On-Line Handwriting Recognition Based on Bigram Co-occurrences

Adnan El-Nasan, George Nagy
DocLab, Rensselaer Polytechnic Institute, Troy, NY 12180
elnasan@rpi.edu

Abstract

We propose a handwriting recognition method that utilizes the n-gram statistics of the English language. It is based on the linguistic property that very few pairs of English words share exactly the same letter bigrams. This property is exploited to bring context to the recognition stage and to avoid segmentation. The recognition is based on detecting bigram co-occurrences. Even with naive features and a limited reference set, it recognizes over 45% of lexicon words that it has never seen before in handwritten form.

1. Introduction

The proposed recognition system is based on detecting letter bigrams, or longer segments, from a feature-level representation of word patterns. The system has access to a *lexicon*, and a *reference set*. The lexicon is the set of all plausible words. Words in the reference set are words from the lexicon for which we have some feature representation, i.e., ink traces. The recognition system consists of three stages: lexical processing, signal matching and classification. The *lexical stage* pre-computes the bigram match properties for each word in the lexicon by matching the label of a lexicon word against the label of each reference word. The *signal matching* stage reports the length of the longest matching segment between the feature representation of the unknown and the feature representation of each reference word. The *classification* stage then finds a label from the lexicon that has match properties that best resemble the match properties of the unknown.

A letter n-gram is a sequence of *n* consecutive letters. N-grams have been investigated since the sixties. Raviv introduced Markov models to OCR [9] and Shinghal and Toussaint applied the Viterbi algorithm [10][11]. Hull and Srihari quantized n-grams probabilities [6] and combined them with dictionary lookup [7]. Suen tabulated the growth in the number of distinct n-grams as a function of vocabulary size [12]. The entropy of n-grams for $n \le 5$ is computed in [8]. We have not, however, found any study of n-gram co-occurrences between pairs of words. We introduced letter n-gram co-occurrences for word discrimination in [1]. In [2] we showed that with a reasonable number of reference words, bigrams represent the best compromise between the recall ability of single letters and the precision of trigrams. We also studied the performance of an ideal system as a function of lexicon and reference set sizes.

Partial-word matching was introduced by Hong and Hull for patterns from the same source with similar shapes [3][4]. Feature-level bigram detection, using partial-word matching, combines some of the advantages of character-level and word-level recognition. Like character-based recognition, vocabulary is expandable and recognition is not limited to words with explicit samples in a training set. However, feature-level bigram detection is more stable than character-based recognition because it avoids segmentation and uses ligatures to match longer segments. Like the widely used Hidden Markov methods [5], feature-level bigram detection brings context into the recognition stage instead of relegating it to post-processing. Unlike HMM, it requires the estimation of only two parameters and storing a reference set of representative patterns and their labels.

2. Method and Notation

Figure 1 shows the components of the system.

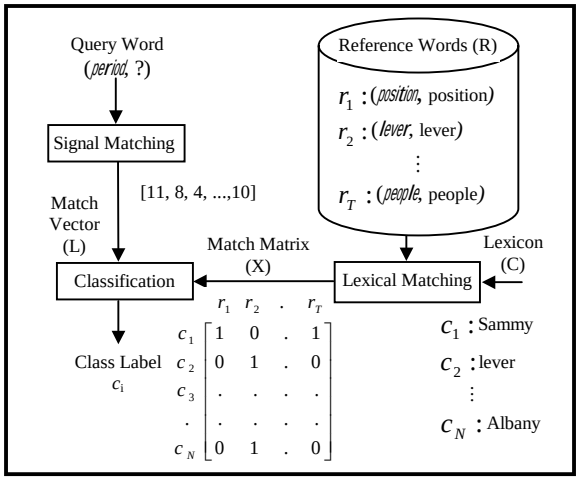

Figure 1. Data flow

2.1. Lexical Pre-Processing

Definitions:
Lexicon (C): a set of *N* valid words, i.e., the set of all valid labels. $C = \{c_i : 1 \le i \le N\}$, where c_i is the *ith* lexicon word.
Reference Set (R): a set of feature-level strings.
$R = \{r_j : 1 \le j \le T \le N\}$, where r_j is the *jth* feature string of elements from a feature space $\Im$, and *T* is the number of reference words.

The lexical stage pre-computes a binary *match matrix X* by matching the label of each lexicon word against the label of each reference word. Each row corresponds to a word in the lexicon and each column corresponds to a reference word. A "1" indicates that the indexed words share at least a letter bigram (Table 1).

$$X = \begin{bmatrix} B_1 \\ B_2 \\ . \\ . \\ B_N \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & . & . & b_{1T} \\ b_{21} & b_{22} & . & . & b_{2T} \\ . & . & . & & . \\ . & & & . & . \\ b_{N1} & . & . & . & b_{NT} \end{bmatrix}$$

Table 1. Example of a match matrix: 4 reference words and 8 lexicon words.

Lexicon \ Reference	have	lever	people	position
period	0	1	1	1
position	0	0	0	1
lever	1	1	1	0
people	0	1	1	0
open	0	0	1	0
ever	1	1	0	0
have	1	1	0	0
hazard	1	0	0	0

2.2. Signal Matching

The signal matching stage generates a feature match vector $L = [l_1, l_2, l_3, ..., l_T]$ where l_j represents the length of the longest common subsequence between the *query* and reference word r_j.

Words are represented as strings of feature symbols. These symbols represent extremal points, cusps and intersections of the trace of the stylus. Each of these features is assigned a label from the alphabet (Figure 2).

The string representation of the word is constructed by analyzing its coordinate sequence and concatenating the corresponding feature labels (Figure 3).

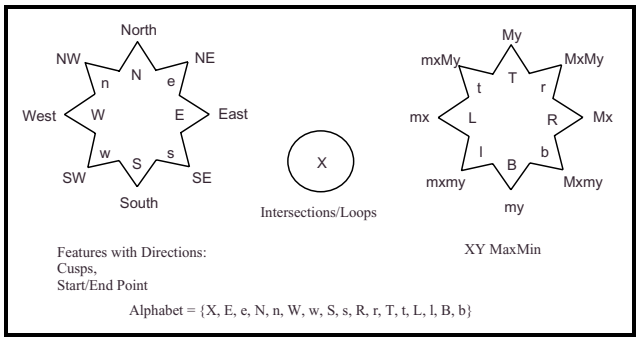

Figure 2. Features and feature labels

The longest common subsequence (LCS) between the unknown and each of the reference words is now determined. Figure 4 shows the LCS between two words. The length of the LCS will be used to determine the presence or absence of a bigram match between a reference word and a lexicon word.

Figure 3. The feature string of the word *has*.

2.2.1. Detecting Bigram Matches

We model detecting lexical matches given a feature-level match length l_i as a two-class problem: *matches* (M) vs. *no matches* ($\overline{M}$). We expect that the probability of a lexical match is higher for longer matches.

Figure 4. Corresponding segments between the words *totally* and *established* are highlighted.

2.3. Classification

We can formulate the classification problem as choosing the lexical word c_i, represented with respect to the reference set by $B_i = [b_{i1}, b_{i2}, b_{i3}, ...b_{iT}]$ of binary values, given vector $L = [l_1, l_2, l_3, ...l_j, ...l_T]$ of match lengths.

$$P(c_i|L) = \frac{P(c_i)P(L|c_i)}{P(L)} = \frac{P(c_i)}{P(L)} \prod_{j=1}^{T} P(l_j|c_i)$$
$$= \frac{P(c_i)}{P(L)} \prod_{j=1}^{T} \left[P(l_j|M)\right]^{b_{ij}} \left[P(l_j|\overline{M})\right]^{1-b_{ij}}$$

The query word *q* represented by its feature match vector will be classified to class c^*, where

$$c^* = \arg\max_i P(c_i) \prod_{j=1}^{T} \left[P(l_j|M)\right]^{b_{ij}} \left[P(l_j|\overline{M})\right]^{1-b_{ij}}$$

$P(l_j|c_i)$ is the probability that query word q and reference word r_j exhibit a feature-level match of length l_j, where q has the same lexical label as c_i. Therefore $P(l_j|c_i) = P(l_j|M)$ if c_i has a lexical match with r_j, and $P(l_j|c_i) = P(l_j|\overline{M})$ otherwise. The presence or absence of a lexical match between c_i and r_j is indicated by $b_{ij} = 1$ or $b_{ij} = 0$, respectively.

$P(l_j|M)$ and $P(l_j|\overline{M})$ are estimated by fitting binomial distributions to the empirical distributions of the feature-level match lengths among the words of the reference set. *Word-based binomials* are approximated using the observed distribution of the match lengths of query word r_j with every other reference word. *Global binomials* $\left(P(l_j|M) = P(l|M), P(l_j|\overline{M}) = P(l|\overline{M})\right)$ are approximated using every pair of words in the reference set (Figure 5).

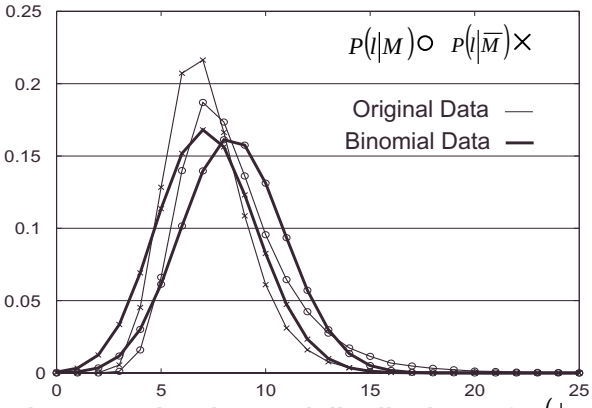

Figure 5. The observed distributions of $P(l|M)$ and $P(l|\overline{M})$ and their Binomial estimates.

3. Experiments and Preliminary Results

The words were written without any constraints on a CrossPad, by a single writer. We randomly selected two mutually exclusive sets of samples (words ranging from 5 to 15 characters): a reference set *RSe*t, and a test set *TSe*t. Less than 50% of distinct word instances that have the same label appear in both *RSet* and *TSet*. Table 2 describes the statistics of the database and of *RSet* and *TSet*.

Table 2. Statistics of data used in testing

	Database	RSet	TSet
Size	5940	1000	1000
Lexically unique	1661	674	660
Characters (average)	1-25 (4.3)	5-15 (7.32)	5-15 (7.33)

We modeled the class-conditional distributions as Binomials and estimated their parameters either separately for each reference word, or for all the reference words together.

3.1. Word-Based Binomials

Some reference words have more discriminating match properties than others. This means that lexical matches can be detected more accurately. We estimate the distributions $P(l_j|M)$ and $P(l_j|\overline{M})$ for each word by matching each reference word against the 999 other words in *RSet*.

3.2. Global Binomials

We match each of the 1000 words of *RSet* against the 999 other words. The resulting feature-level match lengths will be accumulated into two classes based on whether a lexical match exists or not between the matched words. Figure 5 shows the distribution of the feature-level match lengths and their Binomial models.

3.3. Preliminary results

We study the effect of adding new words to the reference set and of increasing the size of the lexicon, on a fixed set of 100 words from *TSet*. Each of these words will be used as a query and will be matched against the reference words to generate a feature match vector.

Figure 6 shows how the accuracy improves as the number of reference word increases. Figure 7 shows how performance degrades as the size of the lexicon increases given a fixed number of reference words. Both Figures 6, and 7, suggest that word-based estimation is more accurate, despite the small sample size, than global estimation.

4. Discussion

The accuracy of the system improves as the number of the reference words increases because additional reference words compensate for matching errors due to letter-form or stroke variations. As the size of the lexicon increases, given a fixed reference set, performance degrades

as a result of attempting to pack more samples in the fixed-size feature space.

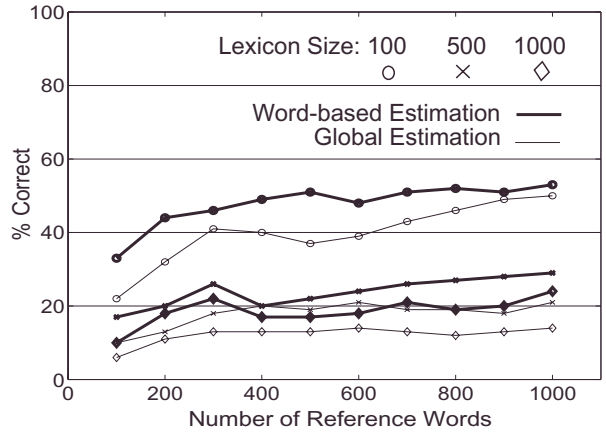

Figure 6. Performance as a function of the number of reference words

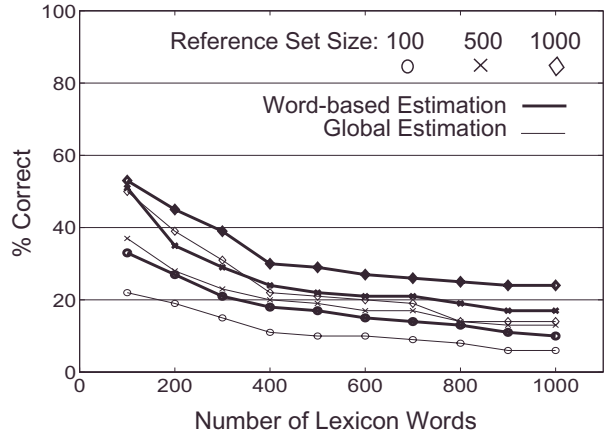

Figure 7. Accuracy as a function of the size of Lexicon

We showed in [2] how the system behaves on simulated data, i.e., when the detection of the presence or absence of bigram matches is perfect. Perfect detection of the presence or absence of a bigram match is possible only when $P(l|M)$ and $P(l|\overline{M})$ have disjoint support. From Figure 5, notice that these distributions are almost indistinguishable, yet we achieve an accuracy of about 50%. As expected, the accuracy on words which appear once or twice in the reference set is about the same as on unseen words. On the few words which have multiple representations in *RSet*, the accuracy is over 80%.

Bigram detection was accomplished using a basic set of features and simplistic string matching. We are currently modifying the features and signal matching routines to improve the separation between the class-conditional distributions. We plan to use features that are more expressive and implement more elaborate approximate string matching. Other information, such as an estimate of the length of the query word, and an estimate of the location of each bigram match, will also be utilized.

We will make use of standard word frequencies to resolve multiple candidates. These will eventually be modified to account for the writer's own word-usage statistics. We will consider using dynamic word-transition models as well.

Acknowledgment

We thank Yarmouk University, Jordan, for their financial support. We are grateful to Harsha Veeramachaneni for his sound comments and suggestions.

References

[1] A. El-Nasan and G. Nagy, "Ink-Link," *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 2, pp. 573-575, Barcelona, 2000.

[2] A. El-Nasan, S. Veeramachaneni and G. Nagy, "Word discrimination based on bigram co-occurrences," *Proceedings of the 6th International Conference on Document Analysis and Recognition*, pp. 149-153, Seattle, 2001.

[3] T. Hong and J. Hull, "Character segmentation using visual inter-word constraints in a text page," *Proceedings of the SPIE - The International Society for Optical Engineerin*g, vol. 2422, pp.15-25, 1995.

[4] T. Hong and J. Hull, "Visual inter-word relations and their use in OCR post-processing," *Proceedings of the Third International Conference on Document Analysis and Recognitio*n, pp. 442-445, 1995.

[5] J. Hu, M. Brown, Turin W., "HMM Based Online Handwriting Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligenc*e, vol. 18, no. 10, pp. 1039–1045, 1996.

[6] J. Hull and S. Srihari, "Experiments in text recognition with binary n-grams and Viterbi algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligenc*e, vol. 4, no. 5, pp. 520–530, 1982.

[7] J. Hull, S. Srihari and R. Choudhari, "An integrated algorithm for text recognition: comparison with a cascade algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligenc*e, vol. 5 no. 4, pp. 384–395, 1983.

[8] H. Kucera and W. Francis, *Computational analysis of present-day American Englis*h, Providence, RI, Brown university press, 1967.

[9] J. Raviv, "Decision making in Markov chains applied to the problem of pattern recognition," *IEEE Transactions on* Information Theory, vol. 3, no. 4, pp. 536-551, 1967.

[10] R. Shinghal, G.T. Toussaint, "Experiments in text recognition with the modified Viterbi algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligenc*e, vol. 1, no. 2, pp. 184–193, 1979.

[11] R. Shinghal, G.T. Toussaint, "The sensitivity of the modified Viterbi algorithm to the source statistics," *IEEE Transactions on Pattern Analysis and Machine Intelligenc*e, vol. 2, no. 2, pp. 1181–1184, 1980.

[12] C. Suen, "N-gram statistics for natural language understanding and text processing," *IEEE Transactions on Pattern Analysis and Machine Intelligenc*e, vol. 1, no. 2, pp. 164–172, 1979.