

Confidence-Scoring Post-Processing for Off-Line Handwritten-Character Recognition Verification

John F. Pitrelli and Michael P. Perrone

IBM T. J. Watson Research Center, 1101 Kitchawan Rd., Yorktown Heights, NY 10598 U.S.A.
{pitrelli,mpp}@us.ibm.com

Abstract

We apply confidence-scoring techniques to verify the output of an off-line handwritten-character recognizer. We evaluate a variety of scoring functions, including likelihood ratios and estimated posterior probabilities of correctness, in a post-processing mode, to generate confidence scores. Using the post-processor in conjunction with a neural-net-based recognizer, on mixed-case letters, receiver-operating-characteristic (ROC) curves reveal that our post-processor is able to reject correctly 90% of recognizer errors while only falsely rejecting 18.6% of correctly-recognized letters. For isolated-digit recognition, we achieve a correct rejection rate of 95% while keeping false rejection down to 8.7%.

1. Introduction

While high-accuracy character recognition has been achieved, in some applications even the few errors which are made are extremely costly. For example, in processing financial forms, mistaking *e.g.* a money amount or an identification number creates transaction errors which ultimately must all be detected and corrected manually. Fortunately, however, some such accuracy-sensitive applications are of so high a volume that automating even a fraction of the processing provides substantial benefits. This situation suggests the use of a **recognition verification** strategy: compute a measure of recognition confidence, and if it is above a threshold, accept the recognition result, thereby automating the processing. When confidence is below threshold, the item is “rejected”, meaning that the recognition result is assumed to be unreliable and the item needs to be processed manually. Essentially we are sacrificing a fraction of the automation in return for higher accuracy on the portion of processing which remains automated; as the threshold is increased, automation goes down but, typically, accuracy on the automated processing goes up. For any given application, the threshold is determined by weighing the cost of an

erroneous automation against the cost of manual processing of the rejected items. Such a verification strategy can be extremely useful for an accuracy-sensitive application that does not require 100% automation, acting somewhat analogously to Sarkar’s “triage” [12], though that functions at the page level. Alternatively, confidence measures may be used to prioritize human verification of the recognition results.

Achieving these goals requires a confidence scoring method capable of assigning generally higher confidences to correct recognition results than to incorrect ones. Confidence scoring may consist of a simple function of appropriate parameters drawn directly from the recognition process, or it may be considered a learning task in which a classifier is trained to use an array of such parameters to distinguish correct recognition from incorrect. For the sake of modularity and re-usability, our approach to computing confidences is architected to be independent of (1) which measure of confidence is used, (2) which level of unit is recognized – character, word, phrase etc., (3) which level of unit is to have recognition confidence measured, (4) character set, (5) writer-independent vs. writer-dependent, etc., and (6) specifics of the recognition algorithm, including whether it is for on-line or off-line handwriting recognition. An example of this flexibility is that recognition algorithms must have distinct ways of treating characters and words, while the confidence scoring need not. For reasons such as these, we do not integrate confidence scoring with recognition, as in some other studies [5]; rather, we choose a **post-processing** approach to confidence scoring [1], in which confidence is measured following recognition. Further reason to architect confidence scoring as a module as independent as possible of the recognizer include code modularity and re-usability, *e.g.* re-using an object-oriented post-processor across various recognizers and applications. In fact, the post-processor applied here was originally used for on-line handwriting recognition [11] [10]. The drawback of such flexibility is the loss of use of the more detailed data from the recognition process that could have been exploited by tailoring confidence scoring to the specifics of a particular recognizer. We also do not employ techniques which

require modifying the recognition process itself. Rather, we make the fairly general assumption that an unmodified recognizer provides the post-processor a graph containing one or more **recognition hypotheses**, each consisting of (a) a text label, (b) indices identifying which portion of the input handwriting corresponds to the label, and (c) a **recognition score** representing the recognizer's judgment of how well the label matches the handwriting. The recognition score is typically an estimated probability or likelihood (probability density), but it may be any other numerical quantity. For the hypothesis with the highest recognition score, the post-processor computes a **confidence score** as a function of the data in the set of hypotheses.

Section 2 describes the recognizer used in this study. Section 3 outlines applicable measures of confidence explored in the handwriting and speech recognition literature. Section 4 describes the handwriting data used in the experiments. Section 5 describes evaluation methodology, Section 6 details digit-verification experiments, and Section 7 describes letter-verification experiments. Finally, Section 8 draws conclusions and suggests future work.

2. Handwriting recognition system

In this paper, the character recognition method follows that of Mao *et al.* [7] and Gopisetty *et al.* [2]. Each character image in a given data set is normalized to a 24x16 bitmap. From each bitmap, 208 features are extracted, including contour-direction and bending-point features. The contour-direction features are extracted by scanning a 2x2 pixel mask (for each possible scan offset) over the normalized bitmap in each of four directions (vertical, horizontal and two diagonals). The 2x2 regions are primitive features which are binned into four categories. For each scan direction and offset, a tally is made of the 2x2 categories observed. These tallies are the contour-direction features. The bending-point features (such as high-curvature points, terminal points and fork points) and their corresponding attributes (such as acuteness, position, orientation, convexity and concavity) are grouped into one of 12 categories. The bending-point categories are extracted from each of 12 sub-regions from each of the normalized bitmaps. These features were used to train a multi-layer perceptron (MLP) with 208 input units and 40 hidden units. All experiments reported here are based on running the recognizer in American English writer-independent mode. For digit recognition, 10 output units are used, one for each digit class. Letter recognition for this recognizer is used for case-insensitive applications; therefore, for each of the 11 letters c, k, o, p, s, u, v, w, x, y, and z, the two cases are collapsed into single output units due to the similarity of the upper- and lower-case versions of these letters, while the other 15 letters have distinct upper- and lower-case output units,

thereby creating a total of 41 output units. The MLP was trained using standard gradient descent; for digits, 10,000 occurrences were used, 1000 per digit, and for letters, approximately 26,000 occurrences were included, with proportions skewed to reflect usage frequency in English.

3. Measures of recognition confidence

In this study, we perform both recognition and confidence measurement at the character level, so hypothesis graphs here are simply lists of single-character hypotheses, precluding the need for certain complex confidence measures such as multi-level (*e.g.* word confidence derived from character- and word-level parameters) [10] or hypothesis-lattice-based [6] parameters. Our post-processor includes eight confidence measures to explore for this task:

1. **Recognition score:** The "raw" recognition score for the best-scoring hypothesis provides some evidence of recognition confidence [3].
2. **Likelihood ratio:** The recognizer's MLPs are designed to output probability estimates for hypotheses. Therefore, the recognition score for the best hypothesis divided by the second-best hypothesis's score is a likelihood ratio, a measure of confidence in the best hypothesis relative to the closest competitor [3].
3. **Estimated posterior probability:** While the MLPs provide probability estimates, nothing guarantees that those estimates add to 1. Furthermore, the recognizer only outputs $N = 3$ best hypotheses. Therefore, we compute an estimated posterior probability for the best hypothesis, h_1 , by scaling the probability as a posterior based on treating the hypotheses $h_k, k = 1, 2, 3$ as a complete set, analogously to Stolcke *et al.* [13]:

$$P(h_1) = \text{score}(h_1) / \sum_{k=1}^N \text{score}(h_k) \quad (1)$$

4. **Negative Entropy:** Entropy in the same N -best list is a measure of recognition uncertainty. Accordingly, we compute estimated posterior probabilities $P(h_k)$ for each hypothesis as above, and define a negative-entropy confidence measure [3]:

$$\text{NegativeEntropy} = \sum_{k=1}^N P(h_k) \log_2 P(h_k). \quad (2)$$

5. **Selectivity:** We define the recognizer's N -best list's selectivity as the probability that the first word on the list is correct while all the other words in the list are incorrect. Since it is possible that some handwriting

could have more than one interpretation (*e.g.* capital *i* vs. lower-case *l*), or none (*e.g.* scribbles, cross-outs), we relax the constraint that each input must have exactly one “true” label. By assuming each hypothesis h_k to be independent, like Perrone [9] we have:

$$\text{Selectivity} = P(h_1 \cap \neg h_2 \cap \dots \cap \neg h_N) \quad (3)$$

$$= P(h_1) \prod_{k=2}^N [1 - P(h_k)] \quad (4)$$

6. through 8. **Measures with exponentiated probabilities:** We assume that we may over-estimate confidence because some competing hypotheses are pruned by the recognizer, and because of incomplete character-shape modeling. We address this issue by raising all probabilities to a constant exponent less than 1, thus creating “exponentiated” variants of the preceding three measures [13]. We chose the exponent 0.5 as a middle ground, as too close to 1 would replicate the unexponentiated measures, and too close to 0 would approach elimination of differences among the probabilities.

We also consider the possibility that these “raw” confidence measures may convey complementary information, and so some measure which combines them may outperform each taken singly. Another MLP can be trained for this purpose, doing for English character recognition what Gorski did for French bank-check recognition [3], except without his application-dependent input units. Similar approaches have also benefitted speech recognition [14] and on-line handwriting recognition [11]. Here, we use raw confidence measures as input units to each of four additional MLPs [8], separate from the recognizer. These MLPs are trained to predict whether to reject or accept the output of the recognizer. A ninth confidence measure, which we will call **MLP**, results from averaging the outputs of these four MLPs, thereby providing a smoothing benefit, at a cost of relatively few additional parameters vs. simply using one MLP. The choice to use four MLPs was made empirically. Beyond the eight raw measures, we add as inputs indicator variables representing each of the recognizer’s output units. For example, for digit recognition, 10 such units are used, with nine of them being 0 and the one which is 1 indicating which of the 10 digits was hypothesized by the recognizer. The teaching signal is simply 1 when the recognizer was correct and 0 when incorrect, with case-only errors, such as recognizing *E* as *e*, treated as correct results for training and for evaluation, in keeping with the case-insensitive design of the recognizer. The MLPs use 10 hidden units, sigmoidal threshold units, bias units, random weight initialization, and a fixed-stepsizes mean-squared-error stochastic gradient descent training algorithm with a cross-validation stopping criterion. Other than training, no attempt was made to optimize the MLP parameters or architecture.

4. Database

We evaluate using the NIST database [4], sections HSF 0, 1, 2 and 3. These sections total 2100 writers who did not provide any of the recognizer’s training data. On average, each writer provided 57 digits and 43 letters, for a total of approximately 120,000 digits and 90,000 letters.

Data were jackknifed for the confidence MLP, with thirds of it successively serving as training, cross-validation and test sets. Results for each third as the test set were concatenated together to enable comparison with the entire set being used as the test set for the other eight measures.

5. Evaluation methodology

We treat correct and incorrect recognition outputs as the two relevant classes of input to the verification post-processor, and evaluate it according to its accuracy in classifying them accordingly for acceptance or rejection. Post-processor results therefore categorize into four outcomes:

Recognizer Behavior	Post-Processor Behavior	
	Accepts	Rejects
Correct	Correct accep. (CA)	False rej. (FR)
Incorrect	False accep. (FA)	Correct rej. (CR)

We measure the post-processor’s accuracy in terms of its rate of erroneous behavior for each class of its input as follows: rejection rate on characters which had been recognized correctly, $\#FR/(\#FR + \#CA)$, and acceptance rate on characters which had been mis-recognized, $\#FA/(\#FA + \#CR)$. These two types of verification error naturally trade off; for example, raising the rejection threshold reduces false acceptances but at the cost of increased false rejections. Therefore, for each measure, we sweep a rejection threshold across its entire range of values, plotting the two error types, acceptance rate on wrongly-recognized characters against the rejection rate on correctly-recognized characters, as a receiver-operating-characteristic (ROC) curve. A curve reaching closer to the origin indicates a superior confidence measure, one enabling low rates of both error types simultaneously.

6. Digit verification results

We computed the nine confidence measures for each character, and generated ROC curves, shown in Figure 1, focusing near the origin because the recognizer’s and post-processor’s high performance on this task keep the curves very close to the optimal “L” shape for this analysis.

We observe that the raw recognition score is the top curve in some ranges of the ROC, indicating it is outperformed by the seven other raw confidence scores, which

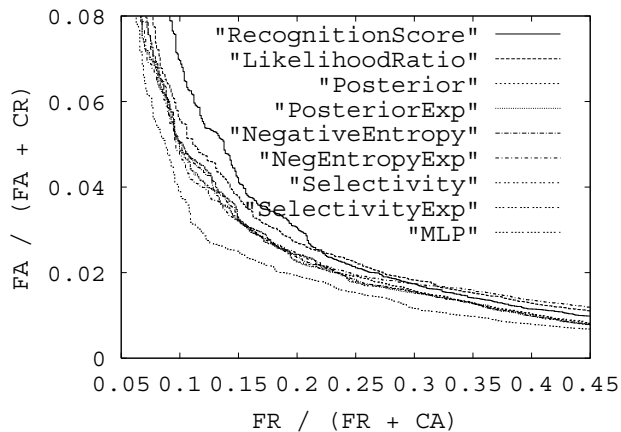


Figure 1. ROC curves for isolated-digit recognition verification using nine confidence measures. The horizontal axis shows the rejection rate on correctly-recognized digits; the vertical axis shows the acceptance rate on incorrectly-recognized digits.

in turn perform very similarly to each other. However, despite their similarity, we conclude that these measures convey complementary information, as evidenced by the lowest curve, the MLP, which achieves superior performance by incorporating all the raw measures. As stated above, the reason to use confidence scoring for recognition verification is often to achieve a particular low rate of false acceptance; thus, it is useful to compare along horizontal lines on the graphs. We observe that the raw recognition score achieves false acceptance below 5% at 13.8% false rejection; of the other raw confidence scores, the best is exponentiated negative entropy, which achieves 5% false acceptance at 9.7% false rejection, representing a 30% relative reduction ($1 - 9.7/13.8 = .30$) in the manual processing of digits rejected by the system. Finally, the MLP does best, reaching 5% false acceptance at just 8.7% false rejection, a 9% further relative improvement over the best raw confidence measure and 37% over the original recognition score. For a more stringent requirement, such as 1% false acceptance, we find that the raw recognition score requires 44.0% false rejection, exponentiated posterior probabilities 39.3%, a 11% relative reduction, and MLP, 34.2%, a further 13% relative reduction, or 22% relative to the raw recognition score.

7. Letter verification results

Letter recognition and verification is considerably more difficult than digits, because of confusable letter pairs such as l and I, o and a, etc. Even when one of these char-

acters is recognized correctly, it is difficult to achieve high confidence because competing hypotheses are likely to be assigned comparable scores by the recognizer.

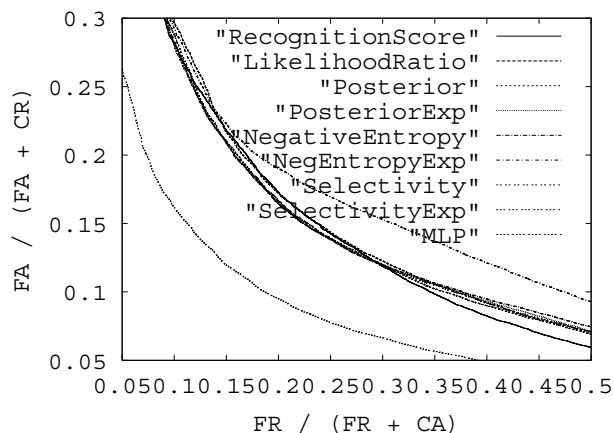


Figure 2. ROC curves for case-collapsed recognition verification on letters.

Figure 2 shows the key portion of the ROC curves. As predicted, the trade-off between false acceptance and false rejection is less favorable than with the digits set. We note that the raw confidence measures do not noticeably outperform the raw recognition score; in fact, exponentiated negative entropy performs noticeably worse, as indicated by the curve which ranges above the rest. However, the benefit of incorporated them all into confidence scoring is especially large here, as shown by the improvement realized by using the MLP, whose curve is far lower than the rest. One possible explanation for the failure of the simple measures and the success of the more-complex classifier is that this verification task is more difficult, as it must accept case-only errors between shapes which are not necessarily similar, such as e, perhaps written in a shape that can also be written as a capital, recognized as E, while rejecting other errors of perhaps comparable difference in character shape, such as o, perhaps written with a spurious tail, mis-recognized as a. To obtain a false-acceptance rate under 20% now requires increasing false rejection to 17.0% using the raw recognition score, 15.7% using the best of the other seven raw confidence scores, negative entropy, and 7.4% using the MLP. To achieve false acceptance below 10% requires 34.6% false rejection using raw rejection scores, no less than 35.9% using other raw confidence scores, here, selectivity, and 18.6% using the MLP. Thus, the seven confidence scores, in part of the range of the ROC, all fail singly to improve performance compared to the raw recognition score, but when combined together provide a major reduction in manual processing, 46% to 53% relative reduction compared to the best raw measures of confidence.

8. Conclusions and future research

Various measures of recognizer-output confidence enable verification, the decision about whether to accept or reject the output of the recognizer. We have shown how a flexible post-processor, configured with appropriate confidence measures, can substantially reduce the false-rejection rate at any given false-acceptance rate. In this way, a system composed of the recognizer and the post-processor achieves higher accuracy than the recognizer alone, at a cost of abandoning recognition on some fraction of the input handwriting. Compared to using a raw recognition score, simple confidence measures reduce the manual labor of hand-processing recognition rejects by up to 30% at the same error rate. Combining multiple confidence measures using an MLP improves performance to a 53% labor reduction.

In the future, confidence measures computed at multiple unit levels, such as word confidence measures incorporating character-level measures, should be evaluated. Varied training schemes for the MLP should be explored, such as optimizing a figure of merit. MLPs incorporating subsets of the input measures can be explored to determine whether comparable performance can be achieved with reduced computation. Finally, a larger N -best, at least 10-best, should be used in place of the 3-best set of hypotheses for the recognizer to pass to the post-processor, in order to enrich the confidence measures which would use the larger N -best.

9. Acknowledgments

We gratefully acknowledge Doug Billings for assistance with the recognizer, Benoît Maison for assistance in developing the confidence-scoring methodology, and Jane Snowdon for assistance with the paper.

References

- [1] Chigier, B., "Rejection and Keyword Spotting Algorithms for a Directory Assistance City Name Recognition Application", *Proc. ICASSP 1992*, San Francisco, California, U.S.A., March, 1992, v. 2, pp. 93-96.
- [2] Gopisetty, S., R. Lorie, J. Mao, K. M. Mohiuddin, A. Sorin, and E. Yair, "Automated Forms-Processing Software and Services", *Journal of Research and Development*, v. 40, no. 2, 1996, pp. 211-230.
- [3] Gorski, N., "Optimizing error-reject trade off in recognition systems", *Proceedings ICDAR*, Ulm, Germany, August 18-20, 1997, v. 2, pp. 1092-1096.
- [4] Grother, P. J., *NIST Special Database 19 Handprinted Forms and Characters Database*, National Institute of Standards and Technology report, March 16, 1995.
- [5] Madhvanath, S., E. Kleinberg, V. Govindaraju, and S. N. Srihari, "The HOVER System for Rapid Holistic Verification of Off-line Handwritten Phrases", *Proceedings ICDAR*, Ulm, Germany, August 18-20, 1997, v. 2, pp. 855-859.
- [6] Mangu, L., E. Brill, and A. Stolcke, "Finding Consensus among Words: Lattice-Based Word Error Minimization", *Proceedings of Eurospeech '99*, Budapest, Hungary, September 5-9, 1999, v. 1, pp. 495-498.
- [7] Mao, J., K. Mohiuddin, and T. Fujisaki, "A Two-Stage Multi-Network OCR System with a Soft Pre-Classifer and Network Selector", *Proceedings ICDAR*, Montreal, Canada, August, 1995, v. 1, pp. 78-81.
- [8] Perrone, M. P., and L. Cooper, "When Networks Disagree: Ensemble Method for Neural Networks", in R. Mammone, ed., *Artificial Neural Networks for Speech and Vision*, Chapman-Hall, London, 1993, ch. 10.
- [9] Perrone, M. P., "Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization", Ph.D. Thesis, Brown University Institute for Brain and Neural Systems, May, 1993.
- [10] Pitrelli, J. F., J. Subrahmonia, and B. Maison, "Toward Island-of-Reliability-Driven Very-Large-Vocabulary On-Line Handwriting Recognition using Character Confidence Scoring", *Proceedings of ICASSP 2001*, Salt Lake City, Utah, U.S.A., May 7-11, 2001.
- [11] Pitrelli, J. F., and M. P. Perrone, "Confidence Modeling for Verification Post-Processing for Handwriting Recognition", *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Niagara-on-the-Lake, Ontario, Canada, August 6-8, 2002, pp. 30-35.
- [12] Sarkar, P., H. S. Baird, and J. Henderson, "Triage of OCR Output using 'Confidence' Scores", *Proceedings of SPIE/IS&T Document Recognition & Retrieval Conference*, San Jose, CA, U.S.A., January, 2002.
- [13] Stolcke, A., Y. König, and M. Weintraub, "Explicit Word Error Minimization in N-Best List Rescoring", *Proceedings of Eurospeech '97*, Rhodes, Greece, September 22-25, 1997, v. 1, pp. 163-166.
- [14] Weintraub, M., F. Beaufays, Z. Rivlin, Y. König, and A. Stolcke, "Neural-Network Based Measures of Confidence for Word Recognition", *Proceedings of ICASSP 1997*, Munich, Germany, April 21-24, 1997, v. 2, pp. 887-890.