we include each device identifier into each of the device menus, thus eliminating the selection menu. Switching on the light requires either a command sequence of length 2: *light* and *switch on*, or of length 1: *switch on*, depending on the recently selected menu. We call this arrangement *connected*. Compared to the hierarchical one, the connected arrangement requires fewer steps on larger subvocabularies.

5.3 Command Sequence Behaviour

The error rates defined so far apply to single-pattern reception. Now we define a command sequence failure rate. A command sequence is considered successful, if each command is recognized correctly, otherwise it is considered failed. For the purpose of simplicity, we assume that command sequences are not interrupted by garbage reception. Consequently, the command sequence failure rate below gives a lower bound for the actual figure, and the false acceptance rate has no impact on the command sequence performance and is kept as an extra figure. Assuming independent recognition errors, the failure rate F_L of a command sequence of length L is given by

$$F_L = 1 - \prod_{i=1}^{L} \left(1 - (C_{N_i}^{S_i} + R_{N_i}^{S_i})\right), \qquad (10)$$

where N_i and S_i are the recognizer's vocabulary and subvocabulary size, respectively, in the user interface state corresponding to the reception of the i-th command. Equation (10) holds for all menu arrangements and all subvocabulary types.

6 BASIC INTERFACE DESIGNS

Combining the alternatives given in Sections 4 and 5, we identify four basic user interface designs:

design HE: hierarchical menu arrangement, embedded subvocabularies,

design HS: hierarchical menu arrangement, separated subvocabularies,

design CE: connected menu arrangement, embedded subvocabularies,

design CS: connected menu arrangement, separated subvocabularies.

In Section 7, we will evaluate, for three example systems, these four designs, in order to point out the advantages and drawbacks of each approach under the conditions stated.

The evaluation of a design consists of four steps:

step 1: estimate the vocabulary and subvocabulary sizes corresponding to each menu,

step 2: estimate the necessary speech recognizer error rates via testing or by using Equations (1) to (3) with appropriate test figures as inputs,

step 3: calculate the observable error rates for each menu, using Equations (4) to (6) or (7) to (9),

step 4: for each command sequence type, calculate the failure rate according to Equation (10).

7 EXAMPLE SYSTEMS

7.1 Example Speech Recognizer

Suppose a speech recognizer with the error rates: $c_{10} = 0.005, r_{10} = 0.03$, and $a_{10} = 0.20$ for a vocabulary of size 10, on which the voice control systems of Sections 7.2 to 7.4 shall be based. We do not consider modifications of the recognizer itself. Instead, we ask for the optimum user interface design for the respective task, given the recognizer as it is.

7.2 Light, Heater, and Telephone Control

Our first example system shall control a light with the commands *switch on*, *switch off*, *brighter*, and *darker*, a heater with *switch on*, *switch off*, *warmer*, and *cooler*, and a hands-free telephone with *connect*, *disconnect*, *louder*, and *softer*. The device selectors are *light*, *heater*, and *telephone*. For the hierarchical arrangement, *select* shall be used to enter the device selection menu. Note that each device has the same number of control commands, which facilitates our analysis; in real world systems, different devices will, in general, have control command sets of different sizes. We analyze the system following Section 6, using Equations (1) to (3) together with the recognizer figures of Section 7.1 in step 2. The results are shown in Table 1; there, the *select-and-control failure rate* corresponds to command sequences required to select a device and invoke a control action on it, whereas the *control failure rate* corresponds to commands invoking a control action on an already selected device.

In this example, the lowest select-and-control failure rate is achieved by employing the connected menu arrangement together with separated subvocabularies; however, it is paid with the highest false acceptance rate.

Table 1: Performance figures for different user interface designs of a voice control system featuring device menus. In the hierarchical designs, the false acceptance rate is taken from the device menus, since voice control systems assume that state most of the time (in the connected designs, the false acceptance rate is equal for all menus).

	HE	HS	CE	CS
select-and-control failure rate	0.1073	0.0928	0.0719	0.0657
control failure rate	0.0371	0.0323	0.0366	0.0334
false acceptance rate	0.0926	0.1111	0.1321	0.1489

7.3 Single–Device Control

The second example relates to a single 50-commands control menu without subvocabularies. All designs are identical, each command sequence is of length 1. The performance is given in Table 2.

Table 2: Performance figures for a large single device control menu.

	all designs
command sequence failure rate	0.0560
false acceptance rate	0.5556

Here, the exorbitant false acceptance rate is likely to cause permanent unintentional activation of the system. This undesired behaviour is a consequence of the large number of commands allowed at each time instant without any context.

7.4 Keyword Activation

As a third example, we modify the interface of Section 7.3. We reduce the number of false acceptances by introducing a sleep mode, in which the system accepts nothing but a certain *wake up* keyword. After having been activated with *wake up*, it accepts each one of its 50 commands and an extra *sleep* keyword, which brings it back into the sleep mode.

The results of the analysis are shown in Table 3. The *wake-up-and-control failure rate* corresponds to command sequences required to get the voice control system out of its sleep mode and invoke a control action, whereas the *control-or-sleep failure rate* corresponds to commanding an already active system or bringing it back into the sleep mode. Since there are only two menus, hierarchical and connected arrangement are identical. In the active mode (first and second row), the interface behaves very similar to the one of Section 7.3. In the sleep mode, false acceptances are almost suppressed (last row), and we find that the technique of embedded subvocabularies yields a false acceptance behaviour far better than that of separated subvocabularies, at moderate cost with respect to the wake-up-and-control failure rate (third row).

Table 3: Performance figures for different user interface designs of a voice control system with keyword activation.

	HE, CE	HS, CS
control-or-sleep failure rate	0.0570	0.0565
active mode false acceptance rate	0.5543	0.5604
wake-up-and-control failure rate	0.1108	0.0849
sleep mode false acceptance rate	0.0109	0.0244

If, in case of an out-of-subvocabulary rejection in the sleep mode, we examine a second hypothesis according to Equations (7) to (9), we can lower the sleep mode false acceptance rate almost without raising the wake-up-and-control failure rate compared to the separated subvocabulary design:

Table 4: Performance figures for a voice control system with keyword activation, examining a second hypothesis in case of an out-of-subvocabulary rejection in the sleep mode.

	HE, CE ($H = 2$)
wake-up-and-control failure rate	0.0861
sleep mode false acceptance rate	0.0170

8 CONCLUSION

We presented an analytical approach for estimating the performance of voice control user interfaces, applicable to systems based on isolated word recognition and featuring menus. It allows for deriving design guidelines and focuses on user interface optimization, given the speech recognizer's performance. In order to make such estimation feasible even if sufficient test data are not available, we derived a formalism for predicting a speech recognizer's error rates on different vocabulary sizes from the rates obtained on a single vocabulary of arbitrary (however reasonable) size.

We illustrated the approach by comparing four basic user interface designs. Mechanisms for improving voice interfaces which were left out of the study, such as the weighing of errors according to their relative

importance or the minimization of overall error rates by taking into account the a priori probability of commands, can easily be integrated into the formalism.

The framework given here can also be used to select the best recognizer for a particular task, which may be characterized by non-negotiable parameters such as the number of menus and the menu sizes.

The methods presented were developed in the course of the design of voice remote control systems for physically disabled people (Tschirk, 2001). They were found useful for early detection of design strengths and weaknesses. Clearly, they cannot eliminate the need for exhaustive real world testing.

APPENDIX

We view the recognition of a non-garbage pattern of class W_k as an experiment (Papoulis, 1984); its outcomes are the class indices $j \in \{0, \ldots, N\}$. To each outcome j, we assign a probability $p(j)$, which is either $p_1(N)$: the probability of correct recognition, or $p_2(N)$: the probability of confusion into a specific class, or $p_3(N)$: the probability of rejection.

$$\begin{aligned} p(k) &= p_1(N), \\ p(i, i \notin \{k, 0\}) &= p_2(N), \\ p(0) &= p_3(N), \end{aligned}$$

with

$$p_1(N) + (N-1)p_2(N) + p_3(N) = 1. \quad (11)$$

Confusion rate and false rejection rate are given by

$$c_N = (N-1)p_2(N), \quad (12)$$

$$r_N = p_3(N). \quad (13)$$

From assumption (1) of Section 2.2 follows that $p_3(N)/(p_1(N)+p_3(N))$ does not depend on N; from assumption (2) follows that $p_2(N)/(p_1(N)+p_2(N))$ does not depend on N. Thus, both expressions are constant, and there exist constant $u = p_1(N)/p_3(N)$ and $v = p_2(N)/p_3(N)$, such that we can rewrite Equations (11) to (13) to

$$c_N = \frac{(N-1)v}{u + (N-1)v + 1},$$

$$r_N = \frac{1}{u + (N-1)v + 1}.$$

Since this holds for all $N > 1$, we obtain Equations (1) and (2).

Now we view the recognition of a garbage pattern as an experiment and assign to each outcome either $q_1(N)$: the probability of rejection, or $q_2(N)$: the probability of acceptance with respect to a specific class.

$$\begin{aligned} p(0) &= q_1(N), \\ p(i, i \neq 0) &= q_2(N), \end{aligned}$$

with

$$q_1(N) + Nq_2(N) = 1. \quad (14)$$

The false acceptance rate is given by

$$a_N = Nq_2(N). \quad (15)$$

From assumption (3) of Section 2.2 follows that $q_2(N)/(q_1(N) + q_2(N))$ does not depend on N. Thus, there exists a constant $w = q_1(N)/q_2(N)$, such that we can rewrite Equations (14) and (15) to

$$a_N = \frac{N}{w + N}.$$

Since this holds for all $N > 0$, we obtain Equation (3).

REFERENCES

Glass, J., Polifroni, J., Seneff, S., and Zue, V. (2000). Data collection and performance evaluation of spoken dialogue systems: The MIT experience. Massachusetts Institute of Technology.

IEEE (1994). Special section on robust speech recognition. In *IEEE Transactions on Speech and Audio Processing vol. 2, no. 4, pp. 549-643, October 1994*.

IEEE (2002). Special issue on automatic speech recognition for mobile and portable devices. In *IEEE Transactions on Speech and Audio Processing vol. 10, no. 8, pp. 529-658, November 2002*.

Lin, B.-S. and Lee, L.-S. (2001). Computer-aided analysis and design for spoken dialogue systems based on quantitative simulations. In *IEEE Transactions on Speech and Audio Processing vol. 9, no. 5, pp. 534-548, July 2001*.

Niimi, Y. and Nishimoto, T. (1999). Mathematical analysis of dialogue control strategies. In *Proceedings of EUROSPEECH 99 vol. 3, pp. 1403-1406*.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ.

Souvignier, B., Kellner, A., Rueber, B., Schramm, H., and Seide, F. (2000). The thoughtful elephant: Strategies for spoken dialog systems. In *IEEE Transactions on Speech and Audio Processing vol. 8, no. 1, pp. 51-62, January 2000*.

Tschirk, W. (2001). Neural net speech recognizers. Voice remote control devices for disabled people. In *e & i Artificial Intelligence 7/8/2001, pp. 367-370*. Springer.

Villarrubia, L. and Acero, A. (1993). Rejection techniques for digit recognition in telecommunication applications. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1993, pp. 455-458*.