CAN CONFIDENCE SCORES HELP USERS POST-EDITING SPEECH RECOGNIZER OUTPUT?

Taku Endo, Nigel Ward* and Minoru Terada

{entaku, nigel, terada}@sanpo.t.u-tokyo.ac.jp
Mechano-Informatics, School of Information Science and Technology
University of Tokyo
113-0033 Tokyo, Japan

ABSTRACT

When dictating with speech recognition, most of the user's time is spent correcting errors. To decrease the burden we propose new editor functions specifically to speed up the correction process. The idea is to use a recognition confidence measure to predict which words are likely to be in error, to display that information to the user by highlighting suspect words, and to provide a command to let the user jump the cursor to the next suspect word. Simple experiments suggest that these functions can be valuable, even with today's speech recognizers and confidence measures.

1. MOTIVATION AND PROPOSAL

Speech dictation software, although increasing popular, is still not in wide use. One reason in the need to correct errors. To estimate the magnitude of this problem, we had a few subjects enter the same short passage in Japanese using one of the best commercial dictation systems. Although speaking the text was 3 or 4 times faster than keying it, when the time spent correcting errors was considered, there was only a small speed advantage. About 60% of the total dictation time was spent correcting errors.

Correcting an error takes three steps:

i identify an error

p position the cursor at the error

c correct the error

By highlighting words recognized with low confidence (Figure 1) we should be able to speed up step i, and by providing a "jump-to-next-likely-error" editor command we should be able to speed up step p. This paper analyzes the feasibility of these ideas.

2. CONFIDENCE MEASURES AND USERS' NEEDS

Confidence measures have recently seen remarkable developments [1]. However confidence measures, like speech recognition itself, are inherently inaccurate. This raises the question of what threshold to use for highlighting. The problem can be seen by referring to Figure 2, which illustrates the sort of distributions expected for the confidence, where the large lump indicates the scores for correctly recognized words and the striped lump, on the left, indicates the scores for incorrectly recognized words. To highlight suspected misrecognitions we need to set a threshold, but there is a trade-off since the two distributions overlap. If the confidence threshold is set too high, then many words which were in fact correct will be highlighted in error (F). On the other hand, if the threshold is too low, then many incorrect words will not be highlighted (M).

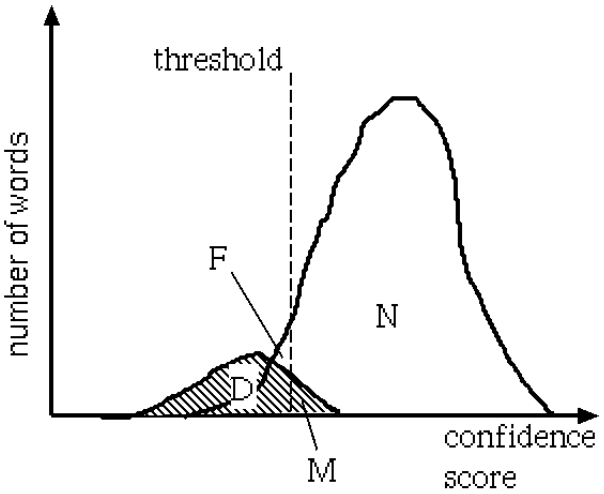

Fig. 2. Confidence Scores and Highlighting

In the figure:

D = detected errors (incorrect words which get highlighted)

F = false rejections (words which are correct but get highlighted)

M = missed errors (words which are incorrect but are not highlighted)

N = non-problematic words (which are correct and not highlighted)

To set the threshold we need to consider the user's

*Currently at the University of Texas at El Paso

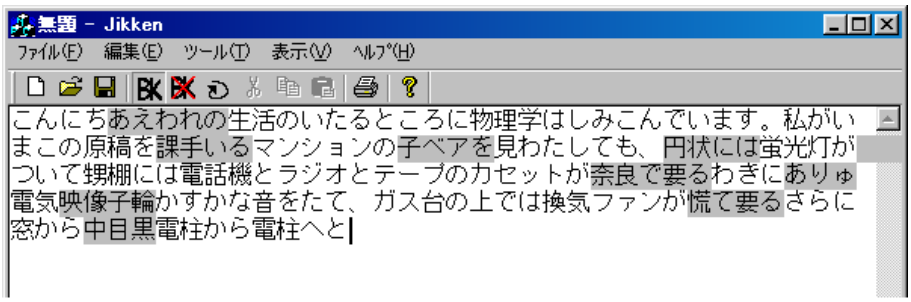

Fig. 1. Illustration of the Proposal: Words recognized with low confidence are highlighted

needs. The user is balancing two goals: to create a document with high quality (few mistakes), and to do so in a short time. Here again there is a trade-off. The relative priority given to the two goals will vary from user to user and task to task.

We quantify this trade-off with a parameter, e, representing the error penalty, that is, the cost which the user feels for one uncorrected misrecognition, expressed in seconds. In other words, e is the number of extra seconds that the user would be willing to spend to find and correct one more error. The value of e probably ranges from about 3 seconds for chatty e-mail to a friend, to 100 seconds and up for important documents.

3. MODELING THE CORRECTION PROCESS

As a baseline, we estimate the average editing time per word of text as:

$$Cost = T_i + T_p(D + M) + T_c(D + M) \qquad (1)$$

The coefficient of the first term, T_i, is 1 (= D+M+F+N) because the user scans every word. The second and third terms express the expected time to move the cursor and to correct the word.

For the proposed system, we estimate the average time cost as:

$$Cost' = T_i(D + F) + T'_p(D + F) + T_cD + eM \qquad (2)$$

The first term is smaller than in the baseline because we assume that the user does not scan all words, only the highlighted ones.

In the second term we have T'_p instead of T_p because we provide a "jump-to-the-next-likely-error" command bound to a single keystroke. The need for this arises from the special nature of editing speech recognition output. When creating text from the keyboard, users usually correct mistakes as soon as they are made, which is possible because the user can immediately feel a misstroke or see when a word is wrong. When dictating, however, users do not have immediate feedback, and so they typically do not start editing until an entire paragraph has been input [2]. Step p, positioning the cursor at an error, thus takes a non-negligible amount of time. The importance of speeding up cursor positioning is, incidentally, the reason that we propose that highlighting be binary, rather than using shades of color proportional to the confidence score. Although binary highlighting gives less information to the user, it does make the "jump-to-the-next-likely-error" command possible.

The third term is smaller than in the baseline because we assume that the user does not correct any non-highlighted words.

The last term, eM, is the time penalty ascribed to the presence of errors which the user missed since they were not highlighted.

4. SETTING THE THRESHOLD

The threshold for highlighting should be set so as to minimize the overall cost, $Cost'$. For any given recognition rate, R, we have $D = 1 - R - M$ by definition. Substituting into Equation 2 we see that $Cost'$ is minimized when

$$\frac{F}{M} = \frac{e - (T_i + T'_p + T_c)}{T_i + T'_p} \qquad (3)$$

To illustrate what this means, we consider the case of the first author, an experienced user of text-editors. For him, based on some self-observations of the typical number of keystrokes for the various steps when editing Japanese text, and assuming a typing rate of .25 seconds per keystroke, we estimate that $T_i = .8$, $T_p = 1.0$, $T_c = 4.0$, and $T'_p = .2$. For editing lab-internal e-mail we estimate e as 10. Thus for this task and this user $\frac{F}{M}$ should be about 5, meaning that optimal performance should be obtained here when there are about 5 words incorrectly highlighted for every 1 word incorrectly not highlighted. When writing a term paper, assuming $e = 50$, the ratio should be 45 to 1. Whether this is appropriate in practice depends on the quality of the confidence measure.

5. PREDICTING UTILITY

We predict that if

$$Cost' \leq Cost \qquad (4)$$

then the proposed editor enhancements should be useful.

To determine whether current confidence measures are good enough to make the proposal usable, we examine Wessel et al.'s results for the recognition of the North-American

Broadcast News corpus with 64K vocabulary [4]. They report a recognition rate R of 88.9% and a confidence measure with the trade-off curve seen in Figure 3. To convert to their notation, we use:

$$\text{false acceptance rate} = FAR = \frac{M}{D+M} = \frac{M}{1-R} \quad (5)$$

$$\text{false rejection rate} = FRR = \frac{F}{F+N} = \frac{F}{R} \quad (6)$$

For Wessel's recognition rate, comparing Equations 1 and 2, and using the parameters given above (since we expect that the values for English are roughly similar), we obtain the condition:

$$FRR \leq .9 + (.62 - .12e)FAR \quad (7)$$

That is, highlighting and the jump function will be useful if there is a point on the confidence measure's detection trade-off curve which satisfies this equation. We illustrate this in Figure 3, where the lines indicate the criterion for usability for various values of e.

If the trade-off curve falls to the left of an e line, our proposal is predicted to be worthwhile when implemented with Wessel's algorithm for a user creating documents with that e value. This predicts, for example, that the threshold can be chosen so that even a user who is willing to spend 100 seconds to find and correct one error is still likely to find highlighting to be of value.

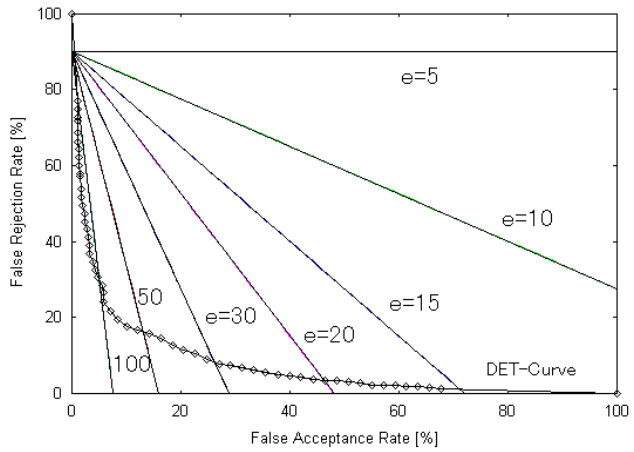

Fig. 3. The Effectiveness of the Proposal as a Function of e, the FAR and the FRR, for a recognition rate of 88.9%

6. MEASURING THE ACCURACY OF THE MODEL

We performed a small study to evaluate our model, specifically whether it accurately predicts the time-speedup of highlighting and whether it accurately predicts user satisfaction.

We seeded various texts with errors and had users correct them using three systems: A. a normal editor, B. a system with highlighting, and C. a system with highlighting plus the "jump-to-next-likely-error" command. Highlighting was set to be inaccurate sometimes; in these cases we requested subjects to only correct those errors which were highlighted.

The system was built on mule, an emacs derivative. We introduced errors at rates corresponding to recognition rates of 71% to 90%, and added highlighting corresponding to false alarm rates (F) of 19% to 33%, and miss rates (M) of 3% to 10%. D varied from 10% to 20% and N varied from 44% to 70%. From these parameters we predicted, in each case, what the speed-up in editing would be, and if positive (that is, if the speed-up ratio was greater than 1), we predicted that users would consider the new functions valuable, as per the model.

We did a total of 44 runs, each involving all three systems, with 10 subjects. Details are given in [3].

6.1. Time

We measured the total time subjects took to correct all errors with both systems. To avoid the effects of individual differences in typing speed, we measured the speed-up obtained when using the new editor functions, expressed as a percentage of the baseline editing speed. To avoid having to compensate for differences in difficulty the text and the error locations were fixed for each run. To correct for the fact that subjects got faster with each repetition, we had subjects use some systems twice: thus within a run the subject would use systems in the order ABCBA, CBABC and so on. For systems used twice we used the average time of the two runs.

While there was substantial variation, on average corrections were faster when using the editor with highlighting, as seen in Table 1.

However the speed-up was much less than predicted. Surprisingly the jump function contributed no speed-up, however this is probably in part because the jump function was unfamiliar and so was not much used, as some subjects remarked.

Editor Features Added	Speedup over Baseline	Predicted Speedup
Highlight	1.13	1.46
Highlight and Jump	1.10	1.96

Table 1. Average Speed-ups (Ratio of Correction Time without/with these features)

6.2. Satisfaction

We also solicited subjects' opinions regarding the utility of the proposed functions, asking them to imagine using them for the editing of dictated e-mail. Ranking was on a 5 point scale.

We also used Equations 1 and 2 (or a version where

$T'_p = T_p$ if the jump function was not provided) to predict whether or not the new functions would be found useful in each case. For this we used a value of 5 seconds for e, which seemed appropriate for e-mail.

Table 2 shows the results. There is a clear correlation between the utility predictions of the model and subjects' rankings, although the model tends to over-estimate utility.

Predicted Usefulness	Subjects' Ratings				
	1	2	3	4	5
useful	6	9	13	33	3
not useful	4	11	9	0	0

Table 2. Numbers of Times the Proposed Function(s) Received the Various Ratings. For example, of the runs where users gave a rating of 1, meaning unusable, the model predicted that 4 would be judged not useful, and (incorrectly) that 6 would be judged useful.

7. DISCUSSION

The model has many obvious flaws. For example, our estimates for T_i, T_p, T'_p, T_c and e are very rough; the model needs to be refined to handle individual differences; and the model assumes that speech recognition errors are always simple substitutions, where an incorrect word substitutes for a correct one. There are also some non-obvious problems:

Subjects told us they were dissatisfied if some important word was in error but not highlighted. This was a significant cause of incorrect predictions (Table 2). Perhaps content words or words otherwise apparently more important should be given a lower threshold for highlighting.

In a preliminary experiment, for one subject highlighting actually increased editing time; he claimed that the highlighting made the text harder to read. While this was probably an idiosyncrasy, it is probably also inaccurate to use the same parameter T_i for the time to judge correctness of words in both the baseline case and in the highlighted case. Certainly it seems that reading 100 adjacent words in a text is probably faster than reading 100 words scattered across the text.

As the fraction of highlighted words increases, the assumption that users will see only the highlighted ones becomes problematic, and the relative utility of highlighting probably decreases. Referring to graph 3, this means that the iso-utility e curves are not actually linear, we suspect that they bend left as FRR increases above 30% or so.

The idea that users are willing to correct errors unless it is too time-costly, represented with parameter e, may not be the best model of user behavior. It may be that users have instead some criterion of desired document quality, and continue editing until they judge that the number of remaining errors is few enough.

Thus the model has much room for improvement. Rather than resolving these issues, however, we think that actually implementing and testing the functions has higher priority. In practice, the time required to compute an accurate confidence measure may be an issue.

8. SUMMARY

We propose that correcting speech recognition output can be easier if done with an editor which highlights words which are likely to be incorrect and which provides a command for jumping directly to these words.

We also propose a model of editing costs and show how this enables approximate prediction of the conditions under which these editor functions are useful, depending on three parameters: speech recognition rate, confidence measure quality, and user tolerance for uncorrected errors.

Although preliminary, there is evidence that the proposed functions can speed up editing and be valued by users.

9. REFERENCES

[1] Chin-Hui Lee: Statistical Confidence Measures and their Applications. In Proc. ICSP 2001, Taejon, Korea, pp.1021-1028, 2001.

[2] Claire-Marie. Karat, Christine Halverson et al.: Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems. In Proc. CHI '99, pp.568-575, 1999.

[3] Taku Endo, Nigel Ward and Minoru Terada. Displaying Speech Recognition Confidence Information to Support User Correction of Misrecognitions (in Japanese). 38th Spoken Language Processing Workshop, Information Processing Society of Japan. pp 29–36. 2001.

[4] Frank Wessel, Ralph Schlüter et al.: Confidence Measures for Large Vocabulary Continuous Speech Recognition. In IEEE Trans. Speech and Audio Processing, 9, pp.288-298, 2001.