Incorporating diverse information sources in handwriting recognition postprocessing

Djamel Bouchaffra*, Eugene Koontz, V. Krpā sundar, Rō hini K. Śrihari

Article first published online: 7 DEC 1998

DOI: 10.1002/(SICI)1098-1098(199624)7:4<320::AID-IMA7>3.0.CO;2-A

Issue

International Journal of Imaging Systems and Technology

Volume 7, Issue 4, pages 320–329, Winter 1996

Abstract

This article describes the proposed implementation of a new model for the linguistic postprocessing component of the Human Language Technology (HLT) project. The model was designed for handwriting recognition applications but can be used for other text recognition problems and speech recognition. We demonstrate here that the current implementation (the POS model) fails to incorporate new sources of information such as word *n*-grams, and further handles the recognizer's scores incorrectly. We propose an alternative approach (the SSS model) which remedies these shortcomings. We also show that the SSS algorithm has a direct interpretation as a Hidden Markov Model whose states correspond to words that have been tagged with their parts of speech, and whose observations are discretized recognizer confidences. The HMM interpretation has the added advantage that the approach can be naturally extended to handle error recovery of the recognizer. Preliminary results indicate that the SSS model is successful in selecting the truth path over alternate paths. © 1996 John Wiley & Sons, Inc.

Incorporating Diverse Information Sources in Handwriting Recognition Postprocessing*

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

Center of Excellence for Document Analysis and Recognition (CEDAR)
State University of New York at Buffalo
Buffalo NY 14260

Abstract

This paper describes the proposed implementation of a new model for the linguistic postprocessing component of the Human Language Technology (HLT) project. The model was designed for handwriting recognition applications but can be used for other text recognition problems and speech recognition. We demonstrate here that the current implementation (the POS model) fails to incorporate new sources of information such as word n-grams, and further handles the recogniser's scores incorrectly. We propose an alternative approach (the SSS model) which remedies these shortcomings. We also show that the SSS algorithm has a direct interpretation as a Hidden Markov Model whose states correspond to words that have been tagged with their parts of speech, and whose observations are discretised recogniser confidences. The HMM interpretation has the added advantage that the approach can be naturally extended to handle error recovery of the recogniser. Preliminary results indicate that the SSS model is successful in selecting the truth path over alternate paths.

Keywords: Handwriting recognition, language modelling, linguistic post-processing, formalising context, combining information sources, measuring influence, HMM approaches to post-processing

1 Introduction

There is currently a substantial and growing interest in the use of linguistic postprocessing techniques for recognising handwritten text. Most methods described in the literature use mainly word n-gram information. Their predictions are based on the criterion of Kullback–Liebler entropy assigned to strings [1, 2]. The bridge between the word-recognition process and the linguistic postprocessing is not always set up.

In this work we explore a new approach to modelling language for post-processing the output of a recogniser for handwritten text. Our approach (the SSS model) allows us to incorporate the recogniser's confidence scores directly into the re-ranking process. We also show that the SSS algorithm has a direct interpretation as a Hidden Markov Model whose states correspond to words that have been tagged with their parts of speech, and whose observations are discretised recogniser confidences.

In the following section we briefly review the notation that we use. In Section 3, we look at our current implementation (the POS model), and examine its effectiveness for the task at hand. We present the SSS

*This research is jointly funded by NSF and ARPA under the Human Language Technology program (Grant# IRI-9315006).

model in Section 4, and develop the theory associated with the model. This section also contains the HMM interpretation of our model, and sets the stage for our future work on Hidden Markov Mesh Fields.

Sections 5, 6 and 7 describe our estimation methods for the computation of the central terms in the SSS model. We discuss our benchmarking plans in Section 8, and outline our future work in Section 9.

2 Notation

Here we introduce the notational conventions that we use in our work. We use lower-case letters $(w, t, ...)$ to denote atomic elements such as words and tags, upper-case letters $(W, T, P, ...)$ to denote sequences, and calligraphic letters ($\mathcal{W}$, $\mathcal{T}$, $\mathcal{P}$, ...) to denote sets.

We will refer to numeric sequences as vectors. We make a notational distinction between atomic sequences such as a word-sequence $W = \langle w_1, w_2, ..., w_n \rangle$, and numeric vectors like the score-vector $S = [s_1, s_2, ..., s_n]^t$. (The superscript '$t$' denotes the transpose form of the vector.) In our application, there is always a one-to-one correspondence between the elements of a sequence and its associated vectors.

2.1 The Trellis

A trellis is a layered graph, whose vertices are words, tags, or word::tag pairs. Each layer of the trellis corresponds to a word-position in the input stream, and the vertices in the layer represent candidates for this word-position. Each pair of adjacent layers in the trellis forms a complete bi-partite subgraph of the trellis.

Thus the trellis is a compact representation of all possible transition sequences that can arise from the alternatives available from the recogniser and from the tagging process. The vertices of the trellis can be embedded in a two-dimensional matrix, with the vertices in each layer being assigned to elements in the corresponding column of the matrix.

We will refer to a particular trellis as being a word-trellis, a tag-trellis or a word::tag –trellis, according to the type of its vertices. The lexicon $\mathcal{W}$ that we use in our system is assumed to provide 100% coverage for the training and test sets. The set of tags $\mathcal{T}$ that we use is modelled after the UPenn tagset, and is a refinement of traditional English parts of speech.

2.2 The Notion of a Path

A path in the trellis is a sequence of elements such that there is exactly one element for each column in the trellis. Thus, a word::tag path $W{::}T$ is a sequence of word::tag pairs (of the form $\langle w{::}t_1, w{::}t_2, ..., w{::}t_n \rangle$) such that there is exactly one word::tag pair $w{::}t_i$ for each column i in a word::tag trellis. Thus the probability of a word::tag path with respect to a word::tag trellis is the probability $P(W|T)$ of a given path of word::tag pairs through the trellis.

2.3 Path Sets

We denote the set of all word::tag paths through a word::tag trellis by $\boldsymbol{\mathcal{P}}$. So, we have $\forall(W{::}T)\ W{::}T \in \boldsymbol{\mathcal{P}}$. For a particular vertex (i, j) in a trellis, $\boldsymbol{\mathcal{P}_{ij}} \subset \boldsymbol{\mathcal{P}}$ denotes the subset of paths that pass through it. We use double subscripts (typically i, j) to denote overall word– and tag-positions in the trellis. However, when a particular path is being discussed, we drop the *first* subscript, which indicates the rank of the word (or its tag) in the candidate-list. Thus, the notation w_i refers to the i^{th} word in the current path, and t_i refers to the tag associated with it.

3 The Current Implementation: The POS Model

The current HLT linguistic postprocessor at CEDAR, which we refer to as the Part-of-Speech (POS) Model, uses a modified version of the formulation in [3]. The POS Model maximises the following quantity:

$$P(T|W) \quad = \quad \frac{P(W|T) \times P(T)}{P(W)}. \tag{1}$$

Separating the word terms, the word::tag terms, and the pure tag terms, we write:

$$\begin{aligned} P(W|T) &= \prod_{i=1}^{n} P(w_i|t_i) \\ P(T) &= P(t_1) \times \prod_{i=2}^{n} P(t_i|t_{i-1}) \\ P(W) &= P(w_1) \times \prod_{i=2}^{n} P(w_i|w_{i-1}) \end{aligned}$$

to yield:

$$P(T|W) \quad = \quad \frac{P(w_1|t_1) \times P(t_1)}{P(w_1)} \prod_{i=2}^{n} \frac{P(w_i|t_i) \times P(t_i|t_{i-1})}{P(w_i|w_{i-1})}.$$

Since the natural logarithm is a monotonically increasing function, we can equivalently maximise the logarithm of the above quantity. The actual implementation therefore uses natural logarithms to avoid potential under-flow errors (due to machine limitations) when the input sequence is unusually long:

$$\begin{aligned} &\ln(P(w_1|t_1)) + \ln(P(t_1)) - \ln(P(w_1)) + \\ &\sum_{i=2}^{n} \big(\ln(P(w_i|t_i)) + \ln(P(t_i|t_{i-1})) - \ln(P(w_i|w_{i-1})) \big) \quad . \end{aligned} \tag{2}$$

We also make some attempt to incorporate HWR confidence $C(w_i)$ by including an additional term:

$$\begin{aligned} &\ln(P(w_1|t_1)) + \ln(P(t_1)) - \ln(P(w_1)) + \\ &\sum_{i=2}^{n} \big(\ln(P(w_i|t_i)) + \ln(P(t_i|t_{i-1})) + F \cdot \ln(C(w_i)) - \ln(P(w_i|w_{i-1})) \big) \quad . \end{aligned} \tag{3}$$

where F is a scaling factor that serves as a knob to control the relative weights of the recogniser's decision and the linguistic information that we extract from the trellis. The linguistic postprocessor ranks paths through the trellis by determining the order of the paths' relative values for this expression.

3.1 Analysis of the POS Model

Although the current formulation has been shown to improve the recognition results on average, we see at least two problems with it that detract from its performance. The research we present here revises the existing model to compensate for the flaws in that approach. We discuss these flaws now, and provide the foundation for the reasoning in the rest of the paper.

In Table 1, we show the performance of the POS Model in comparison to a reranking scheme based on word-bigrams. Although the POS Model fares slightly better, we can see from the table that it is affected by the degree of ambiguity present in the input sentence or phrase. In fact, there is no clear winner in this comparison, since the POS Model improves five of the cases, but degrades three.

Let us analyse the POS approach briefly, in order to improve upon its re-ranking strategy. The first (and central) difficulty with the POS Model is that $P(T|W)$ is not the appropriate quantity to be maximised in the task at hand. Maximising $P(T|W)$ is, of course, appropriate for traditional parsing, and for part-of-speech tagging, where the word-sequence is the given and not the goal. But here we are interested in re-ranking a trellis of words[1] so as to maximise the linguistic plausibility of the top-choice *word*-sequence. Both words and tags are variable in our application.

There are a number of sources of information, besides the syntactic tag, that can contribute to a determination of the probability of a word sequence W. Firstly, there is the handwriting recogniser (HWR)'s confidence. As the handwriting recogniser becomes more sophisticated and accurate, it is increasingly important to consider its judgement concerning the ranking of candidates. The HWR's ranking must be combined with linguistic probabilities in a theoretically sound manner.

The second problem with the current formulation is that there is no principled way to incorporate other sources of information about the best word sequence other than the tag information from the lexicon. Gale & Church [4] have shown that improper estimates of contextual influence can actually affect the performance adversely. In our new formulation, which we discuss in Section 4, the recogniser's confidence values as well as the word–tag correlations are captured in independent terms, and this allows us to incorporate new sources of information that might affect the decision of the ideal word-sequence $W^\star$.

We note that the new approach must ensure that the terms involving recogniser-confidence $C(w_i)$ are in the range $(-\infty, 0]$, in order to preserve the sign of the final computation, and to make these terms "compatible" with the conditional probability estimations. We now propose a new formulation that addresses all of the above concerns.

4 Incorporating Recogniser Information: The SSS Model

In our revised approach to linguistic post-processing (the SSS Model), we try to capture two sources of information in making the best possible decisions about the "ideal" word-sequence $W^\star$: the part-of-speech tag information $T = \langle t_1, t_2, ..., t_n \rangle$ associated with any word-sequence $W = \langle w_1, w_2, ..., w_n \rangle$, and the vector

[1] The system will eventually have the ability to propose new words (words not suggested by the recogniser) where appropriate.

$S = [s_1, s_2, ..., s_n]^t$ of scores produced by the recogniser. The score-vector S is also correlated to a vector $C = [c_1, c_2, ..., c_n]^t$, to be defined in Section 4.3. Each element c_i of the vector C represents a measure of the correctness of the recogniser's decision about the word w_i, and so we refer to C as the **correctness-vector** associated with the path W.

The correctness-vector C, and the score-vector S represent mode-specific information associated with the **S**ignal-level of the input. The sequences W and T represent respectively, **S**emantic and **S**yntactic information in the input stream. We hence refer to our overall approach as the SSS Model.

Now, we are interested in determining (from the given two-dimensional word-matrix of choices from the recogniser) $W^\star$, the word-sequence that is maximally conformant to our knowledge about the input. This is equivalent to finding a path from the path-set $\mathcal{P}$ in the word-trellis such that the word-sequence W and the auxiliary information-sources T, and C exhibit maximal mutual support. In Section 4.2 we examine the alternatives available within the SSS Model, in determining $W^\star$ from W, T, C, and S.

4.1 Phase I: Unique Tag Assignment

In the SSS Model, we adopt a two-phase strategy for solving the problem. The previous algorithm (the POS Model) was inspired by part-of-speech tagging techniques. However, there is a crucial difference between the task of tagging and that of recognition. In the former, the words in the input are fixed, and the task consists of tagging these words, and disambiguating among multiple tags where necessary. In the task of recognising natural language input, however, both words and tags are variable.

An exhaustive solution to the latter task would then be of a higher order of complexity than in the case of tagging. In our approach, we handle tags first, in a Unique Tag Assignment phase, and then handle word re-ranking as a separate phase. The assignment of unique tags in Phase I reduces the time complexity of the postprocessing task. We now look at the task of Unique Tag Assignment.

The output of the HWR is a word-trellis which can be represented as a matrix of the form $[w_{ij}]$. In Phase I, we assign unique tags to the different word-candidates in the trellis. When multiple tags are available for a particular word (as is very often the case), our algorithm will attempt to assign that tag which maximises the chance of the best-path computation to be compatible with the linguistic knowledge and recogniser confidence information available to the system.[2]

Figure 1 outlines the `UTA` algorithm used to accomplish unique tag assignment, and Figure 2 illustrates its method. For instance, when the algorithm considers the alternate tags N(ou)N and V(er)B for the word 'help' ($= w_{2,2}$) in the word-trellis, it looks at all paths $W \in \boldsymbol{\mathcal{P}_{2,2}}$, and chooses VB over NN.

4.2 Phase II: Finding the Best Path

As we noted in the beginning of this section, we are interested in determining (from the given two-dimensional word-matrix of choices from the recogniser) $W^\star$, the word-sequence that is maximally conformant to our knowledge about the input. This is equivalent to finding a path from the path-set $\mathcal{P}$ in the word-trellis such that the word-sequence W is well-supported by the auxiliary information-sources T, and C.

The notion of 'well-supported' can be captured by two alternate definitions of the quantity that we seek

[2] The uniqueness criterion states that no word-candidate can be assigned multiple tags. It is, however, possible for the `UTA` algorithm to assign the same tag to different word-candidates at a given word-position — *i.e.,* $\Diamond \exists j_0 i_1 i_2 : t_{i_1 j_0} = t_{i_2 j_0}$.

to maximise. If we consider W to be information that we are looking for, and T, C, and S to be evidence that affirms or denies W, we can define $W^\star$ to be $W^\star = \arg\max_W P(W|T.C)$.

On the other hand, we can consider them all to be mutually-supporting pieces of evidence, in which case, $W^\star = \arg\max_W P(W.T.C)$, maximising the joint distribution of the three variables. In the following sections, we compare the two approaches from a theoretical perspective, and justify our decision to choose one over the other.

4.3 The correctness-vector C

It is an empirical observation that all recognisers tend to perform poorly for shorter words. We introduce the correctness-vector C to compensate for this within a numerical framework that is based on the recogniser's performance over a training set. We define the elements c_i of the correctness-vector $C = [c_1, c_2, ..., c_n]^t$ in terms of the words w_i, their lengths l_i, and their recogniser scores s_i. We can first define a continuous correctness variable χ_i, and then discretise it to obtain c_i.

In its most general form, χ_i would be: $\chi_i = \psi(w_i, s_i, l_i) \oplus \phi(w_i, s_i, l_i)$. Here, the $\psi()$-term is meant to capture quirks of the recogniser that reflect domain-dependent features such as the word shape, and the $\phi()$-term will model external noise introduced by infinitesimal variations in the input signal. Thus, $\psi()$ captures *systematic bias* in the recogniser due to inherent limitations of the input modality, while $\phi()$ captures *random error* due to fluctuations of the input from the ideal. In fact, the noise introduced by these two components is directly responsible for producing multiple recogniser choices at any word position.

In general, the functions $\psi()$ and $\phi()$ could be of arbitrary analytical order, and the operator $\oplus$ could combine the $\psi()$ and the $\phi()$ terms in any fashion. In practice, however, we propose to compute χ_i as a multi-linear combination of the variables w_i, s_i, and l_i with the substitution:

$$\{\psi(w_i, s_i, l_i) ::= s_i; \quad \phi(w_i, s_i, l_i) ::= \eta(w_i, l_i)\}.$$

Thus, we specify the $\psi()$-term to be the recogniser's score, and the $\phi()$-term to be Gaussian[3] $\eta()$. The operator $\oplus$ is a simple weighted sum of its operands. Therefore, the expression for χ_i becomes:

$$\chi_i = \alpha s_i + \beta \cdot \eta(w_i, l_i)$$

where α and β are constants. To obtain c_i, we then discretise χ_i based on the interval that it falls into:

$$c_i = \begin{cases} (\delta_\mathbf{1})/2 & \text{if } 0 \le \chi_i \le \delta_\mathbf{1}, \\ (\delta_\mathbf{1} + \delta_\mathbf{2})/2 & \text{if } \delta_\mathbf{1} < \chi_i \le \delta_\mathbf{2}, \\ (\delta_\mathbf{2} + \delta_\mathbf{3})/2 & \text{if } \delta_\mathbf{2} < \chi_i \le \delta_\mathbf{3}, \\ \quad \cdots & \\ (\delta_\mathbf{k-1} + \delta_\mathbf{k})/2 & \text{if } \delta_\mathbf{k-1} < \chi_i \le \delta_\mathbf{k}. \end{cases}$$

The number k of the intervals, and the exact values of the cuts $\delta_\mathbf{j}$, are to be determined by clustering the scores obtained from the recogniser on multiple tokens of the same word-type. We plan to maximise inter-cluster variance, and minimise intra-cluster variance to obtain maximum homogeneity within each cluster.

[3]The $\phi()$-term expresses the quality of the input signal, and contributes to production of this word-path by the recogniser. In order to capture the error due to this random noise, we model the error-function $\eta()$ as a Gaussian distribution. The mean μ and the standard deviation σ of $\eta()$ are to be determined through training, and will depend on w_i and l_i.

4.4 Phase II – Branch A: The Conditional Distribution Approach

If we adopt the Conditional Distribution Approach, where we seek to maximise $P(W|T.C)$, we can simplify the estimation of this conditional probability as follows:

$$
\begin{aligned}
P(W|T.C) &= \frac{P(W.T.C)}{P(C.T)} \\
&= \frac{P(C|W.T) \times P(W.T)}{P(C.T)} \\
&= \frac{P(C|W.T) \times P(W|T) \times P(T)}{P(C|T) \times P(T)} \\
&= \frac{P(C|W.T) \times P(W|T) \times \cancel{P(T)}}{P(C|T) \times \cancel{P(T)}} \\
&= \frac{P(C|W.T) \times P(W|T)}{P(C|T)} \\
&= P(c_1|w_1, w_2, ..., w_n;\ t_1, t_2, ..., t_n) \prod_{i=2}^{n} P(c_i|c_1, ..., c_{i-1};\ w_1, w_2, ..., w_n;\ t_1, t_2, ..., t_n) \\
&\quad \times P(w_1|t_1, t_2, ..., t_n) \prod_{i=2}^{n} P(w_i|w_1, ..., w_{i-1};\ t_1, t_2, ..., t_n) \\
&\quad \div P(c_1|t_1, t_2, ..., t_n) \prod_{i=2}^{n} P(c_i|c_1, ..., c_{i-1};\ t_1, t_2, ..., t_n).
\end{aligned}
$$

We can now make a zero-history assumption for terms computing $P(c_i|any\ event)$, since the recogniser score or correctness for a particular word-token is not affected by c-, w-, and t-values at other word positions in the input stream. We can also make a single-history assumption for terms that compute $P(w_i|any\ event)$ and $P(t_i|any\ event)$. This yields the following expression for $W^\star$:

$$
W^\star = \arg\max_W P(w_1|t_1) \prod_{i=2}^{n} P(w_i|w_{i-1}, t_{i-1}, t_i) \times \prod_{i=1}^{n} \frac{P(c_i|w_i t_i)}{P(c_i|t_i)}.
$$

Further, we can deem c_i to be independent of the tag information t_i since the recogniser does not take any note of the POS ambiguities associated with a particular word, but only works on the input signal, namely the word shape. (Notice that c_i already incorporates length information, as described in Section 4.3.) We now have the final computation for $W^\star$ as:

$$
\begin{aligned}
W^\star &= \arg\max_W P(w_1|t_1) \prod_{i=2}^{n} P(w_i|w_{i-1}, t_{i-1}, t_i) \times \prod_{i=1}^{n} \frac{P(c_i|w_i)}{P(c_i)} \\
&= \arg\max_W P(w_1|t_1) \prod_{i=2}^{n} P(w_i|w_{i-1}, t_{i-1}, t_i) \times \prod_{i=1}^{n} \frac{P(c_i|w_i)}{\sum_{w \in \mathcal{W}} (P(c_i|w).P(w))}. \qquad (4)
\end{aligned}
$$

4.5 Phase II – Branch B: The Joint Distribution Approach

If we consider W, T, and C to be mutually-supporting pieces of evidence, we must adopt the Joint Distribution Approach, where we seek to maximise $P(W.T.C)$. In this case, we have:

$$\begin{aligned} P(W.T.C) &= P(C|W.T) \times P(W|T) \times P(T) \\ &= P(c_1|w_1, w_2, ..., w_n;\ t_1, t_2, ..., t_n) \prod_{i=2}^{n} P(c_i|c_1, ..., c_{i-1};\ w_1, w_2, ..., w_n;\ t_1, t_2, ..., t_n) \times \\ &\quad P(w_1|t_1, t_2, ..., t_n) \prod_{i=2}^{n} P(w_i|w_1, ..., w_{i-1};\ t_1, t_2, ..., t_n) \times P(t_1) \prod_{i=2}^{n} P(t_i|t_1, ..., t_{i-1}). \end{aligned}$$

Now, as with the Conditional Distribution Approach, we can make a zero-history assumption for terms computing $P(c_i|any\ event)$, and a single-history assumption for terms that compute $P(w_i|any\ event)$ and $P(t_i|any\ event)$. In the Joint Distribution Approach, these assumptions have the additional merit of being equivalent to assuming a Hidden Markov Model of order one, as we will demonstrate immediately. The expression for $W^\star$:

$$W^\star = \arg\max_W \Big(\prod_{i=1}^{n} P(c_i|w_i t_i) \times P(w_1|t_1) \prod_{i=2}^{n} P(w_i|w_{i-1}, t_{i-1}, t_i) \times P(t_1) \prod_{i=2}^{n} P(t_i|t_{i-1}) \Big)$$

can be further simplified. Here, too, we can deem c_i to be independent of the tag information t_i. Therefore, our task becomes the computation of $W^\star$, such that:

$$W^\star = \arg\max_W \Big(\prod_{i=1}^{n} P(c_i|w_i) \times P(w_1|t_1) \prod_{i=2}^{n} P(w_i|w_{i-1}, t_{i-1}, t_i) \times P(t_1) \prod_{i=2}^{n} P(t_i|t_{i-1}) \Big). \quad (5)$$

4.5.1 The Underlying Hidden Markov Model $\boldsymbol{\lambda}$

The Joint Distribution Approach is especially appealing because it has a very direct interpretation as a Hidden Markov Model. Apart from strengthening the theory through cross-confirmation from a second perspective, the HMM interpretation is also consistent with our planned approach for error recovery.[4] The two interpretations, Bayesian and HMM, are equivalent, as can be seen from the following derivation.

<u>Note</u>: To distinguish between random variables and the values they can take, we use subscripts to denote variables, and square brackets ([]) to denote their potential values. Thus, $q[3]$ represents the third state in the model, whereas q_3, which is shorthand for $q(\tau\!=\!3)$, is the state of the system at time $\tau = 3$.

In the HMM interpretation, we treat the correctness-vector C (defined in Section 4.3) as the observation-sequence of the Hidden Markov Model. The observations are therefore drawn from the observation set $\boldsymbol{\mathcal{O}} = \{o[1], o[2], ..., o[k]\}$ (where each $o[j] = (\delta_{\mathbf{j-1}} + \delta_{\mathbf{j}})/2$, as described in Section 4.3). The hidden states of the model are word::tag pairs w::t, drawn from the state set $\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{W}} \times \boldsymbol{\mathcal{T}} = \{q[1], q[2], ..., q[K]\}$. The cardinality of $\boldsymbol{\mathcal{Q}}$ is: $K = |\boldsymbol{\mathcal{Q}}| = |\boldsymbol{\mathcal{W}}| \times |\boldsymbol{\mathcal{T}}|$.

We can now define the initial probability $\pi_i = P(Q_{\tau=1} = \langle q[i] \rangle)$ that the system starts off at state $q[i]$ when time $\tau = 1$. Also, we let the matrix A contain the state-transition probabilities $a_{i,j}$ for all state-pairs

[4] In error recovery, or restoring the true word-choice when the recogniser did not succeed in including it among the candidates for the word position, we plan to use Hidden Markov Mesh Fields [6], where classes of states will act as single entities.

$(q[i], q[j])$ — *i.e.*, $A[i,j] = a_{i,j} = P(q_{t+1} = q[j] \mid q_t = q[i])$. Finally, we define the elements $b_i(j)$ of the matrix B as $B[i,j] = b_i(j) = b_{q[i]}(o[j])$ to be the conditional probability $P(o_t = o[j] \mid q_t = q[i])$ of the observation $o[j]$ occurring from the state $q[i]$ at any given time. Then, the HMM λ is defined by the triplet:

$$\lambda = \begin{bmatrix} \Pi & = & [\pi_i]^t \\ A & = & [a_{i,j}] \\ B & = & [b_i(j)] \end{bmatrix}.$$

Given this HMM model λ, we need to choose $W^\star$ such that the state sequence $Q_{\tau=n} = \langle q_1, q_2, ..., q_n \rangle$ and the observation sequence $O_{\tau=n} = \langle o_1, o_2, ..., o_n \rangle$ are maximally likely. (Note again, that the value of q_2 is that state $q[i] \in \mathcal{Q}$ which the system happens to be in at time $\tau = 2$.) We hence need to maximise:

$$\begin{aligned} P(O, Q|\lambda) &= \Pi[q_1] \times B[q_1, o_1] \times \prod_{i=2}^{n} \big(A[q_{i-1}, q_i] B[q_i, o_i]\big) \\ &= \pi_{q_1} \times b_{q_1}(o_1) \times \prod_{i=2}^{n} \big(a_{q_{i-1}, q_i} b_{q_i}(o_i)\big) \\ &= P(q_1) \times \prod_{i=2}^{n} \big(P(o_i|q_i) P(q_i|q_{i-1})\big) \\ &= (P(t_1).P(w_1|t_1)) \times \prod_{i=2}^{n} \big((P(c_i|w_i, t_i) \times (P(t_i|t_{i-1}).P(w_i|w_{i-1}, t_{i-1}, t_i))\big). \end{aligned}$$

Consequently, the expression for $W^\star$ becomes:

$$W^\star = \arg\max_W \big((P(t_1).P(w_1|t_1)) \times \prod_{i=2}^{n} \big(P(c_i|w_i, t_i) \times (P(t_i|t_{i-1}).P(w_i|w_{i-1}, t_{i-1}, t_i))\big)\big). \tag{6}$$

which is identical to Equation 5! We thus have that the Bayesian derivation with the above-mentioned history assumptions is exactly equivalent to the first-order HMM λ.

In our application, we can further assume safely that the c_i are independent of the t_i, since the recognisers that we use do not utilise any tag information in assigning scores. Thus, in both equations (5)&(6), we can replace the terms $P(c_i|w_i, t_i)$ by the corresponding terms $P(c_i|w_i)$.

We can think of the states q_i as urns containing coloured marbles. The marbles represent possible observations from the state. The values of observations at a given state can range from $\delta_\mathbf{0} = 0$ to a recogniser-dependent maximum of $\delta_\mathbf{k}$. This range can be divided into intervals using the variance-based clustering discussed in Section 4.3. We can associate colours with these intervals, such that each colour corresponds to a continuous half-open interval $(\delta_\mathbf{j-1}, \delta_\mathbf{j}]$ of the range of the observation variable χ. This in turn means that the distribution of colours in each urn captures the distribution of the observation probabilities associated with that state. Figure 3 illustrates the assignment of colours to intervals. In the figure, each marble is placed at the centre of its interval, to denote that the mean of the interval is chosen to represent it (*ie.*, $c_i = (\delta_\mathbf{j-1} + \delta_\mathbf{j})/2$ as described in Section 4.3).

Now, different urns can have identical distributions of marbles. Specifically, for a given word $w \in \mathcal{W}$, for every tag t that applies to w, all the states $w{::}t$ will share the observation probabilites. This is demonstrated in Figure 4. This corresponds to having identical rows in the matrix B. However, these states have *distinct* state-transition probabilites, and hence cannot be collapsed together.

We also note that the use of hypertags [3] is facilitated by this framework. A hypertag is essentially a cluster of tags, and so it can be modelled by a class of states, rather than by a single state of the HMM [5].

4.6 Comparing the Branches of Phase II

In the Conditional Distribution Approach, we seek to maximise $P(W|T.C)$, as contrasted to $P(W.T.C)$ in the Joint Distribution Approach. Now, we have:

$$P(W.T.C) = P(W|T.C) \times P(T.C) = P(W|T.C) \times P(T) \times P(C)$$

given the mutual independence of C and T.

Thus, the branches differ in that the Joint Distribution Approach takes explicit account of the tag-sequence probability, and the correctness-vector probability. We expect that this will not make a significant difference in computing $W^\star$.

5 Estimating the terms of $P(W|T)$

We estimate $P(W|T)$ using frequency-counts from our e-mail corpus[5]:

$$\begin{aligned} P(w_1|t_1, t_2, ..., t_n) \quad &\approx \quad \widehat{P}(w_1|t_1) \;=\; \frac{N(w_1{::}t_1)}{\sum_{w \in \boldsymbol{\mathcal{W}}} N(w{::}t_1)} \\ P(w_i|w_1, w_2, ..., w_{i-1};\; t_1, t_2, ..., t_n) \quad &\approx \quad \widehat{P}(w_i|w_{i-1}, t_{i-1}, t_i) \;=\; \frac{N(w_{i-1}{::}t_{i-1}, w_i{::}t_i)}{\sum_{w \in \boldsymbol{\mathcal{W}}} N(w_{i-1}{::}t_{i-1}, w{::}t_i)} \end{aligned}$$

where $N(x, y, z, ...)$ is the number of times that the sequence of events $\langle x, y, z, ... \rangle$ occurred in the training corpus.

The independence assumption involved in the transition from P to $\widehat{P}$ above represents a trade-off between computational expense and the accuracy of the model. The current assumption allows us to model language as a first-order HMM, which is consistent with our analysis in Section 4. Relaxing the assumption (to include a larger history) is equivalent to modelling n-grams for larger n, but this requires correspondingly larger amounts of training data to obtain reliable estimations.

As noted before, our post-processing component has to incorporate not only statistical knowledge, but also recogniser confidences for the different word-candidates. This is crucial in order to avoid over-riding the recogniser's decisions indiscriminately. The next section deals with computing $P(C|W)$, which incorporates word recognition confidences into the computation.

6 Estimating $P(C|W)$

The expression $P(C|W)$ represents the probability that a particular correctness-vector $C = [c_1, c_2, ..., c_n]^t$ is associated with the word-sequence $W = \langle w_1, w_2, ..., w_n \rangle$. Such a probability would be meaningless if we had

[5] The e-mail corpus has been built up here at CEDAR to reflect the informal nature of spontaneously generated language. The corpus is a better representative of the *genre* of online handwritten text [3].

not transformed the recogniser-score vector $S = [s_1, s_2, ..., s_n]^t$ and discretised it to obtain C. But now, with our intervals, and with our zero-history assumption for terms computing $P(c_i|any\ event)$, we can simplify $P(C|W)$ as:

$$P(C|W) = \prod_{i=1}^{n} P(c_i|w_i).$$

$P(c_i|w_i)$ can itself be estimated through one of two approaches — *Maximum Likelihood Estimation* (MLE) and *Maximum Mutual Information* (MMI). Our first implementation will be based on MLE, because of its proven success in other post-processing systems [11]. We plan to test the system in the future, using MMI estimations, and then to compare the two estimation methods for relative merit.

We assume here, as a first hypothesis, that the HWR word confidences should be given exactly equal weight with the linguistic information derived from the corpus. We can then use the recogniser training corpus to perform the following estimation (*cf.* Section 4.3):

$$\widehat{P}(c_i|w_i) \;=\; \widehat{P}(\chi_i \in (\delta_{\mathbf{j-1}}, \delta_{\mathbf{j}}] \mid w_i) \;=\; \frac{N(\chi_i \in (\delta_{\mathbf{j-1}}, \delta_{\mathbf{j}}];\ w_i)}{N(\chi_i \in (\delta_{\mathbf{0}} = 0, \delta_{\mathbf{k}}];\ w_i)} \tag{7}$$

where $N(\chi \in (\delta, \delta'];\ w)$ is the number of times the word-token w was recognised in the corpus with its χ-value in the range $(\delta, \delta']$. (The corpus will contain instances of written words that will cover the entire lexicon. It will consist of single-writer sub-corpora, but will itself contain samples from several different writers and writing styles.)

7 Sparseness and the Need for Smoothing

Sparseness of training data is a well-documented issue in statistical approaches [7]. Sparseness means that the number of training events is less than the "ideal" number of events that would be necessary for reliable statistical estimation. Sparseness is an acute problem in the case of word-level statistics, since the lexicon $\mathcal{W}$ contains tens of thousands of words. We will use smoothing techniques that have been used in the literature [8, 9, 10] to handle 0-events gracefully.

When the system considers a trigram in the trellis that was not encountered in the the training sequence, it is inappropriate to assign a probability of 0 to this unseen event. It would be preferable to assign this 0-event a small, theoretically motivated, probability value that reflects the intuitive degree to which the training corpus is "incomplete".

We plan to compare the performances of two approaches to smoothing, namely, Linear Interpolation and Flooring. We briefly describe these approaches below.

7.1 Linear Interpolation

In the Linear Interpolation method [8], we try to re-estimate n-gram probabilities as a weighted sum of probabilities of sequences of lower order. Therefore the Linear Interpolation method is applicable mainly to the sub-task of computing $P(W|T)$. Thus, we can derive the re-estimation $P(w_i|w_{i-1}, t_{i-1}, t_i)$ as:

$$\begin{aligned} P(w_i|w_{i-1}, t_{i-1}, t_i) \;&=\; \alpha_0 P_0 + \alpha_1 P(w_i) + \alpha_2 P(w_i|t_i) + \alpha_3 P(w_i|w_{i-1}) + \\ &\quad\; \alpha_4 P(w_i|t_{i-1}, t_i) + \alpha_5 \widehat{P}(w_i|w_{i-1}, t_{i-1}, t_i) \end{aligned}$$

with the constraints that $\alpha_0 P_0 \neq 0, \sum_j \alpha_j = 1$, and that P_0 is a "reasonable" constant. P_0 is often chosen to be $1/N$, where N is the size of the training corpus.

7.2 Flooring Method

The Flooring method sacrifices theoretical soundness for ease of implementation and for computational savings. It is also more broadly applicable than the Linear Interpolation method, and can be used to smooth the computation of $P(C|W)$, $P(W|T)$ and $P(T)$. Here we demonstrate the technique with the computation of $P(c_i|w_i)$.

$$P(c_i|w_i) = \begin{cases} \epsilon & \text{if } N(c_i;\ w_i) = 0 \\ \widehat{P}(c_i|w_i) & \text{otherwise.} \end{cases}$$

In this method also, ϵ can be set to k/N for some $k \in \Re$ (the field of real numbers). The motivation here is again that we want a small non-zero number that reflects the size of the training set.

8 Quantifying Performance of the SSS Model

Figure 5 shows four example input sentences, and compares the performances of the POS Model and the SSS model (the Joint Distribution Approach) on two selected paths in each trellis. The scores associated with the SSS model in this example were hand-computed, and are representative of actual scores that would be generated by the approach.

We plan to extend the performance evaluation of both the branches of the SSS model, in order to compare the relative merits of the Joint Distribution Approach and the Conditional Distribution Approach. We will perform perplexity-based comparison of five models:

1. The n-gram model — used as a baseline to compare the other models
2. The "pure" POS model (Equation 2)
3. The POS model with the recogniser-score $C(w_i)$ "added on" (Equation 3)
4. Branch A of the SSS model — the Conditional Distribution Approach (Equation 4)
5. Branch B of the SSS model — the HMM-based Joint Distribution Approach (Equation 6)

Since our goal is not just to model language for a prediction task [13], but to maximise the success of the recognition process, the classical definitions of perplexity [14] are not sufficient for our purposes. We rather need a definition that captures the overall performance of the system, with emphasis on the goal of recognition. We are in the process of developing such a perplexity measure.

We will also compare the smoothing effectiveness of the Linear Interpolation Method and the Flooring Method individually for the two branches of the SSS model. As discussed in Section 4.6, we expect the Conditional Distribution Approach and the Joint Distribution Approach to give similar results.

9 Conclusions & Future Directions

We have presented here a theoretically sound and internally coherent approach to modelling language for the task of postprocessing the output from a handwriting recogniser. Our approach has a direct interpretation as a Hidden Markov Model, and allows us to incorporate recogniser confidences into the computations in a principled manner.

The preliminary results presented in Figure 5 show that the SSS model is very effective in choosing the truth path over other candidate paths. We expect to complete the implementation of our postprocessing model very soon, and to have performance benchmarking available for the document analysis community in the near future. Further, we have reason to believe that the methodology is directly applicable to other input modalities such as speech.

In terms of future work, limits of the SSS model need to be explored and established. For instance, there are semantic relationships (such as collocations and triggers) between words in text which cannot be captured directly in the $P(W|T)$ terms. These types of lexical relationships can also be captured by modelling the trellis as a Markov Random Field (MRF) [6]. We are already investigating the role that MRF's can play in conjunction with the SSS model to yield a new unified approach.

Acknowledgements

The authors would like to thank Professor S.N. Srihari and Dr. J. Favata for their discussions regarding the paper.

References

[1] D. Ron, S. Singer and N. Tishby. "The power of amnesia", in *Advances in neural information processing systems*, vol. **6**. J. Cowan *et al.* (eds.), Morgan-Kaufmann, 1994.

[2] I. Guyon and F. Pereira. "Design of a linguistic postprocessor using variable memory length Markov Models". Technical Memorandum, AT&T Bell Labs, 1995.

[3] R. Srihari and C. Baltus. "Incorporating Syntactic Constraints in Recognizing Handwritten Sentences", in *Proceedings of IJCAI-91*, 1991.

[4] 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.

[5] D. Bouchaffra and J. Rouault. "Capturing observations in a nonstationary hidden Markov models", in *Selecting Models from Data*, pp. 263–271, P. Cheeseman and R.W. Oldford (eds.), Springer-Verlag, 1994.

[6] D. Bouchaffra. "Recovering from recogniser failure using Hidden Markov Mesh Fields: Extending linguistic post-processing", *in preparation*. 1995.

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

[8] F. Jelinek and R.L. Mercer. "Interpolated estimation of Markov source parameters from sparse data", in *Pattern Recognition in Practice*, pp. 381–397, E.S. Gelsema and L.N. Kanal (eds.), North-Holland, 1980.

[9] 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.

[10] R. Rosenfeld. "The CMU Statistical Language Modeling Toolkit". *Unpublished(?) manuscript.*, 1994.

[11] L.R. Bahl, F. Jelinek, and R.L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, **5**:2, pp. 179–190, 1983.

[12] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, **18**:4, pp. 467–479, 1992.

[13] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, **27**:379–423, 623–659, 1948.

[14] R. Rosenfeld. A hybrid approach to adaptive statistical language modeling. *Proceedings of the ARPA workshop on human language technology*, pp. 76–81, 1994.

List of Tables & Figures

Table 1. Performance table for the POS Model.
Table 2. Test sentences used in Table 1.

Figure 1. The `UTA` algorithm for Unique Tag Assignment.
Figure 2. The `UTA` algorithm at work.
Figure 3. What the colours mean.
Figure 4. Urns and marbles.
Figure 5. Comparing the performance of the two models.

S_i	$\#w$	$\#a$	$@a$	$\#w'$	HWR		Bigram		HWR+POS	
					$\#c$	$@c$	$\#c$	$@c$	$\#c$	$@c$
S_1	6	6	1.0	0	3	0.50	5	0.83	5	0.83
S_2	7	9	1.3	2	2	0.29	4	0.57	4	0.57
S_3	7	11	1.5	6	5	0.71	5	0.71	6	0.86
S_4	5	8	1.6	3	2	0.40	5	1.00	5	1.00
S_5	8	13	1.6	5	7	0.88	8	1.00	6	0.75
S_6	7	13	1.8	4	4	0.57	5	0.71	6	0.86
S_7	6	11	1.8	6	5	0.83	4	0.67	4	0.67
S_8	10	20	2.0	7	6	0.60	8	0.80	10	1.00
S_9	9	20	2.2	6	4	0.50	8	1.00	8	1.00
S_{10}	9	20	2.2	9	6	0.67	8	0.89	8	0.89
S_{11}	8	19	2.4	8	6	0.75	7	0.88	5	0.63
S_{12}	5	9	2.25	3	3	0.60	5	1.00	4	0.80
S_{13}	4	10	2.5	3	3	0.75	4	1.00	4	1.00
S_{14}	6	15	2.5	5	3	0.50	5	0.83	6	1.00
S_{15}	8	24	3.0	7	4	0.50	6	0.75	8	1.00

Table 1: **Performance table for the POS Model:** For each sentence S_i that we tested, we have computed the following statistics: $\#w$ = number of words; $\#a$ = number of total word senses; $@a = \#a/\#w$ = ratio of senses to words; and $\#w'$ = number of words with ambiguity (*ie.*, with more than one POS associated with them). The average number of candidates per word-position, which is also the average trellis-height, varied from 7.5 (for S_6) to 14.0 (for S_2). The sentences have been ordered by increasing ambiguity, as measured by $@a$. We then compare the number $\#c$ and the ratio $@c = \#c/\#w$ of correct top-choices in the trellis for the HWR with the corresponding figures after post-processing. We show the performance of the POS model relative to a word-bigram based re-ranking scheme.

1. we are expecting the closest inspection
2. in general they appreciated learning the business
3. except for feeling mad i recall nothing
4. i am testing the recognizer
5. be sure to have their favorite order ready
6. grant each extra request given by her
7. i guess the guide got lost
8. it came as a surprise that these have slow return
9. he may wonder why the wrong file got trashed
10. last time they were welcome to cut the deal
11. what kind of subject can split this group
12. time flies like an arrow
13. we still preferred spring
14. some are thinking power means right
15. this will test and close each clean set

Table 2: **Test sentences used in Table 1:** The sentences are ordered by increasing ambiguity as measured by the parameter $@a$. To conform to the recogniser's requirements, we have eliminated all punctuation, and have transcribed all upper-case letters to the lower case.

```
begin UTA
  foreach path W ∈ 𝓦 do
    Tag path W with a stochastic tagger
    # This assigns unique tags for this path
  od
  foreach index-pair i, j ∈ the trellis do
    if w_ij has_multiple_tags_assigned_to_it
      W_ij := arg max_{W ∈ 𝓟_ij} (tagger-score(W))
      t_ij := t ∈ 𝓣 s.t. (w_ij::t ∈ W_ij)
    fi
  od
end UTA
```

Figure 1: **The UTA algorithm for Unique Tag Assignment:** The tagger-score measures the reliability of the particular tag-sequence chosen by the tagger, and so provides a measure of confidence for the particular word::tag pair under consideration in the global context of the path for which this assignment was chosen by the tagger. The figure below illustrates the performance of the UTA algorithm.

The Truth:

Police	help	dog	bite	victim

The Recogniser's Choices:

Please	tulip	dog	bile	victim
Police	help	does	bite	system
Place	half	clog	lute	—

Tag Choices proposed by the tagger: (*In descending order of the tagger-score*)

Please::{UH,VB}	tulip::NN	dog::{NN,VB}	bile::NN	victim::NN
Police::{NN,VB}	help::{VB,NN}	does::{VBX,VB,NN}	bite::{JJ,VB,NN}	system::NN
Place::{NN,VB}	half::{JJ,NB}	clog::{VB,NN}	lute::NN	—

After Unique Tag Assignment:

Please::UH	tulip::NN	dog::NN	bile::NN	victim::NN
Police::NN	help::VB	does::VBX	bite::JJ	system::NN
Place::NN	half::JJ	clog::VB	lute::NN	—

Figure 2: **The UTA algorithm at work:** This example shows the trellis generated by the recogniser for a particular input sentence, and the state of the trellis after each of the `foreach()` loops in the UTA algorithm. The tag-set used here is: $\boldsymbol{\mathcal{T}}$ = {JJ – Adjective; NB – Number; NN – Noun; UH – Interjection; VB – Verb; VBX – Auxiliary verb}.

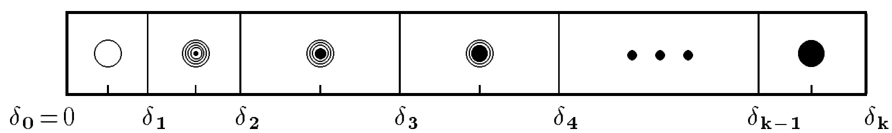

Figure 3: **What the colours mean:** The values of observations at a given state can range from $\delta_0 = 0$ to a maximum of δ_k. This range can be divided into intervals as shown above. Each interval is associated with a particular colour. The marbles in an urn represent possible observations from the state that the urn itself represents. The marbles are coloured according to the different intervals $(\delta_{j-1}, \delta_j]$ into which the observations fall.

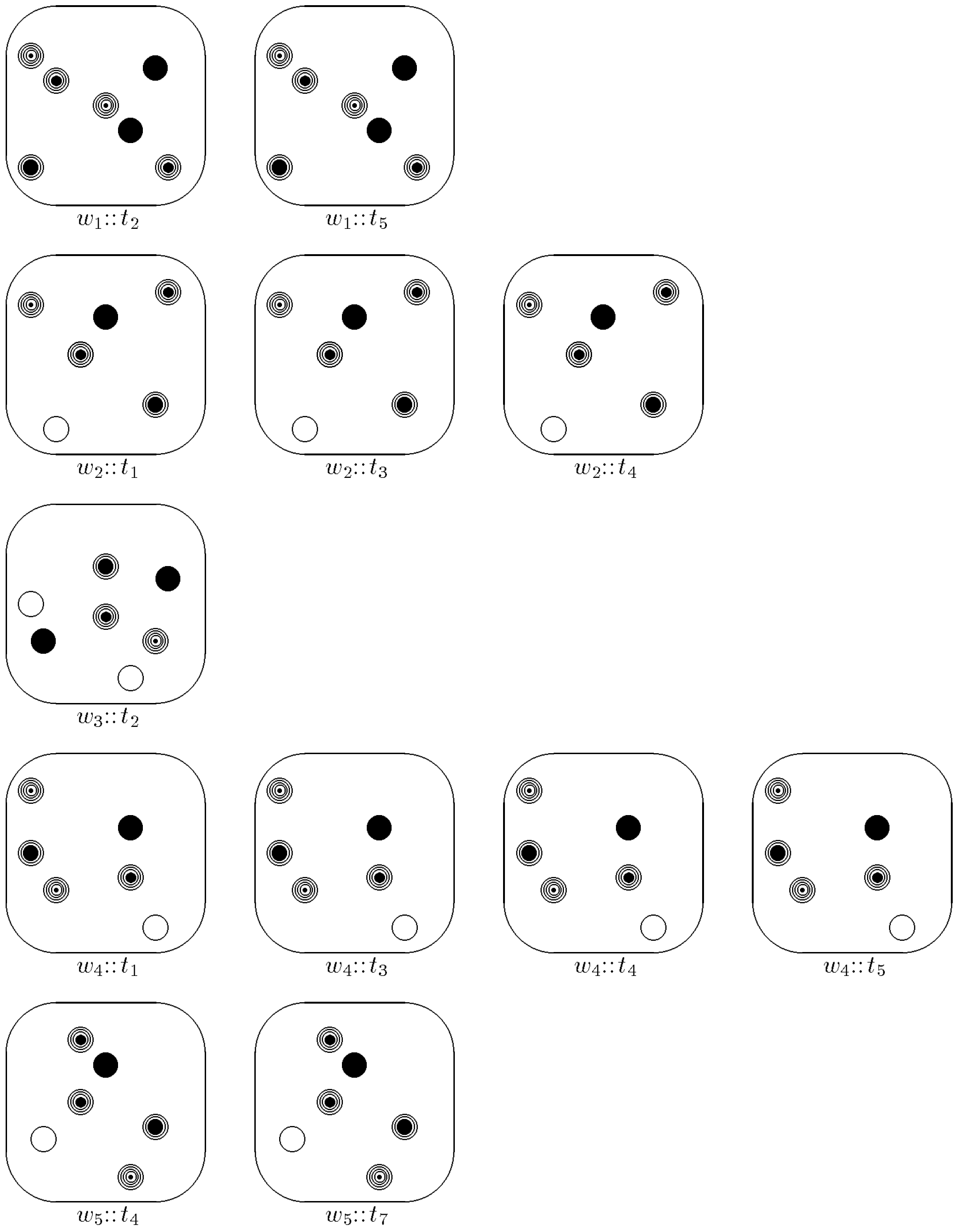

Figure 4: **Urns and marbles:** The urns represent states w::t, and the marbles represent possible observations at that state. Each colour corresponds to an interval of the range of the observation variable χ. The diffferent urns in any given row i correspond to the different parts-of-speech that are associated with the word w_i. Therefore the distribution of marbles is identical in all the urns of a particular row.

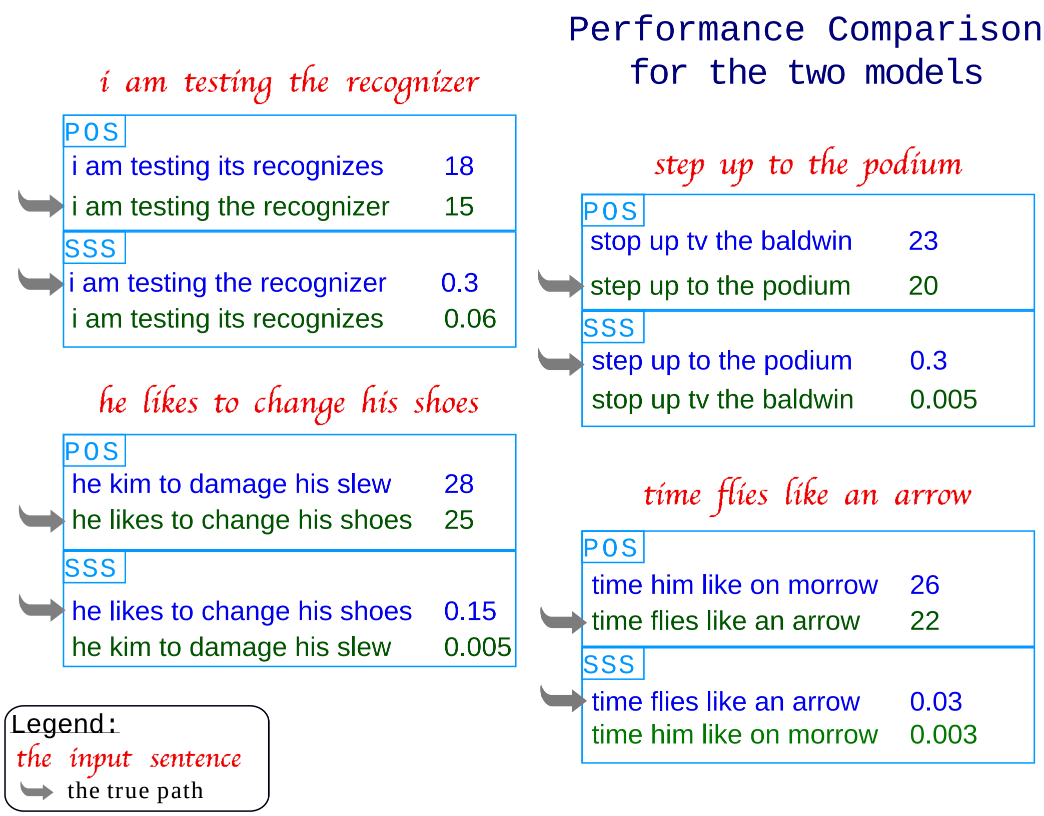

Figure 5: **Comparing the performance of the two models:** We show here the reranking performance of the POS Model and the SSS Model. Each box in this figure shows two *paths* in the input trellis, and shows the scores assigned by the two models to the two paths. The scores themselves are not comparable across the models, since the two approaches compute inherently incomparable quantities. But the truth path is chosen by the SSS Model over the alternate in every one of these examples.