Cursive Script Recognition using Wildcards and Multiple Experts

A. Hennig, N. Sherkat

Department of Computing, The Nottingham Trent University

Burton Street, Nottingham NG1 4BU, UK

Andreas.Hennig@mchp.siemens.de, ns@doc.ntu.ac.uk

Tel. +44-115-9486032; Fax: +44-115-9486518

Abstract

Variability in handwriting styles suggests that many letter recognition engines cannot correctly identify some hand-written letters of poor quality at reasonable computational cost. Methods that are capable of searching the resulting sparse graph of letter candidates are therefore required. The method presented here employs 'wildcards' to represent missing letter candidates. Multiple experts are used to represent different aspects of handwriting. Each expert evaluates closeness of match and indicates its confidence. Explanation experts determine the degree to which the word alternative under consideration explains extraneous letter candidates. Schemata for normalisation and combination of scores are investigated and their performance compared. Hill climbing yields near-optimal combination weights that outperform comparable methods on identical dynamic handwriting data.

Keywords

Poor quality handwriting recognition, dynamic handwriting data, multi-expert combination, wildcards, wildcard likelihood, explanation expert, self-confidence

1 Introduction

Recognition of unconstrained cursive handwriting has to deal with various problems on different levels. When it comes to the on-line recognition of English words, two major strategies can be observed in the literature: Holistic methods and Segment-and-Recognise methods [1].

Holistic methods extract a set of features that assumedly identify the written word as a whole. Features such as local extrema, loops or diacritical marks are chosen for ease of detection or for their stability over a variety of handwriting styles. In 'poorer' writing, however, (i.e. writing that is more difficult to recognise for both humans and machines) most features become less stable. Ascenders and descenders might become less prominent; i-dots might be misplaced, joined up or missing; or spurious loops might be introduced by overlapping letters. Comparison of the observed features with the entries of a database must therefore be tolerant. Tolerance, however, usually leads to recognition of a large number of additional words with similarly high scores. This reduces the number of possible words out of the full lexicon efficiently but often fails to identify the target word as top choice. In poor writing, the size of the reduced set of possible words can reach half the original vocabulary and more.

Segment-and-Recognise methods are based on the fact that words are constructed from single letters and the assumption that they can be recognised separately. The word image is therefore segmented into smaller parts of approximately letter size. Letter candidates are then recognised and represented as a letter graph, before a *word combination* procedure tries to construct words from the graph, see [2][3][4]. Combining the scores of the constituting letters can render these methods more discriminative than holistic approaches. However, as they usually require all letters of the target word to be present in the letter graph (i.e. the graph must be non-*sparse*), the target may be missed easily. To compensate, especially for poorer script, the aperture of the letter-recognition can be widened. Apart from the increased compu-

tational demands, this usually results in a substantial increase in additional but incorrect (i.e. *'surplus'*) letter candidates, rendering the letter graph more ambiguous. Even then, it is likely that some letters will still be missed due to the variability of handwriting styles.

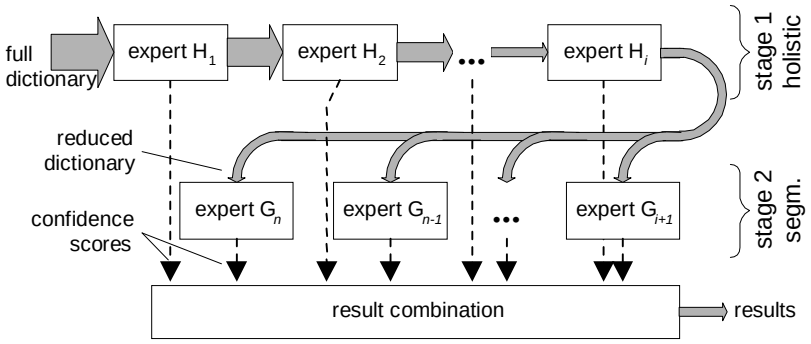

Fig. 1: Overview of the system architecture.

Accepting that the production of non-sparse letter graphs can be prohibitively expensive, the **w***ild***c***ard word* ***c****ombination* (WCC) Method focuses on a more flexible combination of letters into words. Similar to [11],[10] and [13], multiple experts are used and their results combined to increase overall performance. First, holistic methods are used to reduce the size of the dictionary under investigation in serial result combination. As a segmentational approach, a letter-graph is produced by the Hierarchical Fuzzy Inference (HFI) method [5], a forward-reasoning production rule system, which thereby also furnishes automatically obtained hypotheses of - usually overlapping - letter boundaries. The second stage of the WCC then evaluates validity and quality of the constructed words. Missing letter candidates in sparse graphs are represented by *wildcards* if and when they are required. Holistic experts are named $H_1 \ldots H_i$, segmentational experts $G_{i+1} \ldots G_n$. All experts then contribute to the parallel result combination, each representing a different aspect of a well-written word (Fig. 1) by a score $S_{i,h}$ (the score of the i^{th} expert for the word hypothesis h).

The handwriting data used for development and verification has been collected from 25 writers from various backgrounds, each donating 11 English sentences consisting of 150 words (119 thereof distinct) twice, for training and testing sets respectively. Thus, two times 3750 words have been collected on an NCR3125 pen-computer, showing a variety in styles and quality. The average quality of handwriting is rather 'poor', judging by both human judgement and the drop of an alternative method's performance (Multiple Interactive Segmentor or MIS [2], normally 70%) to below 40% top-choice recognition rate. The lexicon used to create word hypotheses consists of 4126 words, the most frequent words in the English language plus those used in the sentences. The system thus attempts to recognise poor-quality, writer-independent script based on a medium-size vocabulary. More details about data sets, lexicon and the system can be found in [6].

The following sections describe the principles of wildcard-based word recognition (where missing letter candidates are replaced by transient pace-holders or 'wildcards'), the holistic and segmentational experts employed, and the method of parallel combination experts, where the normalised results of the experts are combined with near-optimal weighting. Finally, experimental results are presented and final conclusions are drawn.

2 Using Wildcards

Fig. 2 depicts a schematic view of concepts involved. The target word 'am' has been written and is available as electronic signal. The example word is of average quality with respect to the entire set, despite the overall downwards slant of the text and the grossly oversized upwards stroke in ´a´. A letter recognizer produces a letter graph (the '*graph'*) of letter candidates ('*candidates'* for short). Letter candidates bear attributes provided by the HFI, notably the ink used, the rectangular bounding box encompassing the entire letter and a confidence

score. Word hypotheses (the '*hypotheses'*) are obtained from the reduced dictionary and evaluated with respect to the graph. Knowing the average widths of candidates, the expected positions of constituent letters can be established from hypothesis and image. We call a letter graph *sparse* with respect to a word hypothesis if one or more letters required for the hypothesis are missing from the graph. The graph in Fig. 2 for example is sparse with respect to the hypothesis 'am' as no letter candidate 'm' has been produced, but not with respect to 'an'. A *wildcard* representing 'm' is therefore created while considering 'am'. The location of the wildcard is defined by the two considered neighbouring letter candidates, and consists of ink that is not unused by them. The graph is thus been completed temporarily, the wildcard acting as a letter candidate.

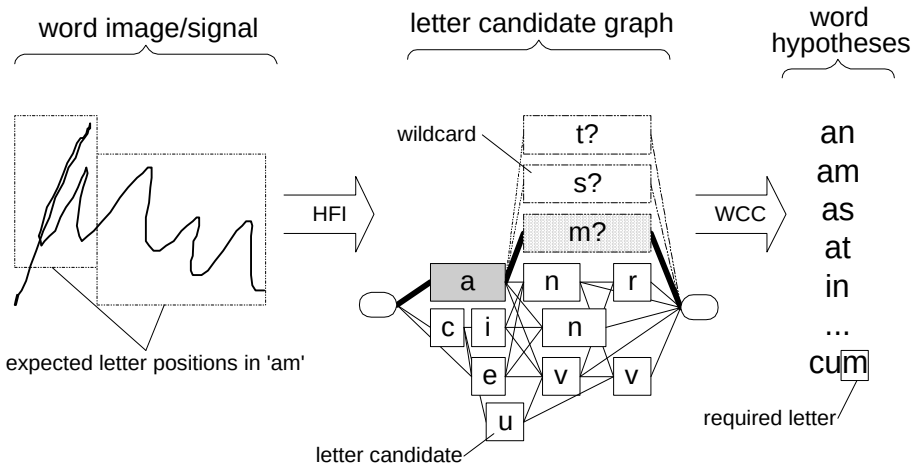

Fig. 2: A sparse letter graph for the target word 'am'.

If wildcards are produced without distinction, however, the number of constructible words increases dramatically, rendering the graph excessively ambiguous (e.g. allowing 'as' and even 'at' in Fig. 2). This corresponds to an expensive low-sparseness, high-surplus recognition strategy. To control wildcard creation, the following measurements have been defined. The likelihood $p_{wc,l}$ that a wildcard for a letter l is required is defined as

$$p_{wc,l} = \frac{No.\ target\ letters_l\ missed}{No.\ target\ letters_l\ written}\ . \qquad (1)$$

Where the number of missed letters indicates how many of the written letters in the raining set have not been produced by the HFI at their expected position. In our data-set, the values of $p_{wc,l}$ have been observed to vary widely from $p_{wc,'c'}$=5% for the letter 'c' up to $p_{wc,'b'}$=63%. Using a wildcard for 'c' should thus be penalised more strongly than a 'b'-wildcard, i.e. by a penalty $p_{pen,l}$=1-$p_{wc,l}$.

The more surplus candidates are produced for a letter *l*, the less important is their presence in the graph. The reliability $p_{rel,l}$ is therefore based on the average 'correctness' of letter candidates produced:

$$p_{rel,l} = \frac{No.\ letter\ candidates_l\ correct}{No.\ letters\ candidates_l\ produced}\ . \qquad (2)$$

The above measures are based on the known overall performance of the letter recognizer on the entire training set, i.e. without attempting to complete graphs with wildcards. They can therefore only be used to penalise wildcards based on the general difficulty they pose for the recognizer (exploited in the 'sparseness' expert below) but not on their congruency with the word encountered. The use of a wildcard 't' in Fig. 2 for example, should be strongly penalised independent of $p_{rel,'t'}$ or $p_{wc,'t'}$ as the word does not contain a corresponding ascender or t-bar. This aspect is addressed in the 'explanation' expert described later.

3 Holistic dictionary reduction

The first stage uses physical features and supplementary 'sparseness' features to reduce the active dictionary. Physical features include word length, diacritical marks, ascenders, descenders, combined as/descenders, and segments crossing the word's axis. Approximating

spline functions [7] are used to represent the boundaries of handwriting zones (i.e. the zones containing ascenders, descenders and middle zone letters) for feature extraction and to define the word's axis as the centre of the middle zone. It is vital that the error introduced by each holistic expert is small, as their serial combination accumulates errors. An error of 0.5% per expert was deemed acceptable and achieved by choosing appropriate tolerances.

Although measuring the **word length in letters** is a highly discriminative and an intuitive feature, its correct extraction from cursive script is difficult as letters vary greatly in complexity and length. As letter length cannot be observed directly this is not a physical feature in the strict sense. [8] and [2] both estimate the word length from the number of a word's axis crossings divided by the average number of crossings per letter. On our data set, this estimate was observed to err by up to 6.5 letters, 0.9 letters on average. In order to achieve a - relatively high - error rate of 1%, a tolerance of ±3.8 letters had to be endorsed. A high tolerance, however, results in a poorer dictionary reduction (quoted as the number of words that pass the filter, averaged over the entire set) in our case a reduction to only 71% (i.e. only 29% of hypotheses were eliminated due to incompatible length). Alternatively, the word length in letters can be estimated from the letter candidates produced by the HFI and the known distributions of the width of those candidates. Based on the accumulated distributions, the expected word length in letters is described by a normal distribution (i.e. μ_{len} and σ_{len}). The length expert H_{len} accepts hypotheses if

$$H_{len}: \mathit{length} \in \left[\left\lfloor \mu_{len} - t_{len} \max(1, \sigma_{len}) \right\rfloor, \left\lceil \mu_{len} + t_{len} \max(1, \sigma_{len}) \right\rceil\right], \qquad (3)$$

where the tolerance t_{len} has been experimentally determined as 1.0 to yields the desired error rate of 0.5%, which improved dictionary reduction to 56% on identical data.

Rather than defining the word length via the logical feature ‘letter’, the number of **axis crossings** can be used directly. For each character in various writing styles, the numbers of possible crossings are denoted by an interval [$mn_{cross,l}$, $mx_{cross,l}$] (i.e. the minimum/maximum of permissible crossings inside the letter *l*). The hypothesis is accepted if its accumulated intervals contain the number of observed crossings:

$$H_{cross}: \; crossings \in \left[-t_{cross} + \sum_{letters\ l} mn_{cross,l} \; , \; t_{cross} + \sum_{letters\ l} mx_{cross,l} \right] . \qquad (4)$$

An additional tolerance t_{cross} is employed to limit the reduction error. With $t_{cross}=3$, the dictionary was reduced to 70%.

Similarly, the number of occurrences of other physical features is evaluated, i.e. the number of **diacritical marks**, **ascenders**, **descenders** and combined as/descenders as in ‘f’. Tolerances *t* are defined individually for each feature, intervals individually for each feature and letter. Except for diacritical marks, features are extracted as certain and potential variations. The top of strokes are classified using fuzzy logic as top-zone points or half-zone points. If the fuzzy classification as a top-zone point is ambiguous, the stroke is marked as a potential ascender. If the classification is certain, the stroke is marked as both a certain and a potential ascender. Likewise, certain descenders include potential descenders, and combined as/descenders include both ascenders and descenders. This facilitates the matching process for script of less clear zoning. In discrete script, combined as/descenders only occur in letters ‘f’. In cursive script, however, the ligature connecting letters ending in a descender and letters starting with an ascender can form an additional full stroke, e.g. ‘g’ and ‘h’ as in  . The maximum permissible number of full strokes is therefore increased if necessary, the minimum required number of ascenders and descenders is reduced. It has been observed that the majority of letters with unexpected zoning occur either at the beginning of a word (par-

ticularly letters ‘c’, ’s’, ’r’, ’j’ and ‘p’) or at its end (letters ‘r’ and ‘s’). The ranges for hypotheses containing these cases are adjusted accordingly. The resulting in dictionary reduction ranged from 53% to 99%.

The overall shape of the word with respect to ascenders and descenders is described by the word’s **zoning**. The word image is divided horizontally into *n* blocks, each block being marked whether or not it contains certain or potential ascenders or descenders. The image (‘question’) for example is represented by ‘□□□□‘ for *n*=4. A similar sequence is obtained for each hypothesis, allowing for style and kerning variations. Hypotheses are only accepted if all blocks correspond.

The **sparseness** of a letter graph is expressed as the estimated number of required wildcards. A required letter is deemed present in the graph if a matching letter candidate overlaps the expected position. If no matching candidates are found, a wildcard is required, i.e. the graph is sparse. This heuristic disregards constraints of mutually exclusive candidates. They are enforced in the second stage. Hypotheses are accepted if more than half of the required letters seem to be present. Very short words are always permitted, as the other holistic features are usually sufficiently discriminative. The maximum number of permissible wildcards has been experimentally restricted to 7 in order to restrict the combinatorial explosion of possible words and thus computational demands:

$$H_{sparse}: \; No.\,wildcards \le \min\left(\lceil \tfrac{1}{2}\, wordlength \rceil, 7\right) \text{ or } \; wordlength \le 2 \qquad (5)$$

This condition, however, is irrespective of the wildcards’ likelihoods. The **weighted sparseness** expert therefore evaluates by how much the accumulated reliability of used letter candidates is surpassed by the accumulated penalty for used wildcards.

$$H_{wsparse}: \sum_{letters\ l\ in\ graph} p_{rel,l} - \sum_{wildcards\ l} p_{pen,l} \ge t_{wsparse} \text{ or } \; wordlength \le 2 \qquad (6)$$

The tolerance $t_{wsparse}$ has been experimentally chosen as -0.12, i.e. word hypotheses are accepted if wildcards not more than slightly outweigh letter candidates. If both p_{rel} and p_{pen} are neglected, i.e. $p_{rel,l} = p_{pen,l} = 1$, this is similar to the 50% rule of the unweighted sparseness expert.

4 Combining letters from sparse graphs

The holistic examination of the letter graph ignores whether or not candidates can actually be combined into a word. The second pass therefore enforces constraints such as mutually exclusive usage of ink and maximum ligature length and evaluates the correlation of constituting letters' heights or widths. Valid combination of candidates and wildcards are searched using a recursive graph traversal. The search is pruned by enforcing the two sparseness criteria throughout. Several experts assess the quality of a valid combination, each furnishing a fuzzy score $S_{i,h} \in [0,1]$. Subsequently, the experts are described in the context of a given word hypothesis w. $S_{i,h}$ is therefore abbreviated to S_i or even to S, where the context provides the expert's i.

A single value S_i, however, is unable to model the appropriateness of an expert. An expert using relative sizes of subsequent letters, for example, is inapplicable if the word consists of a single letter. A single response S_i, however, could only indicate that the word is a good match (if $S_i=1$ is returned) or a bad match (if $S_i=0$) or any value in-between. Any score would, however, affect a combined score of all experts without grounds, as the expert should have been excluded entirely. Similarly, the expert would be more applicable to longer words as more letter pairs can be evaluated, which does not imply that the match is better. An independent second response $C_{i,h}$ is therefore returned by each expert describing the expert's own opinion of its appropriateness (or its *self-confidence*). The self-confidence can thus represent

a binary decision of applicability ($C_i \in \{0,1\}$) or a degree of applicability ($C_{i,h} \in \Re_+^0$, ideally by $C_i \in [0,1]$, although a scaling factor w_i will later obtained automatically and used in Eq. 14 to scale C_i appropriately)

In order to utilise the criteria evaluated by the holistic experts, the difference of the underlying measures (e.g. number of crossings in the script and the expected average number of crossings) are employed as scores S_i. The self-confidence values are defined by the size of the respective acceptancy intervals (e.g. the allowed variation in numbers of crossings)

4.1 Spatial correlation experts

The following experts assess the spatial arrangement of letter candidates according to various aspects. The **horizontal overlap** between subsequent candidates is the 'overlap' of their bounding boxes. If the boxes share a border the overlap is zero, intersecting boxes have a positive overlap, i.e. the width of intersection. Isolated boxes have a negative overlap, i.e. the distance between them. The overlap is normalised by the averaged width of the boxes involved. The score is derived from the observed overall distribution of normalised overlaps (μ=-0.11, σ=0.50, i.e. an 11%-wide gap). The self-confidence is the number of observations, i.e. $C_{overlap}$ =*No.Letters*-1. A **vertical overlap** expert using the same principle is also employed, based on an average vertical overlap of 36% (σ=0.09).

The **horizontal distance** between two centres of gravity is similar but ignores the width of letters, which can vary widely. A slant corrected projection of the centroids to their average height yields the distance, which is on average 92% the averaged box width (μ=0.92,σ=0.48).

The **letter ratio experts** exploit the extents of used candidates in relation to word image and hypothesis. Extents used are width, height, aspect ratio, area and amount of ink both in original ink-points and in number of segments. The width-ratio expert $G_{width\text{-}ratio}$, for example,

uses the candidates' actual widths w_l and their known average width-factors μ_l to obtain the width $w_{\mathcal{L}}$ of an imagined average letter candidate $\mathcal{L}$ in a word of width w_{word}:

$$w_{\mathcal{L}} = w_{word} \Big/ \sum_{letters\ l} \mu_l \ . \tag{7}$$

As the definition of expected width-factors μ_l thus recursively depends on $w_{\mathcal{L}}$, their values have been derived by successive approximated, based on correctly recognised letter candidates in the training set of 3750 words as in [6]. They describe the average width of candidates l relative to $\mathcal{L}$, approximated by the Gaussian distributions $\varphi_l(\mu_l, \sigma_l)$. The highest observed value $\mu_{'m'}$=1.49 therefore indicates that 'm' is on average 1.49 times wider than $\mathcal{L}$ (Fig. 3). The letter closest to $\mathcal{L}$ in term of width is 'v' (1.007), the narrowest is 'c' (0.71).

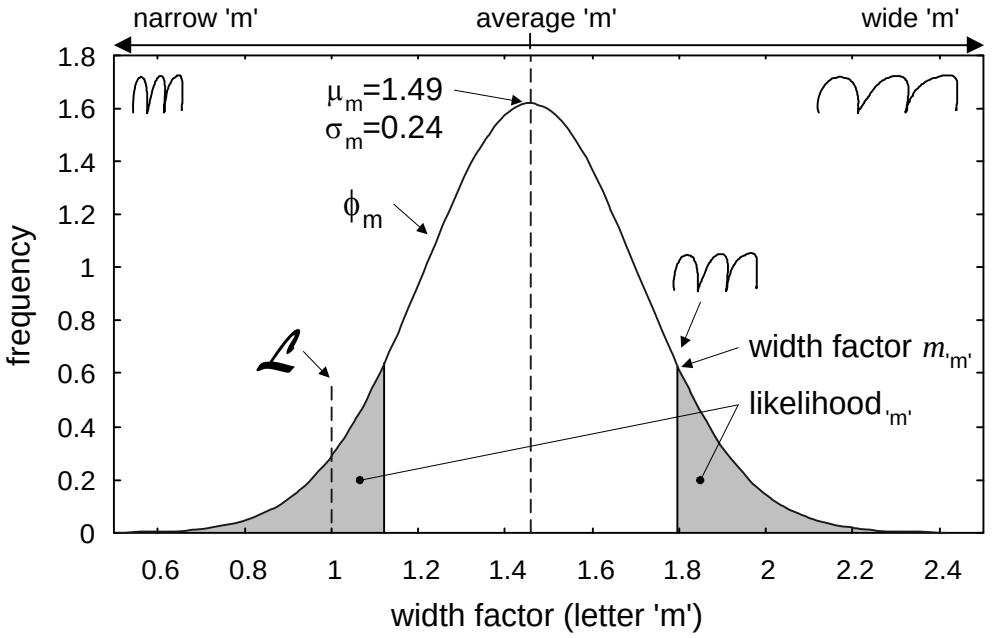

Fig. 3: Obtaining the width-likelihood of a letter 'm'.

For each letter, the observed width factors m_l are then used to define the likelihood measure and the expert's score. Applying 'better-than' fuzzy normalisation as suggested in [9], the

likelihood is defined as the expected number of candidates that are worse than the current one:

$$\begin{aligned} & likelihood_l = 2\Phi_l\left(\mu_l - \left|\mu_l - m_l\right|\right) \quad ; \quad m_l = {w_l}\big/{w_{\ell}} \\ G_{width-ratio}: \quad & S = {1}\big/{No.Letters} \cdot \sum_{letters\ l} likelihood_l \quad ; \quad C = No.Letters \end{aligned} \quad . \qquad (8)$$

The **letter correlation experts** assess the relation between the extents of subsequent candidates, e.g. widths or heights of ⋒ and ℓ in 'me'. The height ratio of letter candidates 'm' and 'l' should thus deviate further from their expected height ratio (~1.6) than candidates 'm' and 'e' (~1.0) . Different from the ratio experts, extents are evaluated with respect to immediate neighbours rather than the entire word, which should increase accuracy. Obtaining statistics of all possible combinations, however, would require a vast training set. Therefore, only combinations that occurred with a large enough frequency ($n_{k,l} \geq 20$, which are only 150 combinations in our training set) are employed - the expert becomes inappropriate otherwise. Extents used are width, height, used ink, area, overlap and the centroid distance. The score is again derived from 'better-than' fuzzification; the confidence depends on the number of examples encountered:

$$\begin{aligned} & likelihood_{k,l} = 2\Phi_l\left(\mu_l - \left|\mu_l - m_l\right|\right) \quad ; \quad m_{k,l} = {w_k}\big/{w_l} \text{ ; if } n_{k,l} \geq 20 \\ G_{width-correl}: \quad & S = {1}\big/{No.Comb} \sum_{combinations\ k,l} likelihood_{k,l} \quad ; \quad C = \sum_{combinations\ k,l} \min\left(0.01\, n_{k,l}, 1\right) \end{aligned} \quad . \qquad (9)$$

As the above experts all use similar features, they are expected to yield similar scores, i.e. to be strongly correlated. While the group of highly specialised experts can render a more precise verdict, their applicability is reduced to fewer well-known cases.

4.2 Letter candidate experts

The ink-usage provided as attribute of letter candidates denotes how important a segment of ink is to a candidate and if it can be shared between different candidates, e.g. a common t-bar. The **ink-usage experts** enforce mutual exclusion as described in [5] and evaluate how much of the word image is used by the candidates. Every ink-segment ps_s can be used in one of four modes per candidate: core, shared, latent and unused. Each mode has an importance factor assigned to it (e.g. $imp(ps_s)$ = 1.0, 0.8, 0.5 and -0.05 res.). Using a negative factor for unused segments penalises ligatures. The segment-oriented expert does not discriminate between segments of different lengths, i.e. a long ligature would be penalised identically to a small hook. The ink-oriented version therefore employs the number of ink-points per segment, i.e. its duration. Although they depend on segmentation accuracy and writing speed respectively, it is assumed these influences remain consistent enough within a single word to obtain a meaningful comparison.

$$\begin{aligned} G_{ink-segments}: \quad & S = {}^{1}\!/_{No.Segm} \sum_{segments\,s} imp(ps_s) && ; \quad C = 0.05\,No.Segm \\ G_{ink-points}: \quad & S = {}^{1}\!/_{\sum inkPoints_s} \sum_{segments\,s} imp(ps_s)\; inkPoints_s && ; \quad C = 0.05\,No.Segm \end{aligned} \tag{10}$$

The degree of certainty to which a letter candidate has been recognised is provided as a fuzzy membership degree of the respective character class. The degrees of all letter candidates used in the word hypothesis are averaged to obtain the score of the **membership expert.** The confidence C_i is the number of candidates, both without counting wildcards. The expert also ignores the many surplus letters, i.e. it 'turns a blind eye' to contradicting evidence. If, for example, the target word contains 'h', the graph will most probably contain both 'l' and 'h' candidates at similar high degrees of membership, possibly resulting in confusion of words

like ‘hook’ and ‘look’. This corresponds to *forward reasoning*, asking how good the hypothesis is ‘explained’ by *observations*, i.e. by selected letter candidates (Fig. 4).

4.3 Explanation experts

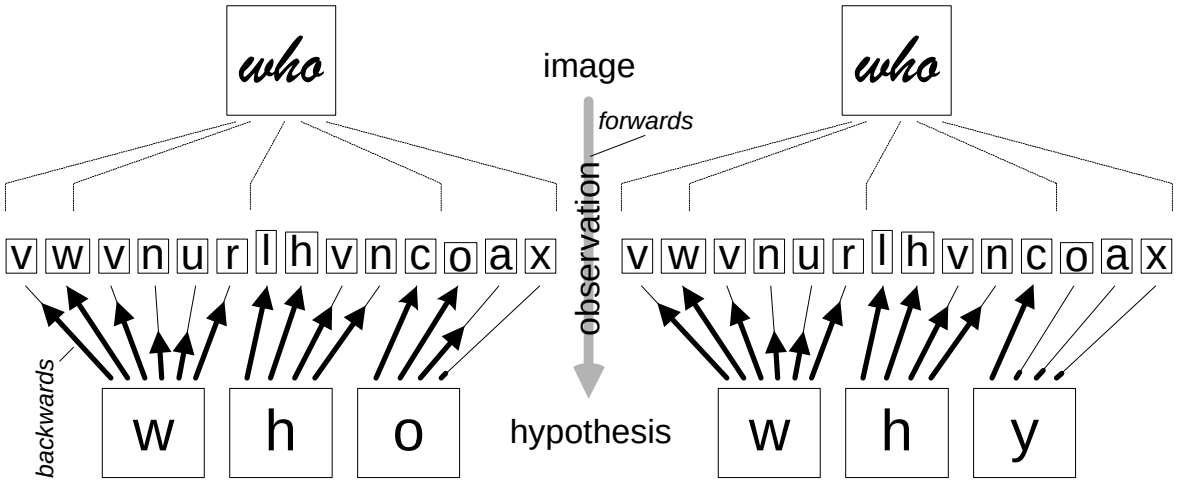

Fig. 4: Forward and backward explanation modes.

The **explanation-experts** therefore evaluate contradicting candidates using *backward reasoning*, asking how well the entire graph is explained by the word hypothesis (Fig. 4). As a basic rule, candidates used for the hypotheses support the hypotheses while the remaining candidates contradict it. Unused candidates representing the same letter at approximately the same position as a used one are the result of oversegmentation and multiple pattern templates per letter and are therefore not contradicting.

In the hook/look example, the ‘h’ explains the ‘l’ completely, as ‘l’ is a sub-shape of most ‘h’, while the inverse is not true. This differs from a cell in a confusion matrix, which would represent a statement like “an ‘h’ produces an erroneous ‘l’ in *X*% of the cases observed”. Explanation, in contrast, reasons that “if the word hypotheses contains an ‘h’, a corresponding surplus candidate ‘l’ is completely irrelevant”. The first is a statement of frequency (*X*%), the second a statement of degree (“completely”). In the example in Fig. 4, longer arrows repre-

sent a higher degree of backward explanation. The hypothesis ‘who’ is thus more likely than ‘why’ as the candidates ‘o’ and ‘a’ are more easily explained by ‘o’ than by ‘y’.

The explanation expert is implemented using a number of rules $expRule_r$, each explaining an observation o from sub-hypotheses h to a degree ξ. Sub-hypotheses are letters or groups of letters in the word hypotheses, observations are the letter candidates in the graph.

$$\xi_{r,h,o} = expRule_r\left(sub-hypothesis_h, observation_o\right) \quad ; \quad \xi_{r,h,o} \in [0,1] \tag{11}$$

Applying all applicable rules yields the degree of the best available explanation for each observation:

$$\xi_o = \max_{rules\, r} \quad \xi_{r,h,o} \ . \tag{12}$$

The overall explanation score is derived from the ratio of unexplained and total observations, weighted by the observation reliability.

$$G_{explain}: \quad S = \frac{1}{C} \sum_{observations\, o} (1-\xi_o) p_{rel,o} \quad ; \quad C = \sum_{observations\, o} p_{rel,o} \tag{13}$$

The granularity of both observation and hypothesis can vary in order to allow different modes of explanation (Fig. 5a): 1:1–Transposition (e.g. a used 'l' explains a surplus 'e' and vice versa as both look alike if written as ℓ, i.e. $\xi_{1,'l','e'}$=1 and $\xi_{2,'e','l'}$=1), 1:N–Splitting (e.g. a 'g' explains both a 'c' and a 'j', $\xi_{3,'g','cj'}$=1) and its inverse, M:1–Merging (e.g. a used 'c' and 'l' explain a surplus 'd', $\xi_{4,'cl','d'}$=1). For reasons of computational complexity, the M:N explanation (e.g. 'nun' explains 'mm') has not been implemented.

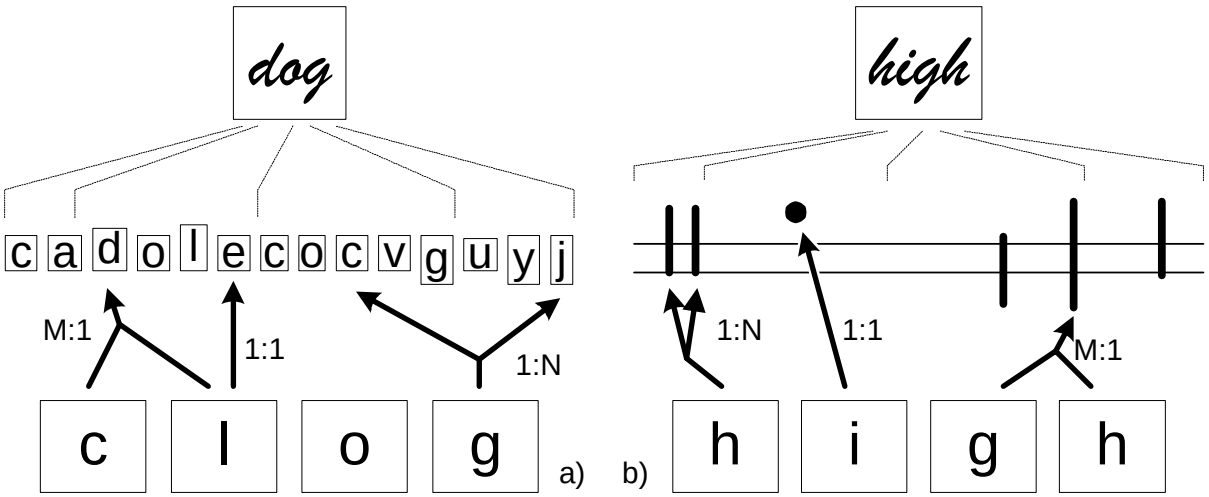

Fig. 5: Explanation modes: 1:1, 1:N, M:1, using a) letter candidates and b) shape features as observation.

The **shape explanation expert** differs from the candidate explanation expert only in using shape features as observations, i.e. ascenders, descenders and diacritical marks. The combination of 'g' and 'h' explains the observation of the descender, the combined as/descender and the ascender by applying rules $\xi_{5,'g',des}=1$, $\xi_{6,'gh',as/des}=1$, and $\xi_{7,'h',as}=1$.

The **wildcard explanation experts** use the same set of rules in inverse direction of reasoning. If, for example, a wildcard 'h' is required for a word hypothesis 'who', the presence of letter candidates 'l' and 'k' suggest that the image might resemble an 'h' even though the letter recognizer failed to produce the candidate (Fig. 6a). Similarly, the presence of descenders explains the need of a 'y'-wildcard well in the shape wildcard explanation expert (Fig. 6b).

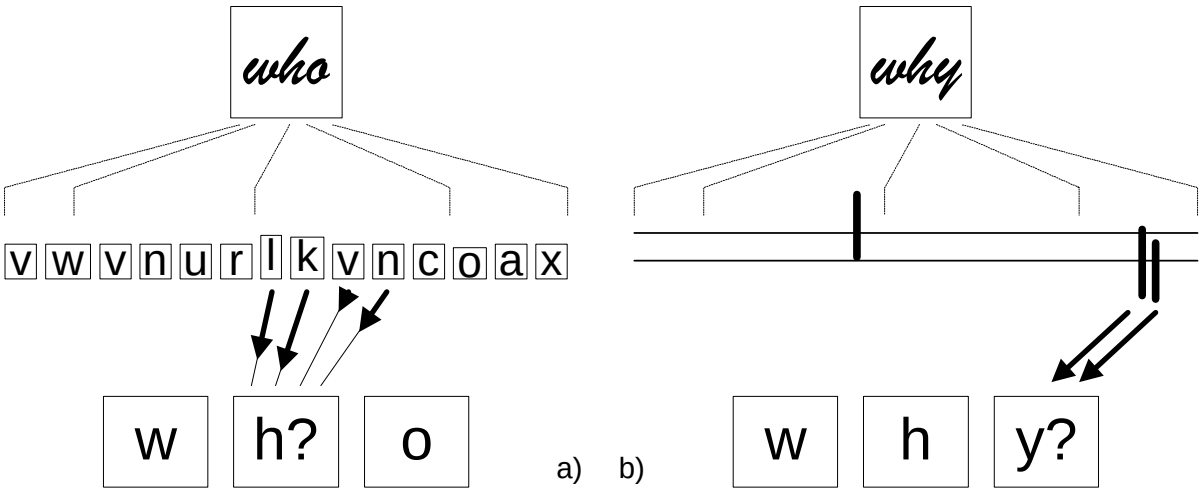

Fig. 6: Wildcards are explained by shape features or letter candidates (forward explanation).

Further to the rules described, 50 more rules have been defined manually, which model the similarities of other common sequences of letter candidates. Many of them use regular expressions as hypothesis or observation, e.g. the rule that any of 'acdgoq' explains 'c' fully is $\xi_{8,[acdgoq],'c'}=1$.

5 Optimised classifier combination

Above considerations result in a total of $n=40$ experts, 6 of which are used first in the holistic and than in the segmentational stage. As many of them exploit similar measures, it is hardly surprising that some are highly correlated (up to a Pearson coefficient of 0.79 for the horizontal-overlap expert and the letter correlation experts using horizontal overlap). The average absolute correlation of scores, however, is unexpectedly small (0.09).

5.1 Normalisation methods

Various implicit assumptions have been made about the scores S_i: a) range, i.e. $S_i \in [0,1]$; b) monotonicity, i.e. a higher score means a better match; c) linearity, i.e. a double score means twice as good a match. Apart from the range, none can be determined directly. For

illustration, frequency histograms of all word hypotheses (light grey area, vertical axis “frequency”) and of correct ones only (dark grey) are shown in Fig. 7 against the scores furnished by one exemplary expert (the membership expert) on the horizontal

Assuming that the individual experts yield consistent scores for the target words, however, their cumulative histogram can be used to normalise the scores with respect to b) and c). The cumulative histogram is also shown in Fig. 7 which corresponds to the **better-than normalisation** (denoted by $\tilde{S}_i^{bt}$, i.e. the normalised score $\tilde{S}$ of expert *i*, using better-than *bt*) normalising S_i (horizontal) into $\tilde{S}_i^{bt}$ (vertical axis “normalised S_i“).

With $\tilde{S}_i^{bt}$, a score of zero is assigned to the worst target word match, one is assigned to the best. Wrong hypotheses, however, might be rated worse than the worst target word, but would also be normalised to zero. To maintain the distinction, the **modified better-than normalisation** function for $\tilde{S}_i^{bt+}$ has been modified to start from an intuitively chosen 10%. This violates the linearity assumption, but only for very poor hypotheses.

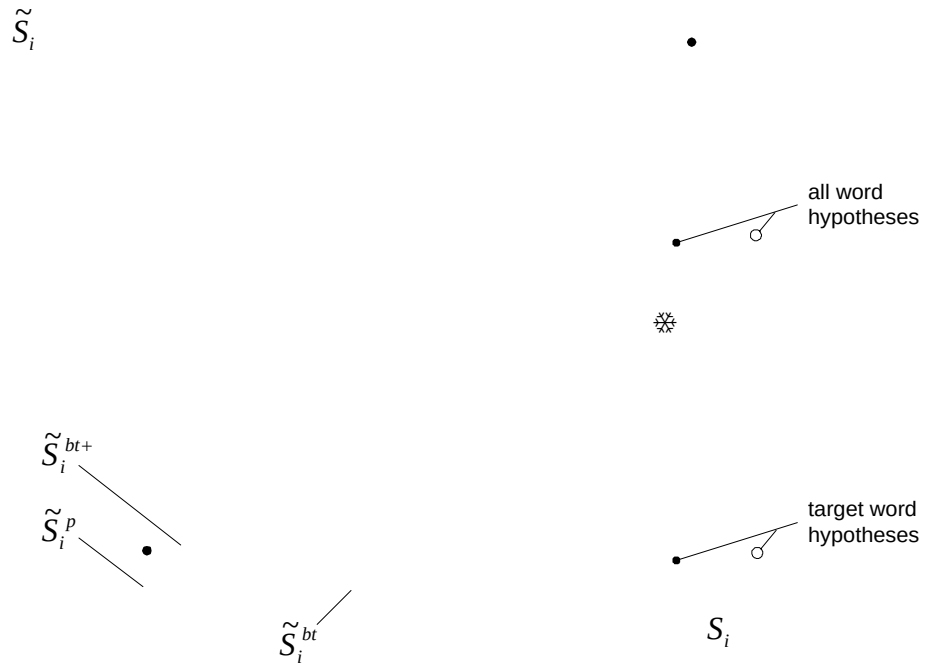

Fig. 7: Better-than and probabilistic normalisation functions based on hypothesis histograms.

Furthermore, **probabilistic normalisation** as employed in [10] has been investigated, replacing scores with the conditional probability $\tilde{S}_i^{p} = p(\text{word is target}, S_i) / p(S_i)$. Determining an apt bin size to extract $p(S_i)$ from the histogram, however, is subjective and can influence the normalisation function substantially. It also discards the monotonicity characteristics that have been carefully designed into the individual experts and thus depends on the data set being highly representative. In some cases, correct word alternatives with higher scores S_i can thus be rated lower solely because they had to compete with more invalid hypotheses in the training set (e.g. Fig. 7, near mark ❄).

5.2 Combination schemata

Investigations into *classifier combination* (e.g. [11]) recommend a sum or median rather than a product combination strategy, as it restricts the effect of occasional complete failures of single experts. The experts' scores are therefore combined using the average of all *n* experts

to produce the final score S_h of a word-hypothesis h. As the experts differ in recognition performance, the average is weighted by weights w_i (which also adjusts the scale of C_i if necessary):

$$S_h = \frac{1}{\sum w_i C_{i,h}} \sum_{i=1}^{n} w_i C_{i,h} \; f_{...}(S_{i,h}) \; . \tag{14}$$

The modifier functions $f_{...}$ allow to model different combination methods: a) sum combination: $f_{sum}(S_i) = S_i$ and b) product combination: $f_{prod}(S_i) = \ln \max(10^{-50}, S_i)$. The total number of outvoted alternatives is used in the c) Borda count [12]: $f_{borda}(S_i) = \|S_{worst}\| - \|S_i\|$, expressed as the distance in ranking (denoted by $\|..\|$) from the worst alternative. The Borda count, however, requires identical numbers of alternatives per expert, i.e. it would not allow an expert to declare itself inappropriate. The d) reverse Borda count is therefore based on the ranking distance from the best alternative: $f_{rev\_borda}(S_i) = \|S_{best}\| - \|S_i\|$. Using ranking discards the actual value of S_i, i.e. it ignores if two subsequently ranked alternatives have almost identical scores or not. Borda count variations based on the measurement thus employ the distance from the score of the best/worst alternatives: e) $f_{mb}(S_i) = S_i - S_{worst}$ and f) $f_{rev\_mb}(S_i) = S_i - S_{best}$. Rather than combining the scores, the e) '*oracle*' (based on [13] where it only applies to experts on abstract level, extended to include rank-level experts) selects the expert that provides the best ranking for the correct hypothesis of the current word (tied ranks are broken randomly). Although theoretical, the oracle yields the upper limit for dynamic expert selection strategies. These combination methods have been chosen for their simplicity and widespread use respectively theoretical interest (oracle). Normalised scores $\tilde{S}_i$ have been investigated for all combination methods except the Borda counts, as ranking is mostly unaffected by normalisation.

The weighting factor w_i serves two purposes: it models the importance of an expert and it scales the expert's confidence values C_i, i.e. it will dampen overly self-confident experts. Obtaining optimal w_i, however, is not trivial.

5.3 Combination optimisation

The problem of obtaining optimal weights w_i translates into finding the maximum of the function $f_{rtop}(w_1,w_2,...w_n)$ which yields the overall top-choice recognition rate for the entire data set. Evaluation f_{rtop} using a full run of the recognizer, however, requires approximately 20hrs (on a Sun UltraSPARC). The values S_i and C_i have therefore been obtained for each word hypothesis considered and stored on file in reduced precision. Evaluation time has thus been reduced to approximately 90 seconds. Even then, the cost of evaluating all possible combinations (e.g. $w_i \in [0,1]$ in – relatively coarse - steps of 0.01 for all n experts) is prohibitive. Instead of searching the n-dimensional search space exhaustively, a simple hill climbing method based on random search has been employed (ref. [6] for details). As the solution is not necessarily the global optimum, the process has been repeated from various starting positions. The different near-optimal solutions obtained differed in their w_i but yielded almost identical recognition rates. This seems to suggest that a) the problem is well posed, i.e. the chance of obtaining substantially inferior solutions is reasonably small, and that b) some experts can be used interchangeably, which is a consequence of them being correlated. Even if it cannot be guaranteed that the obtained solutions are acceptably close to the overall optimum, the gain of approx. 18% over a 'naïve' combination with all weights set to one is very satisfactory.

6 Experimental Results

In the training set, the average letter recognition rate of the MIS [2] was observed to be 89% (at any rank). In order to achieve this high rate, the surplus factor (i.e. the ratio of incorrectly

produced letter candidates to written letters) was allowed to grow as big as 16.7 on average, reaching 68.6 for 'j', which renders the presence of a 'j' in the letter graph virtually meaningless. As the average word length in the data set is 5.7 letters, one would expect to be able to construct only around 53% of all target words, in relation to which the top-choice word recognition rate of 38.5% is rather high.

The HFI, on the other hand, had a poorer letter recognition rate of 69.8/66.2% (values from training/testing sets respectively) at a substantially lower surplus factor (5.2/5.5 on average, up to 21.1/21.3 for 'v'). The expected number of fully constructible words is thus only 12.8/9.5%. The performance of individual experts ranged from 1.8/1.6% to 27.5/24.5% top choice recognition rate, 6.9/6.4% on average. The three most effective experts were holistic zoning (27.5/24.5%), letter-ratio using width (21.5/20.2.4%) and ink-usage (15.0/15.0%). Ties in the top-choice were broken randomly, with 8.0/7.9 top alternatives on average. These rates, however, encompass all target words, and therefore also include cases where experts declared themselves inappropriate. Excluding those, the rates increased up to 31.7/29.2%, 8.1/7.8% on average, which demonstrates the influence and importance of self-confidence.

Pass one reduced the 4k lexicon to an average of 110/103 words (i.e. by 97.3/97.5%). It failed to include the target word in the reduced lexicon in 13.1/16% of the cases. Incompatible physical features caused 1.3/2% thereof, the remaining 11.6/14% were due to excessive wildcard requirements. Top-choice ties have again been broken randomly, which happened only in 1.5% of all words. Table 1 shows the combined and optimised top-choice recognition rates obtained. Their comparison, however, assumes that the obtained near-optimal solutions are similarly close to the global optimum.

The top-choice performance generally declined by about 4% in Table 1 from training to testing set. This is mainly caused by the fact that the training set appears to be of better quality

than the testing set, as it had been obtained first while the subjects wrote more carefully. The decrease is thus more due to the 2.9% increase in pass 1-rejections together with the 3% decrease in the best expert's recognition rate rather than overlearning effects during weight optimisation (which only affects *2*n=80* parameters).

Table 1: Optimised recognition rates for different normalisation and combination methods

normalisation	$\|S_i\|$	S_i	$\tilde{S}_i^{bt+}$	$\tilde{S}_i^{bt+}, \tilde{C}_i^{bt+}$	$\tilde{S}_i^{p}$
f_{sum}	-	52.7/49.4%	54.1/50.1%	55.7/**51.6**%	51.6/46.4%
f_{prod}	-	40.5/37.3%	55.2/50.9%	55.4/**51.6**%	34.3/31.1%
f_{borda}	53.7/48.2%	53.6/49.8%	54.5/51.0%	51.9/48.5%	48.9/43.4%
f_{rev_borda}	53.5/48.0%	53.2/48.8%	55.4/50.6%	54.4/50.6%	50.7/46.1%
oracle		22.9/21.8%	22.6/22.1%	25.8/24.6%	28.2/27.4%

It can be observed that the modified better-than normalisation ($\tilde{S}_i^{bt+}$) method yields highest recognition rates in all combination strategies. The probabilistic normalisation $\tilde{S}_i^{p}$ clearly suffers from the quantisation problems described above. If the modified better-than normalisation is applied to self-confidences as well ($\tilde{C}_i^{bt+}$), the overall recognition rate is increased further to 51.6%. Using the Borda measure improved results over the mere Borda rank, particularly in its normalised variation.

Despite its simplicity, the Borda and reverse Borda distances performed well and even outperformed several measurement-based methods. Product combination of modified better-than normalised scores outperforms all other methods and significantly improves over unnormalised product combination that were discussed (and discommended) in [11]. This adverse finding probably derives from the minimum score of 0.1 for target words, which effectively prevents single experts from rejecting correct hypotheses by assigning a score of zero.

All of the investigated combination schemata substantially outperformed individual experts, some more than doubled the best individual expert. This represents an improvement of about factor ten over the average performance of all experts .

The imagined optimal dynamic expert selection obtained from the oracle was also clearly outperformed. This probably results from using only one expert, rather than exploiting agreement and disagreement between different experts. It also derives from the low recognition rates of the individual experts as the target word only rarely appears as top choice in any of them and the oracle is unable to improve beyond the best expert. The rank of the combined score, however, can exceed the best individual rank, i.e. outperform the oracle. This suggests that dynamic selection should only be applied if it can be assumed that at least one of the experts involved performs well.

7 Conclusion

A Wildcard Word Combination method has been presented that can deal with sparse letter graphs. It exploits the concepts of wildcards, wildcard likelihoods, explanation of wildcards and surplus letters as well as the optimised combination of multiple experts. It has shown an improved recognition rate against a comparable method (Powalka's MIS, [2]) if applied to 'poorer' handwriting (51.6% against 38.5%). Although a general quantitative comparison is difficult due to the proprietary data sets used, the comparison with the MIS on identical data allows to draw the following conclusions. Using wildcards reduces the number of correct word hypotheses that cannot be evaluated by the segmentational stage due to sparseness of the graph and therefore increases the overall performance even if only some of the additional target words are scored top. Secondly, parallel combination of multiple experts can improve performance beyond the best individual expert, especially if optimised weighting is employed

rather than naïvely assuming all experts of equal importance. Furthermore, it has been shown, that normalisation of scores allows to combine measurement-level experts based on different scoring mechanisms without resolving to the rank-level as in [8].

As most errors derived from excessively sparse letter graphs (particularly of letters 'b', 'k' and 'x') a selective improvement in letter recognition rate is expected to improve performance further. Although the HFI currently uses on-line data, the principles of the WCC method should be applicable to recognition of static cursive handwriting as well. Identifying and removing redundant experts could reduce computational demands and is therefore envisaged for further research although preliminary experiments indicate that removing several seemingly ‘unnecessary’ experts can impede overall performance and that some ‘redundant’ experts might still be important as a balance against distortion by spurious scoring of individual experts. Furthermore, whilst use of empirically derived thresholds have provided a starting point it is acknowledged that further work in developing a rigorous method which allows repeatability of the experiments is needed. This development also forms part of the ongoing research.

8 References

[1] Lu, Y. and M. Shridhar, “Character Segmentation in Handwritten Words - an Overview”, in Pattern Recognition, vol. 29, pp. 77-96, 1996.

[2] Powalka, R. K., N. Sherkat, L. J. Evett, and R. J. Whitrow, “Multiple Word Segmentation with Interactive Look-Up for Cursive Script Recognition”, in Second Int. Conf. On Document Analysis and Recognition ICDAR'93, pp. 196-199, Tsukuba Science City, Japan, October 1993.

[3] Higgins, C. A. and D. M. Ford, “On-Line Recognition of Connected Handwriting by Segmentation and Template Matching”, International Conference on Pattern Recognition ICPR'92, pp. 200-203, 1992.

[4] Wesolkowski, S., “Cursive Script Recognition: A Survey”, in Handwriting and Drawing Research, Basic and Applied Issues, selected papers from the IGS'96, ed. M. L. Simner, C. G. Leedham, A. J. W. M. Thomassen, pp. 267-284, IOS Press, Amsterdam, 1996.

[5] Hennig, A., N. Sherkat, and R. J. Whitrow, “Recognising Letters in on-line Handwriting with Hierarchical Fuzzy Inference”, Fourth International Conference on Document Analysis and Recognition ICDAR'97, pp. 936-940, Ulm, Germany, 18 August 1997.

[6] Hennig, A. “Recognising a Page of Unconstrained Cursive Handwriting ”, PhD-Thesis, The Nottingham Trent University, Nottingham, UK, June 1998.

[7] Hennig, A., N. Sherkat, and R. J. Whitrow, “Zone Estimation for Multiple Lines of Handwriting Using Approximating Spline Functions”, in Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition IWFHR-5, pp. 325-328, Essex, England, September 1996.

[8] Brown, M. K. and S. Ganapathy, “Cursive Script Recognition”, in 5th Int. Conf. On Cybern. And Society, Boston, pp. 47-51, 1980.

[9] Wang, P., “The Interpretation of Fuzziness”, IEEE Transactions on Systems, Man, and Cybernetics, vol. 26, no. 2, pp. 321-326, 1996.

[10] Huang, Y. S. and C. Y. Suen, “Combination of Multiple Classifiers with Measurement Values”, in Second Int. Conf. On Document Analysis and Recognition ICDAR'93, pp. 598-601, Tsukuba Science City, Japan, October 1993.

[11] Kittler, J., “Improving Recognition Rates by Classifier Combination”, in Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition IWFHR'96, pp. 81-129, Essex, England, September 1996.

[12] Borda, J.-C. de, “Essai sur l'Application de l'Analyse a la Probabilite des Decisions Rendues a la Pluralite des Voix”, Histoire de l'Academie Royale des Sciences, Paris, 1781.

[13] Ho, T. K., J. J. Hull, and S. N. Srihari, “Decision Combination in Multiple Classifier Systems”, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI, vol. 16, no. 1, pp. 66-75, 1994.

List of Figures