SELF–OPTIMIZING VOICE CONTROL USER INTERFACE

Wolfgang Tschirk

Siemens AG Austria, Speech Recognition Resort
Erdberger Lände 26, A-1030 Wien, Austria (Europe)
phone: +43 5 1707 35193, fax: +43 5 1707 55120, email: wolfgang.tschirk@siemens.com

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. In this paper, we present an alternative approach: the development of a strategy that minimizes the number of errors visible on the user interface, given a fixed speech recognizer performance. We introduce a unified view on the recognition act, i.e. the classification of an incoming sound pattern, and derive a self-optimization algorithm for the user interface. With this algorithm, each recognition act is optimized with respect to its own criterion, independently from all other recognition acts. As a result, the voice control system adapts flexibly to the demands of the respective situation.

1. INTERFACE COMPONENTS

1.1 Speech Recognizer

In every speech recognizer's front end, the incoming audio signal is converted into parameters, commonly called *features*, that carry relevant information. The result is a pattern of features, which, for example, may be a vector composed of energy values over time and frequency or of other appropriate measures [5]. Since here we do not deal with the feature extraction itself, 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 $V+1$ classes $W_i, 0 \leq i \leq V$, to a pattern $\mathbf{p}$. Without loss of generality we denote W_0 the garbage class, i.e. the class which shall be assigned to all patterns not derived from keyword utterances (e.g. patterns from background noise, music, or conversation). The set of classes $\{W_i, 0 \leq i \leq V\}$ is the recognizer's vocabulary $\mathbb{V}_0$; V, the number of non-garbage classes, is the vocabulary size. A speech recognizer designed to recognize V keywords shall throw away all sounds other than utterances of said keywords; in other words, it shall assign the garbage class to them.

For classifying $\mathbf{p}$, the recognizer estimates a score s_i for each class W_i. The higher s_i, the better $\mathbf{p}$ matches W_i, such that the class with the highest score is assigned to $\mathbf{p}$. (Recognizers that score low for good matches become compatible with the preceding rule if 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 elsewhere in the user interface.

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

The performance of a speech recognizer with vocabulary size V can be described in terms of rates c_V of confusions, r_V of false rejections, and a_V of false acceptances. In non-parametric recognition [1], such rates are usually estimated by feeding a set of patterns 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.

1.2 Menu Control

Also on the user interface we observe confusions, false rejections, and false acceptances. However, the rates of these errors as observed on the user interface are, in general, different from the speech recognizer's error rates, since the highest scoring class can only be accepted if it fits into the context. In a system designed to control a light, a heater, and a telephone, the word *warmer* makes sense with respect to the heater but not with respect to the light or the telephone; the commands *switch on* and *switch off* may be allowed for both the light and the heater. In such a way, for each recognition act only a subset of the speech recognizer's vocabulary is active, and these subvocabularies (menus) may have words in common.

Let the user interface consist of a context-free speech recognizer, followed by a menu control unit. The recognizer passes a list of classes ranked according to their scores to the menu control, which evaluates them, applying one of the following strategies:

1) The menu control scans the set of classes $\mathbb{S}_0 = \mathbb{S} \cup \mathbb{G}$, where $\mathbb{S}$ is the active subvocabulary and $\mathbb{G}$ is the garbage set ($\mathbb{G} = \{W_0\}$), and selects the top ranked class in $\mathbb{S}_0$. We call this strategy the *separated subvocabulary* design [7].

2) The menu control scans the entire vocabulary $\mathbb{V}_0$. It might output a rejection if the top ranked class does not belong to $\mathbb{S}$; alternatively, it might examine several hypotheses, up to a certain maximum number H, starting with the top ranked one, and report a rejection if either none of these hypotheses lies in $\mathbb{S}$ or if W_0 is ranked higher than any class in $\mathbb{S}$. We call this the *embedded subvocabulary* design [7].

3) The menu control scans a subset of $\mathbb{V}_0$ containing $\mathbb{S}_0$, examining up to H hypotheses.

Let $\mathbb{E}_0$ be the set of classes scanned by the menu control (the *evaluation space*) with

$$\mathbb{S}_0 \subseteq \mathbb{E}_0 \subseteq \mathbb{V}_0.$$

Let S be the number of classes in $\mathbb{S}$ (the subvocabulary size), E the number of classes in $\mathbb{E} = \mathbb{E}_0 \setminus \mathbb{G}$ (i.e. the number of non-garbage classes in $\mathbb{E}_0$), and V the number of classes

in $\mathbb{V} = \mathbb{V}_0 \setminus \mathbb{G}$. For each recognition act, S and V are usually determined by the application, whereas the menu control can choose E and H (the maximum number of hypotheses examined) within the restrictions

$$1 \leq S \leq E \leq V, \tag{1}$$

$$1 \leq H \leq E - S + 1, \tag{2}$$

in order to optimize the recognition act with respect to a suitable criterion.

2. INTERFACE PERFORMANCE

2.1 Optimization Criteria

Let C, R, and A be the rates of confusions, false rejections, and false acceptances, visible on the user interface. In general, it will not be possible to choose E and H such that all three error rates are minimized. Therefore, we try to minimize some function of C, R, and A. Some authors propose to use weighed sums of the error rates [8] and to set the weights according to the requirements of the particular recognition act. Others measure dialogue efficiency in terms of the average number of exchanges taken or the transaction success rate [2, 3].

2.2 Deriving *C*, *R*, and *A*

As a basis for all optimization criteria which are functions of C, R, and A, we express these error rates in terms of E and H, given V and S from the application and the speech recognizer's error rates c_T, r_T, and a_T, obtained from experiments on some test vocabulary of size $T > 1$. We base the derivation on five assumptions reasonable in the absence of other evidence.

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

Under these assumptions, there exists a characteristic triple $\mathbf{U} = (u_1, u_2, u_3)$ of constants, describing the speech recognizer independently from vocabulary size, with (see Appendix 1):

$$u_1 = \frac{c_T}{(T-1)(1-c_T-r_T)}, \tag{3}$$

$$u_2 = \frac{r_T}{1-c_T-r_T}, \tag{4}$$

$$u_3 = \frac{a_T}{T(1-a_T)}. \tag{5}$$

Now we consider the examination of the i-th hypothesis W_{h_i}. If $\mathbf{p}$ is a non-garbage pattern, let $p_i^{\mathbb{S}}$ be the probability that W_{h_i} is wrong but belongs to $\mathbb{S}$, let $p_i^{\mathbb{E}\setminus\mathbb{S}}$ be the probability that it belongs to $\mathbb{E}\setminus\mathbb{S}$, and $p_i^{\mathbb{G}}$ the probability that it is garbage. If $\mathbf{p}$ is a garbage pattern, let $q_i^{\mathbb{S}}$ be the probability that W_{h_i} belongs to $\mathbb{S}$, and $q_i^{\mathbb{E}\setminus\mathbb{S}}$ the probability that it belongs to $\mathbb{E}\setminus\mathbb{S}$. We assume that 4) the user does not commit errors, and therefore a correct non-garbage hypothesis always belongs to the active subvocabulary, and 5) a wrong non-garbage hypothesis falls into each of the remaining classes with equal probability. If, for the purpose of convenient notation in the sequel, we set $p_0^{\mathbb{E}\setminus\mathbb{S}} = 1$ and $q_0^{\mathbb{E}\setminus\mathbb{S}} = 1$ (note that there is no 0-th hypothesis), we find for $i \leq H$ (see Appendix 2):

$$p_i^{\mathbb{S}} = \frac{(S-1)u_1}{1+(E-i)u_1+u_2} \quad \text{if } i > 0, \tag{6}$$

$$p_i^{\mathbb{E}\setminus\mathbb{S}} = \begin{cases} 1 & \text{if } i = 0, \\ \dfrac{(E-S-i+1)u_1}{1+(E-i)u_1+u_2} & \text{if } i > 0, \end{cases} \tag{7}$$

$$p_i^{\mathbb{G}} = \frac{u_2}{1+(E-i)u_1+u_2} \quad \text{if } i > 0, \tag{8}$$

$$q_i^{\mathbb{S}} = \frac{Su_3}{1+(E-i+1)u_3} \quad \text{if } i > 0, \tag{9}$$

$$q_i^{\mathbb{E}\setminus\mathbb{S}} = \begin{cases} 1 & \text{if } i = 0, \\ \dfrac{(E-S-i+1)u_3}{1+(E-i+1)u_3} & \text{if } i > 0. \end{cases} \tag{10}$$

From these probabilities, C, R, and A can be estimated as follows (see Appendix 3):

$$C = \sum_{i=1}^{H} p_i^{\mathbb{S}} \prod_{j=0}^{i-1} p_j^{\mathbb{E}\setminus\mathbb{S}}, \tag{11}$$

$$R = \sum_{i=1}^{H} p_i^{\mathbb{G}} \prod_{j=0}^{i-1} p_j^{\mathbb{E}\setminus\mathbb{S}} + \prod_{i=1}^{H} p_i^{\mathbb{E}\setminus\mathbb{S}}, \tag{12}$$

$$A = \sum_{i=1}^{H} q_i^{\mathbb{S}} \prod_{j=0}^{i-1} q_j^{\mathbb{E}\setminus\mathbb{S}}. \tag{13}$$

2.3 Optimization Algorithm

Now the optimization algorithm is straightforward: Given the speech recognizer's performance in terms of c_T, r_T, and a_T, the vocabulary size V of the application, the subvocabulary size S for the particular recognition act, and a function O of C, R, and A, i.e. the optimization criterion, the value of which shall be minimized; then the menu control calculates, for all combinations of E and H according to equations (1) and (2), the values of C, R, and A as shown in Section 2.2, and selects

$$(E,H)_{opt} = \underset{(E,H)}{argmin}\ O(C,R,A)$$

as the optimum solution.

3. EXAMPLE

Consider a voice control system with a total vocabulary size of $V = 100$, based upon a speech recognizer the test results of which are $c_{10} = 0.005$, $r_{10} = 0.03$, and $a_{10} = 0.20$. According to equations (3) to (5), the recognizer has a characteristic triple $\mathbf{U} = (0.00058, 0.03109, 0.02500)$.

The system shall dispose of a sleep mode, i.e. a menu of size 1, in which only a dedicated wake-up-command is accepted, and various menus of different sizes to control a number of electronic devices with their respective sets of commands. Systems of this type are described in [6].

For the optimization criteria, we follow [8]:

$$O = \lambda_c C + \lambda_r R + \lambda_a A,$$

$$\lambda_c + \lambda_r + \lambda_a = 1.$$

In the sleep mode, confusions are impossible, since $\mathbb{S}$ consists of only 1 word. Therefore, we set $\lambda_c = 0$ and put the emphasis on the avoidance of false acceptances by setting $\lambda_r = 0.2$ and $\lambda_a = 0.8$.

In a typical device menu of size 30, we do not care of false acceptances. There, confusions may be slightly more disturbing than false rejection are, such that we set $\lambda_c = 0.6$, $\lambda_r = 0.4$, and $\lambda_a = 0$.

In Tables 1 and 2, we show the results for various settings of E and H: the separated subvocabulary design ($E = S$, $H = 1$), two designs of embedded subvocabularies ($E = V$, with $H = 1$ and $H = 10$, respectively), and the self-optimized interface with $(E,H)_{opt}$ according to Section 2.3.

	C	R	A	O
separated $E = S, H = 1$	0.0000	0.0302	0.0244	0.0255
embedded $E = V, H = 1$	0.0000	0.0810	0.0071	0.0219
embedded $E = V, H = 10$	0.0000	0.0302	0.0237	0.0250
optimized $E = 100, H = 2$	0.0000	0.0328	0.0122	0.0163

Table 1: Interface performance in the sleep mode.

	C	R	A	O
separated $E = S, H = 1$	0.0159	0.0297	0.4286	0.0214
embedded $E = V, H = 1$	0.0153	0.0656	0.2143	0.0354
embedded $E = V, H = 10$	0.0159	0.0297	0.4283	0.0214
optimized $E = 34, H = 5$	0.0159	0.0297	0.4286	0.0214

Table 2: Interface performance in the device menu.

Among the fixed designs, the one that performs best in the sleep mode performs poorly in the device menu and vice versa. Note that in the sleep mode, $E_{opt} = V$, due to the emphasis on false acceptance avoidance (as a general rule, the evaluation space for keyword activation shall be large, thus enhancing the garbage model). For large subvocabularies with emphasis on correct treatment of keywords (Table 2), most designs perform near optimum, except those with H so small compared to $E - S$ that out-of-subvocabulary rejections dominate (second row).

4. CONCLUSION

We presented a self-optimization algorithm for voice control user interfaces. Given the speech recognizer's performance figures, the vocabulary size of the application, and, for each recognition act, the size of the respective subvocabulary and an arbitrary optimization criterion based on error probabilities, a menu control unit located in the user interface sets two search parameters – the evaluation space size and the maximum number of hypotheses examined – to their optima with respect to said criterion. The algorithm relies on assumptions reasonable in the absence of other evidence, and can be applied to all systems based on isolated word recognition and featuring menus without delayed decisions. It does not take into account the a priori probability of commands; however, this aspect could easily be integrated into the formalism.

APPENDIX 1

We view the recognition of a non-garbage pattern belonging to class W_k as an experiment [4], the outcomes of which are the class indices $j = 0, \ldots, T$. To each outcome j, we assign a probability $p(j)$, which is either $e_1(T)$: the probability of correct recognition, or $e_2(T)$: the probability of confusion into a specific class, or $e_3(T)$: the probability of rejection.

$$\begin{aligned} p(k) &= e_1(T), \\ p(i, i \notin \{k, 0\}) &= e_2(T), \\ p(0) &= e_3(T), \end{aligned}$$

with

$$e_1(T) + (T-1)e_2(T) + e_3(T) = 1. \tag{14}$$

Confusion rate and false rejection rate are given by

$$c_T = (T-1)e_2(T), \tag{15}$$

$$r_T = e_3(T). \tag{16}$$

From assumption (1) of Section 2.2 follows that $e_3(T)/(e_1(T) + e_3(T))$ does not depend on T; from assumption (2) follows that $e_2(T)/(e_1(T) + e_2(T))$ does not depend on T. Thus, both expressions are constant, and there exist constant $u_1 = e_2(T)/e_1(T)$ and $u_2 = e_3(T)/e_1(T)$, such that we can rewrite equations (14) to (16) to

$$c_T = \frac{(T-1)u_1}{1 + (T-1)u_1 + u_2}, \tag{17}$$

$$r_T = \frac{u_2}{1 + (T-1)u_1 + u_2}, \tag{18}$$

from which equations (3) and (4) follow.

Now we view the recognition of a garbage pattern as an experiment and assign to each outcome either $f_1(T)$: the probability of rejection, or $f_2(T)$: the probability of acceptance with respect to a specific class.

$$\begin{aligned} p(0) &= f_1(T), \\ p(i, i \neq 0) &= f_2(T), \end{aligned}$$

with

$$f_1(T) + T f_2(T) = 1. \tag{19}$$

The false acceptance rate is given by

$$a_T = T f_2(T). \tag{20}$$

From assumption (3) of Section 2.2 follows that $f_2(T)/(f_1(T) + f_2(T))$ does not depend on T. Thus, there

exists a constant $u_3 = f_2(T)/f_1(T)$, such that we can rewrite equations (19) and (20) to

$$a_T = \frac{Tu_3}{1+Tu_3}, \tag{21}$$

from which equation (5) derives.

APPENDIX 2

Let $\mathbb{E}$ be the non-garbage subset of the evaluation space, $\mathbb{S}$ the active subvocabulary, and W_{h_i} the i-th hypothesis ($1 \leq i \leq H$).

For a non-garbage pattern belonging to $W_k \in \mathbb{S}$, let $p_i^{\mathbb{S}}$ be the probability that the i-th hypothesis is a confusion which falls into $\mathbb{S}$:

$$p_i^{\mathbb{S}} = p(W_{h_i} \in \mathbb{S} \setminus \{W_k\}),$$

and $p_i^{\mathbb{E}\setminus\mathbb{S}}$ the probability that it falls into a non-garbage class outside of $\mathbb{S}$:

$$p_i^{\mathbb{E}\setminus\mathbb{S}} = p(W_{h_i} \in \mathbb{E} \setminus \mathbb{S}).$$

For W_{h_i} to be examined, $i-1$ hypotheses must have been confusions into $\mathbb{E} \setminus \mathbb{S}$. Then, out of the $E-(i-1)$ remaining classes in $\mathbb{E}$, $E-i$ are wrong classes in $\mathbb{E}$, $S-1$ are wrong classes in $\mathbb{S}$, and $E-i-(S-1)$ are classes in $\mathbb{E} \setminus \mathbb{S}$, such that, with assumptions (4) and (5) of Section 2.2,

$$p_i^{\mathbb{S}} = \frac{S-1}{E-i}\, c_{E-i+1},$$
$$p_i^{\mathbb{E}\setminus\mathbb{S}} = \frac{E-S-i+1}{E-i}\, c_{E-i+1},$$

from which we obtain equations (6) and (7) by substituting c_{E-i+1} from equation (17).

A non-garbage pattern is considered garbage with probability

$$p_i^{\mathbb{G}} = r_{E-i+1},$$

such that, using equation (18), we find equation (8).

For a garbage pattern, let $q_i^{\mathbb{S}}$ be the probability that the i-th hypothesis is an acceptance with respect to a class in $\mathbb{S}$:

$$q_i^{\mathbb{S}} = p(W_{h_i} \in \mathbb{S}),$$

and $q_i^{\mathbb{E}\setminus\mathbb{S}}$ the probability that it is an acceptance with respect to a class outside of $\mathbb{S}$:

$$q_i^{\mathbb{E}\setminus\mathbb{S}} = p(W_{h_i} \in \mathbb{E} \setminus \mathbb{S}).$$

For W_{h_i} to be examined, $i-1$ hypotheses must have been acceptances with respect to classes in $\mathbb{E} \setminus \mathbb{S}$. Then, out of the $E-(i-1)$ remaining classes in $\mathbb{E}$, S are classes in $\mathbb{S}$, and $E-(i-1)-S$ are classes in $\mathbb{E} \setminus \mathbb{S}$, such that

$$q_i^{\mathbb{S}} = \frac{S}{E-i+1}\, a_{E-i+1},$$
$$q_i^{\mathbb{E}\setminus\mathbb{S}} = \frac{E-S-i+1}{E-i+1}\, a_{E-i+1},$$

from which we obtain equations (9) and (10) by substituting a_{E-i+1} from equation (21).

APPENDIX 3

An observable confusion occurs, if either the first hypothesis is a confusion into $\mathbb{S}$, or it is a confusion into $\mathbb{E} \setminus \mathbb{S}$ and the second one is a confusion into $\mathbb{S}$, and so on, until eventually the H-th hypothesis is examined, which then must be a confusion into $\mathbb{S}$, yielding the rate of observable confusions:

$$C = p_1^{\mathbb{S}} + p_1^{\mathbb{E}\setminus\mathbb{S}}(p_2^{\mathbb{S}} + p_2^{\mathbb{E}\setminus\mathbb{S}}(p_3^{\mathbb{S}} + \cdots + p_{H-1}^{\mathbb{E}\setminus\mathbb{S}} p_H^{\mathbb{S}})\cdots),$$

which transforms into equation (11).

An observable false rejection occurs, if either the first hypothesis is a false rejection, or it is a confusion into $\mathbb{E} \setminus \mathbb{S}$ and the second one is a false rejection, and so on, until eventually the H-th hypothesis is examined, which then must be either a false rejection or a confusion into $\mathbb{E} \setminus \mathbb{S}$:

$$R = p_1^{\mathbb{G}} + p_1^{\mathbb{E}\setminus\mathbb{S}}(p_2^{\mathbb{G}} + p_2^{\mathbb{E}\setminus\mathbb{S}}(p_3^{\mathbb{G}} + \cdots$$
$$+ p_{H-1}^{\mathbb{E}\setminus\mathbb{S}}(p_H^{\mathbb{G}} + p_H^{\mathbb{E}\setminus\mathbb{S}})\cdots),$$

which transforms into equation (12).

An observable false acceptance occurs, if either the first hypothesis is a false acceptance with respect to a class in $\mathbb{S}$, or it is a false acceptance with respect to a class in $\mathbb{E} \setminus \mathbb{S}$ and the second one is a false acceptance with respect to a class in $\mathbb{S}$, and so on, until eventually the H-th hypothesis is examined, which then must be a false acceptance with respect to a class in $\mathbb{S}$:

$$A = q_1^{\mathbb{S}} + q_1^{\mathbb{E}\setminus\mathbb{S}}(q_2^{\mathbb{S}} + q_2^{\mathbb{E}\setminus\mathbb{S}}(q_3^{\mathbb{S}} + \cdots + q_{H-1}^{\mathbb{E}\setminus\mathbb{S}} q_H^{\mathbb{S}})\cdots),$$

which transforms into equation (13).

REFERENCES

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. London: Academic Press, 1990.

[2] B. S. Lin and L. S. Lee, "Computer-Aided Analysis and Design for Spoken Dialogue Systems Based on Quantitative Simulations," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 534–548, July 2001.

[3] Y. Niimi and T. Nishimoto, "Mathematical Analysis of Dialogue Control Strategies," in *Proceedings of EUROSPEECH 99*, Budapest, Hungary, Sept. 1999, pp. 1403–1406.

[4] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1984.

[5] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs: Prentice-Hall, 1993.

[6] W. Tschirk, "Neural Net Speech Recognizers. Voice Remote Control Devices for Disabled People," *e & i Artificial Intelligence 7/8/2001*, pp. 367–370, 2001.

[7] W. Tschirk, "User Interface Design for Voice Control Systems," in *Proc. ICEIS 2004 (submitted)*, Porto, Portugal, April 2004.

[8] L. Villarrubia and A. Acero, "Rejection Techniques for Digit Recognition in Telecommunication Applications," in *Proc. ICASSP 93*, pp. 455–458, Minneapolis, Minnesota, April 1993.