Integrating Signal and Language Context to Improve Handwritten Phrase Recognition: Alternative Approaches

Djamel Bouchaffra Eugene Koontz V Kṛpāsundar
Rōhiṇi K Śrihari Sargur N Śrihari
{bouchaff,ekoontz,kripa,rohini,srihari}@cedar.buffalo.edu

Center of Excellence for Document Analysis and Recognition (CEDAR)
State University of New York at Buffalo
Buffalo, NY 14260, U.S.A.

Abstract

Handwritten phrase recognition is an important and difficult task. Recent research in this area has focussed on utilising language context to improve recognition performance, without taking the information from the input signal itself into proper account. In this paper, we adopt a Bayesian approach to solving this problem. The Bayesian framework allows us to integrate signal-level information from the actual input with the linguistic context usually used in post-processing the recogniser's output. We demonstrate the validity of a statistical approach to integrating these two sources of information. We also analyse the need for improvement in performance through innovative estimation of informative priors, and describe our method for obtaining agreement from multiple experts for this task. We compare the performance of our integrated signal-language model against existing "language-only" models.

Keywords: *Handwritten text recognition, Recogniser performance, Linguistic post-processing, Re-ranking, Signal & Language context, Dirichlet priors, Recogniser simulation, Bayesian methods.*

1 Introduction

Natural language is the ideal medium for human-computer interaction. With the increasing demand for pen-computing and mobile computing, handwritten phrase recognition has become one of the most important and difficult tasks facing the document recognition community. Figure 1 demonstrates the typical input and output of a phrase recognition system. The system must perform word separation, and then word recognition at each word position.[1] The recogniser outputs a list of word choices for each word position, each list constituting a confusion set of word candidates. It is the job of the post-processing module to improve the overall recognition performance by re-ranking these confusion sets with respect to signal and language information.

Recent research in this area has focussed on utilising language context to improve recognition performance. In this paper, we adopt a Bayesian approach to this problem. The Bayesian framework allows us to integrate signal-level information from the actual input with the linguistic context usually used in post-processing the recogniser's output. We demonstrate the validity of a statistical approach to integrating these two sources of information. We also introduce an innovative estimation of informative priors by using prior training corpora. We show improvement in overall system performance through their judicious use in estimating probabilities. We finally present current performance figures for this approach.

[1] Some researchers choose to maintain a loop between word separation and word recognition, for improved word segmentation at the expense of processing time. It would be useful to maintain this option, and to apply it selectively to difficult instances of input.

Word Position			
#1	#2	#3	#4
the	airliners	abrupt	anti
them	fisheries	runoff	mite
then	fasteners	enough	quite
thee	customers	sunday	write
thin	faulkners	ackroyd	unite

Figure 1: **Handwritten phrase recognition:** The above figure shows a handwritten phrase, and the rectangular word trellis output by the recogniser on this input. The vertical bars denote potential word separation points for the recogniser. The truth word at each word position, when present, is shown in bold-face. Note that the truth has not been detected at all in word position #3.

2 Four Language Models for Re-ranking

In post-processing the output of a recogniser for handwritten text, two tasks can be identified: the task of re-ranking consists of re-ordering the candidates generated by the recogniser to better conform to our language model, while the task of recovery involves *over-riding* the recogniser's candidates. Our presentation here is confined to re-ranking, although our broader research goals include both tasks [3]. We now introduce the central notions underlying re-ranking, and the notation we use to represent these notions.

We denote valid words by w_1, w_2, *etc*, and valid part-of-speech tags by t_1, t_2, *etc*. Valid words are those words present in a pre-defined system lexicon, and valid part-of-speech tags are symbolic representations of common parts of speech (such as "noun", "verb", "adjective", and so on).

A planar trellis is a matrix of words or word::tag pairs arranged at the vertices of a regular two-dimensional (*i.e.*, rectangular) grid. A path in the trellis is then a word-sequence $W = \langle w_1, w_2, ..., w_n \rangle$, or the corresponding sequence[2] of part-of-speech tags $T = \langle t_1, t_2, ..., t_n \rangle$, or the word::tag path (W, T).

The recogniser generates multiple word-candidates for each word position in the input sequence, and associates a confidence value with each of the word-candidates. We denote these confidence values or scores by s_1, s_2, *etc*. The output of the recogniser is thus in the form of a planar word trellis, with a score corresponding to each element of the trellis.

We can now define our re-ranking models in terms of their relationship to the paths in the input trellis. The problem that we tackle here can be formulated as: "Determine that path in the output trellis which is most likely to be the true path". We are currently experimenting with three statistical models for re-ranking: the Word-Tag (WT) model, the Word-Tag-Score (WTS) model, and the Syntax-Semantics-Signal (SSS) model. In keeping with the work of other researchers [10], each of these models can be interpreted as a HMM. We compare our own signal-language models against the Word n-Gram (WNG) model.

2.1 The Word n-Gram (WNG) Model

We provide the Word n-Gram model as a base-line model against which to compare our own language models. This enables us to measure the precise contribution of tag and score with regard to the error rate. We present performance figures of WNG for $n = 2$, and $n = 3$. For $n = 3$ (which corresponds to a word Markov chain of order 2), WNG determines the best word path W^* to be: $W^* = \arg\max_W \left[P(w_1) \prod_{i=2}^{n} P(w_i | w_{i-1}, w_{i-2}) \right]$. We use the flooring method [2] to handle sparseness problems in this and all the other models to be discussed.

[2]The sequence of part-of-speech tags is chosen on-the-fly by a statistical tagger [2].

2.2 The Word-Tag (WT) Model

The Word-Tag Model was described in [12]. Let Ω be the set of observations corresponding to the cartesian product $(W \times T)$. An element of Ω is denoted $(W,T) = \langle (w,t)_1, (w,t)_2, ..., (w,t)_n \rangle$ where the pair $(w,t)_i$ is composed of the word w_i and the tag t_i assigned to it in the phrase $W = \langle w_1, w_2, ..., w_n \rangle$. Our problem then consists of determining the path $(W,T)^*$ such that:

$$\begin{aligned}(W,T)^* &= \arg\max_{(W,T)} P(W \wedge T) = \arg\max_{(W,T)} [P(W|T) \times P(T)] \\ &= \arg\max_{(w,t)_i} \left[\prod_{i=1}^{n} P(w_i|t_i) \times P(t_1) \times P(t_2|t_1) \times \prod_{i=3}^{n} P(t_i|t_{i-1}, t_{i-2}) \right].\end{aligned}$$

This model can be interpreted as a HMM whose set of observations corresponds to words and whose hidden states are tags. We assume that a word depends only on its own tag, and a tag only on its two previous tags (again, a Markov chain of order 2). This is written as:

$$P(w_i|t_i, (w,t)_{i-1}, (w,t)_{i-2}, ..., (w,t)_1) = P(w_i|t_i); \quad P(t_i|t_{i-1}, t_{i-2}, ..., t_1) = P(t_i|t_{i-1}, t_{i-2}).$$

2.3 The Word-Tag-Score (WTS) Model

Estimating the signal parameters $P(s_i|w_i)$ is a very difficult task, as we will see in the next section. The Word-Tag-Score Model adopts a computationally inexpensive approach to incorporating s_i into the true path computation. Here we treat the score as a simple measure of recognition confidence, and so incorporate it as a *multiplicative weight* assigned to each word. Thus, WTS is a first approximation to the SSS model.

WTS makes the simplifying assumption that the normalised recogniser score s_i is itself representative of the probability of occurrence of the corresponding word w_i. This assumption is not unreasonable, since the recogniser generates the score s_i intending it as a measure of confidence that the word-choice w_i is the same as the "true" input word w^*. This fact, in conjunction with normalisation, makes it plausible that s_i is itself a reasonable approximation to $P(s_i|w_i)$.

Thus, the problem now reduces to determining $(W,T)^*$ such that:

$$(W,T)^* = \arg\max_{(W,T)} [S(W,T) \times P(W \wedge T)] = \arg\max_{(W,T)} [S(W,T) \times P(W|T) \times P(T)] \quad (1)$$

$$= \arg\max_{(w,t)_i} \left[\prod_{i=1}^{n} s_i \times \prod_{i=1}^{n} P(w_i|t_i) \times P(t_1) \times P(t_2|t_1) \times \prod_{i=3}^{n} P(t_i|t_{i-1}, t_{i-2}) \right]. \quad (2)$$

2.4 The Syntax-Semantics-Signal (SSS) Model

The Word-Tag model is a *pure language model*, in that it does not deal with the recogniser score-vector $S = \langle s_1, s_2, ..., s_n \rangle$ associated with W. The score $s_i = s(w_i, t_i)$ provides signal-level information about the input, and is useful in discriminating among word-tag paths that are equally likely. In our SSS model, the score vector S is interpreted as an additional dimension in the overall probabilistic framework. We are thus interested in determining the word-tag path $(W,T)^*$:

$$(W,T)^* = \arg\max_{(W,T)} P(W \wedge T \wedge S) = \arg\max_{(W,T)} [P(S|W \wedge T) \times P(W|T) \times P(T)]. \quad (3)$$

We make the valid assumptions that a score value s_i depends only on word w_i and not on other words $w_{j \neq i}$, and that s_i is independent of the tag t_i.[3] Thus:

$$P(S|W \wedge T) = \prod_{i=1}^{n} P(s_i|w_i t_i) = \prod_{i=1}^{n} P(s_i|w_i)$$

[3] In practice, there may be an indirect dependence between s_i and t_i due to a linguistic quirk. Closed classes such as determiner and preposition tend to contain short words compared to open classes (as a corollary to Zipf's "law"). Since there can possibly be a dependence between the length of a word and the score that it gets assigned, our assumption may not hold in all situations.

Again assuming a Markov chain of order 2, we must now determine $(W,T)^*$ such that:

$$(W,T)^* = \arg\max_{(W,T)} \left[\prod_{i=1}^{n} P(s_i|w_i) \times \prod_{i=1}^{n} P(w_i|t_i) \times P(t_1) \times P(t_2|t_1) \times \prod_{i=3}^{n} P(t_i|t_{i-1}, t_{i-2}) \right]. \quad (4)$$

We have proved earlier [2] that this model is equivalent to a HMM where each hidden state is a word-tag couple and each observation is the score assigned by the recogniser to the handwritten word.

3 Parameter Estimation: Dirichlet priors

We compare two approaches to the task of parameter estimation. The first approach attempts to incorporate expert knowledge in the form of informative Dirichlet priors [4, 1]. The second approach consists of maximum likelihood estimations of optimal parameters for the model, and is a special case of the former.

The approach of Maximum *a posteriori* (MAP) Estimation involves incorporating linguistic intuitions into the estimation framework. We introduce α-terms that, in effect, "augment" the training data to reflect these intuitions.

The score vector S constitutes our observational data D, and the word::tag pairs WT form our model Θ. (More precisely, Θ constitutes the *parameters* of the model computed using the training corpus.) This fits in with our interpretation of these parameters as a HMM.

The justification behind the MAP estimation is as follows. If we treat our observational data D and our predictive model Θ as probabilistic events, we are trying to maximise $P(D \times \Theta)$. Now, we have:

$$\begin{aligned} P(D \times \Theta) &= P(D|\Theta) \times P(\Theta), \\ \textit{or: } P(S \times W \times T) &= P(S|W \times T) \times P(W \times T) \end{aligned}$$

Here, $P(D|\Theta)$ is the observation likelihood, and $P(\Theta)$ represents parameter distribution, which encapsulate prior (linguistic) information.

This implies that $P(S|W \times T)$ expands as a multinomial expression, with calls to terms of the form $P(s_i|w_i)$. Correspondingly, $P(W \times T)$ also expands as a multinomial expression, with calls to terms of the form $P((w,t)_i)$. Since these are multinomial expressions, we use the Dirichlet distribution to obtain priors. The *a posteriori* distribution of Θ is itself a Dirichlet distribution, and so the Dirichlet priors are known as conjugate priors.

Therefore, $\Theta = \langle \theta_1, \theta_2, ..., \theta_n \rangle$, where θ_i would be of the form $\theta_i := \widehat{P}(s|w)$ or $\theta_i := \widehat{P}(w|t)$ or $\theta_i := \widehat{P}(t'|t)$ for some s, w, t, or t'. All of these θ_i are computed along similar lines. Suppose for some i and j, $\theta_i := \widehat{P}(w_j|t_j)$. Then:

$$\widehat{\theta_i} = P(w_j|t_j) \simeq \frac{N_{w_j,t_j} + \alpha_{w_j,t_j} - 1}{\sum_{w \in C}(N_{w,t_j} + \alpha_{w,t_j} - 1)}. \quad (5)$$

where N_{w_j,t_j} is the number of occurrences of w_j with tag t_j in the training corpus C, and N_{w,t_j} the number of occurrences of *any* word w with tag t_j in C. The computation for the other two forms of θ_i follows this same pattern.

When $(\forall\{w_i,t_i\})\ \alpha_{w_i,t_i} > 1$, this computation is equivalent to adding $\alpha_{w_i,t_i} - 1$ "virtual samples" of the event $(w,t)_i$ to the training data. This, in turn, implies that MLE is a special case of MAP, with the special assignment: $(\forall\{w_i,t_i\})\ \alpha_{w_i,t_i} := 1$. We also note that the uniform assignment situation, where: $(\forall\{w_i,t_i\})\ \alpha_{w_i,t_i} := \alpha^* \neq 1$ is similar to MLE, although not identical. This latter case corresponds to providing a slant to the empirical distribution, without actually distorting it.

The distribution of the α's reflects our linguistic intuition about the relative probabilities of the corresponding events. We can, therefore, further refine this formulation by distinguishing between transition α's and emission α's, where these two notions refer to corresponding events in the underlying Hidden Markov Model.

4 Computing priors through agreement across multiple corpora

As we have seen, the MAP method requires us to estimate prior values α_i, that capture our linguistic knowledge about the *a priori* probabilities of transmission and emission. We have developed two corpus-based methods to achieve this objective. In the first approach, we use two corpora, one for drawing frequency counts from, and the second for creating virtual samples from.[4] In the second approach, we generalise this notion to agreement across m different corpora. From a cognitive standpoint, this corresponds to consulting multiple language experts, and arriving at a consensus. Consider a specific pattern $p_{i,j} := P(j|i)$ in the input, such as ⟨verb, determiner⟩, denoted by $\langle V, D\rangle$. We treat each of the m available corpora as a "language expert", and ask them their "opinion" $\widehat{\theta}^k_{i,j}$ about the likelihood of $p_{i,j}$ occurring in the input stream. (We use the superscript k to denote terms specific to corpus C_k.)

4.1 Bootstrapping for simulated samples

The technique of bootstrapping to obtain "more" samples from the same corpus has been described in the literature [6]. We expect to be able to use this idea to obtain several points in the space of multiple experts, in order to be able to arrive at agreement analytically.

The bootstrap method enables us to obtain an interval judgement of the expert (a "bootstrapped" interval) with respect to the pattern $p_{i,j}$. The $s^k_i(\theta_{i,1}, \theta_{i,2}, ..., \theta_{i,n})$ in the derivation that follows represents a score that the actual pattern $\widehat{\theta}^a_{i,j}$ computed from the *actual* training corpus (rather than any of the prior training corpora) falls into the bootstrapped interval constructed from corpus k. (Note that the set of (actual) parameters $\{\theta_{i,j} : j \in [1,n]\}$ are constrained by $\sum_{j=1}^{n} \theta_{i,j} = 1$.) Now, we need to compute the score s^k_i for each input pattern.

4.2 Computing α^*

Corpus C_k (expert# k), estimates the likelihood of $p_{i,j} := P(j|i)$ as:

$$\widehat{\theta}^k_{i,j} := \widehat{P}_{C_k}(D|V) \simeq \frac{N_{V,D} + \alpha_{V,D} - 1}{\sum_{t \in \mathcal{T}} (N_{V,t} + \alpha_{V,t} - 1)}.$$

Here, $N_{V,D}$ and $N_{V,t}$ are occurrence counts specific to C_k. Now, since $\widehat{\theta}^k_{i,j}$ is an estimation (by the k^{th} corpus) of the true probability $\theta_{i,j}$, we can estimate the bounds on the error, through the bootstrapped interval $[a^k_{i,j}, b^k_{i,j}]$ within which $\widehat{\theta}^k_{i,j}$ falls 68% of the time.[5] We expect that $|\theta_{i,j} - \widehat{\theta}^k_{i,j}| \approx (b^k_{i,j} - a^k_{i,j})/2$ holds.

Our problem thus reduces to computing the probability that the unknown parameter $\theta_{i,j}$, as estimated by the actual corpus, falls into the probability interval $[\widehat{\theta}^a_{i,j} - (b^k_{i,j} - a^k_{i,j})/2, \widehat{\theta}^a_{i,j} + (b^k_{i,j} - a^k_{i,j})/2] \equiv [A^k_{i,j}, B^k_{i,j}]$. Since $\theta_{i,j}$ follows the multinomial law, its informative priors follow the Dirichlet distribution. We can, therefore, compute a score $s^k_i()$ as follows:

$$\begin{aligned}
s^k_i = s^k_i(\theta_{i,1}, \theta_{i,2}, ..., \theta_{i,n}) &= P\left(\cap_{j=1}^{n} (|\theta_{i,j} - \widehat{\theta}^a_{i,j}| \leq (b^k_{i,j} - a^k_{i,j})/2)\right) \\
&= \frac{\Gamma(\sum_{j=1}^{n} \alpha_{i,j})}{\prod_{j=1}^{n} \Gamma(\alpha_{i,j})} \int_{A^k_{i,n}}^{B^k_{i,n}} \cdots \int_{A^k_{i,2}}^{B^k_{i,2}} \int_{A^k_{i,1}}^{B^k_{i,1}} \theta_{i,1}^{\alpha_{i,1}-1} \theta_{i,2}^{\alpha_{i,2}-1} \cdots \theta_{i,n}^{\alpha_{i,n}-1} d\theta_{i,1}.d\theta_{i,2} \cdots d\theta_{i,n} \\
&= \frac{\Gamma(\sum_{j=1}^{n} \alpha_{i,j})}{\prod_{j=1}^{n} \Gamma(\alpha_{i,j}+1)} \prod_{j=1}^{n} [\theta_{i,j}^{\alpha_{i,j}}]_{A^k_{i,j}}^{B^k_{i,j}}
\end{aligned}$$

The "opinions" of all the experts can be collected into one vector: $S_i := \langle s^1_i, s^2_i, ..., s^m_i \rangle$, and we can obtain agreement by choosing $\langle \alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n} \rangle^*$ so as to maximise the conditional probability of the response function Y, which takes the value 1, if and only if $\widehat{\theta}^a_{i,j}$ is accepted by all experts. This is equivalent to maximising the logit function $\pi(S_i) = \exp(\beta_0 + \sum_{k=1}^{m} (\beta_k s^k_i))$. We can further simplify the problem by

[4] We do not elaborate on this approach here, since it is still under development. The reader is welcome to contact us for details.

[5] The value of 68% is chosen to be compatible with the accepted range of $\mu \pm \sigma$ for the normal distribution, even though the distribution under consideration may not itself be normal.

assuming that none of the experts is preferable to any other (as, indeed, we should, in the absence of any other information): *i.e.*, $\beta_0 = 0; (\forall k > 0)\beta_k = 1$.

Therefore, the $\langle \alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n} \rangle^*$ computed below will now be added as the prior in Equation 5.

$$\begin{aligned}\langle \alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n} \rangle^* &= \arg\max_{\alpha_{i,j}}[P(Y=1|S_i)] = \arg\max_{\alpha_{i,j}} \pi(S_i) \equiv \arg\max_{\alpha_{i,j}} \left[\log \frac{\pi(S_i)}{1-\pi(S_i)}\right] \\ &= \arg\max_{\alpha_{i,j}}[\beta_0 + \sum_{k=1}^{m}(\beta_k s_i^k)] = \arg\max_{\alpha_{i,j}} \sum_{k=1}^{m} s_i^k = \arg\max_{\alpha_{i,j}} \sum_{k=1}^{m} s_i^k(\theta_{i,1}, \theta_{i,2}, ..., \theta_{i,n}).\end{aligned}$$

Therefore, using the Lagrange multipliers λ_j, we have:

$$\langle \alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n} \rangle^* = \arg\max_{\alpha_{i,j}} \left[\frac{\Gamma(\sum_{j=1}^{n} \alpha_{i,j})}{\prod_{j=1}^{n} \Gamma(\alpha_{i,j}+1)} \sum_{k=1}^{m} \prod_{j=1}^{n} [\theta_{i,j}^{\alpha_{i,j}}]_{A_{i,j}^k}^{B_{i,j}^k} + \sum_{j=1}^{n} \lambda_j \sum_{j=1}^{n} (\theta_{i,j} - 1) \right] \quad (6)$$

4.3 Estimating Signal Parameters

Estimating the signal parameters $P(s_i|w_i)$ is a difficult task. In order to make the signal estimation reliable, we need to collect multiple handwritten shapes for each word in the lexicon, and compute their respective scores. This task can only be achieved in practice through a recogniser simulator. The simulator allows us to control such input parameters as the "neatness" (qu) of the writing, and the percentage of connectedness (pc) in the input. We plan to incorporate the behaviour of multiple recognisers into the simulator, using Logistic Regression as the agreement function [7]. We have also developed a realistic writer model, which allows us to switch between simulated single-writer input and multi-writer input.

5 Experiments

In order to analyse the improvement made from recognition to post-processing, we have computed performance measures based on words as well as sentences. We define the Salvage of a model to be:

$$\text{Salvage} \stackrel{\text{def}}{=} \frac{(\#\text{top1 true after}) - (\#\text{top1 true before})}{\#\text{truth present anywhere}}.$$

We also provide the Sentence Count, which computes the number of occurrences of the true path W^* in the top five paths of the output, as a measure of sentence-level correctness.

We tested the models on different values of qu and pc. Figure 3 displays a part of our current results. SSS and WTS clearly outperform the other models on both Salvage and Sentence Count. The performance graphs also reflect the intuition that post-processing is unsuccessful when the recogniser is so good that there is not much room for improvement. We are therefore working on criteria for detecting the "cross-over" point, for the system to decide whether it should invoke post-processing at all.

We also note that the word trigram model has actually performed worse than the bigram model. This is likely to be due to sparseness in the training data. We are aware of existing backoff methods [8] for minimising sparseness problems. We plan to incorporate these schemes in future analyses.

6 Conclusion and Future work

The WTS model and the current SSS model already demonstrate significant improvement in performance over the other models compared here, and we expect the SSS model to do even better when we perform optimal clustering of words with respect to scores. Our immediate goals include finding such an optimal clustering, and detecting the cross-over point in the performance of the system. We are also working on implementing the MAP estimation — made possible now by the set of equations represented by Equation 6 — which can help improve the overall performance as compared to the current MLE implementation (*cf.* Figure 2).

The work reported here deals only with re-ranking word-candidates that were provided by the recogniser. The task of recovering words that were not suggested by the recogniser remains, and is already being pursued by us. We also plan to tackle the task of recognising words that are not even present in the lexicon.

the movie asks us an important question						
Word Position						
#1	#2	#3	#4	#5	#6	#7
Me	movie	asks	us	an	important	dictation
the	soviet	ask	its	can	importance	dilation
then	couple	asked	his	any	opportunity	violation
the	couple	ask	us	any	important	situation

Figure 2: **The need for incorporating priors:** The above table shows the simulation of a sentence, and the top-choice output of the post-processor module. (This module represents a partial implementation of the final SSS model, and not the full implementation described here.) The promotion of 'couple' over 'movie' is because of imbalances introduced by the corpus regarding the relative weights of $P(s|w)$ and $P(w|t)$. This can be rectified by the proper use of prior information concerning these lexical items.

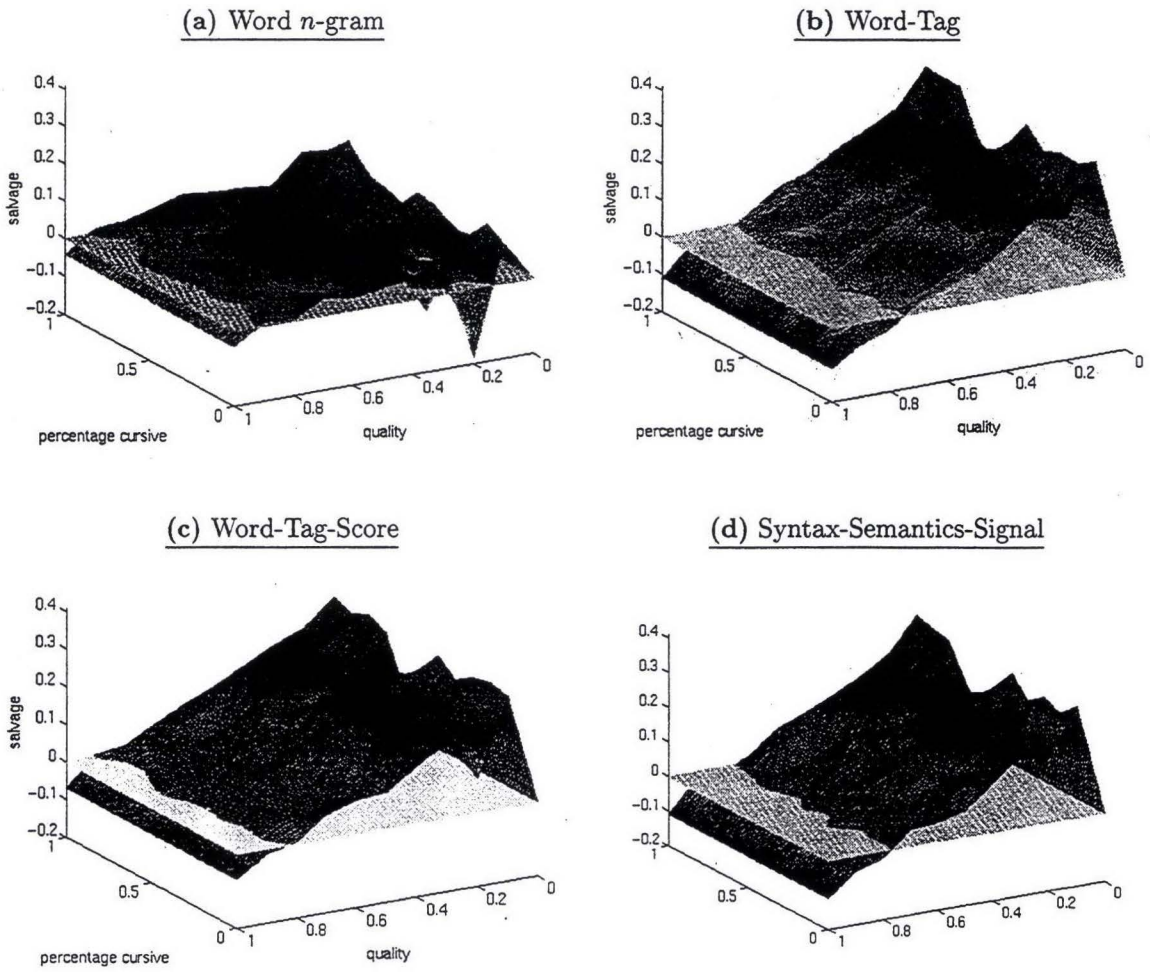

Figure 3: **Current results:** Salvage rate variation with respect to quality and percentage of cursiveness of input scripts: (a) Word bigram (b) Word Tag (c) Word-Tag-Score, and (d) SSS. The higher the salvage rate, the better the performance is. The figure shows that the performance of each model deteriorates as $qu \rightarrow 1$ (an unrealistic situation). But the SSS outperforms the other models for normal qu values. The results shown here are based on MLE.

Acknowledgements

We are indebted to Bobby Kleinberg for his ideas and coding efforts that made many of the features of the simulator possible. We also thank the reviewers of an earlier version of this paper, for their insightful comments and criticism.

References

[1] Buntine, Wray L., "Learning classification trees", in Artificial Intelligence & Statistics III, ed. D.J. Hand, Chapman&Hall, 1992.

[2] D. Bouchaffra, E. Koontz, V Kṛpāsundar and R.K. Śrihari, "Incorporating diverse information sources in handwriting recognition postprocessing", in *International Journal of Imaging Systems and Technology*, special issue, John Wiley, Vol. **7**, Issue 4, Winter 1996.

[3] D. Bouchaffra, R.K. Srihari and Sargur N. Srihari, "Reranking and Recovering Words in Handwritten Phrase Recognition", in preparation, October 1996.

[4] Cheeseman, Peter, James Kelly, Matthew Self, John Stutz, Will Taylor & Don Freeman, "AutoClass: A Bayesian classification system", in *Proc.* 5^{th} *Intnatl. Conf. Machine Learning*, pp. 54–64, Univ. Michigan, Ann Arbor, 1988.

[5] T. Dunning. "Accurate methods for the statistics of surprise and coincidence", *Computational Linguistics*, **19**:1, pp. 61–74, 1993.

[6] B. Efron. "The Jackknife, the Bootstrap, and Other Resampling Plans", *Society for Industrial and Applied Mathematics*, Philadelphia, PA, 1982.

[7] T.K. Ho, J.J. Hull and S.N. Srihari, "Decision combination in multiple classifier systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, N.1, 1994.

[8] S.M. Katz. "Estimation of probabilities from sparse data for the Language Model component of a speech recognizer", reprinted in *Readings in Speech Recognition*, A. Waibel and K.-F. Lee (eds.), Morgan-Kaufmann, 1989.

[9] W.A. Gale and K.W. Church. "Poor estimates of context are worse than none". *Proceedings of the DARPA Speech and Natural Language Workshop*, pp. 283-287, 1990.

[10] J. Kupiec, "Robust part-of-speech tagging using a hidden Markov model", in *Computer Speech and language*, vol. 6, pp. 225-242, 1992.

[11] Seni, Giovanni, Rohini K. Srihari and N. Nasrabadi, "Large Vocabulary Recognition of On-Line Handwritten Cursive Words", in *IEEE, Transactions on Pattern Analysis and Machine Intelligence* (PAMI), July 1996, vol. 18, No. 7, pp. 757-762, 1996.

[12] Srihari, Rohini K., and Charlotte M. Baltus, "Incorporating Syntactic Constraints in Recognizing Handwritten Sentences", in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93)*, Chambery, France, August 1993, pp. 1262-1267.

[13] Relevant URL's:
`http://www.cedar.buffalo.edu/Linguistics/Forms/PostProcDemos.html`
`http://www.cedar.buffalo.edu/~bouchaff/hummis.html`