INK-LINK

Adnan El-Nasan, George Nagy
Rensselaer Polytechnic Institute (RPI)
Troy, NY, 12180 USA
elnasan@ecse.rpi.edu, nagy@ecse.rpi.edu

Abstract

Wide acceptance of inexpensive writing tablets with high functionality motivates the development of individualized, adaptive on-line recognition of cursive script. We demonstrate a lexical algorithm based on bigram matches. The solution we propose is to (i) Generate a match list by partial-word matching against a reference list in the owner's script. (ii) Identify each unknown word by eliminating, from a large lexicon, every word that partially matches the transcript of any word on the reference list that is not on the match list, or that fails to match any word on the match list. With perfect feature-level matching, a surprisingly short reference list yields a high recognition rate.

1. Introduction

Recent research in Intelligent Character Recognition (ICR) has focused on off-line, multi-writer applications such as bank-check interpretation [3], postal sorting [9], and form reading [12]. These applications make effective use of restricted vocabularies for various document components, of inter-field redundancy, and of external databases (legal and courtesy amount agreement; street, city, state, zip-code; item-name, item-number, price; customer-name, customer-address; or entry, subtotal, total). In some forms-reading applications, additional constraints, like boxes, combs, or examples of preferred writing styles, may be preprinted, possibly in a drop-out color. Furthermore, in such large, centralized applications an operating point with a 30% reject rate and a 0.1% field error rate may be economically acceptable.

We are interested, in contrast, in decentralized, individualized, unconstrained applications of on-line recognition. The advent of slim, inexpensive tablets with built-in processors will give new life, with new requirements, to on-line recognition. An early attempt in this direction, the Apple Newton, failed partly because of unsophisticated recognition and burdensome constraints imposed on writers [13].

Successful systems will require an essentially open vocabulary and accept normal writing style. Furthermore, such systems must improve with use and eventually recognize most input. The writer cannot be expected to correct the same mistakes day after day. However, each system will be individualized and recognize only the writing of its owner. It will also be possible to share a tablet with multiple writer profiles, each tuned to one person's idiosyncratic calligraphy and vocabulary.

Current approaches to on-line recognition can be divided into character-level [1] and word-level [8] classification. However, in cursive writing individual characters are indistinct, while whole-word recognition is restricted to a limited vocabulary. Combining the two approaches by partial-word matching eliminates the principal shortcomings of both. Partial-word matching was suggested by Hong and Hull [4,5].

Our input device is the CrossPad [6]. The writing tool is a wireless stylus, the writing surface is a paper pad affixed to an electronic tablet. The writer may (a) save the paper copy, (b) save the "electronic ink" for subsequent printing or display, or (c) upload the file to a computer for recognition.

Tablets will soon either have enough built-in processing power to carry out stand-alone recognition, or remain in wireless communication with the appropriate processor. Such tablets may also be part of fixed or portable appliances as an alternative to keyboard or speech input.

2. Method

The elements of the proposed recognition system are an "ink" phase and a "lexical" phase. New word traces or parts thereof are matched to the writer's script, labeled by lexical processing, and then added to the script database. We summarize the proposed feature-level and bigram-based lexical-level processing. We then present experimental results on

the lexical phase, initially assuming perfect matching at the feature level.

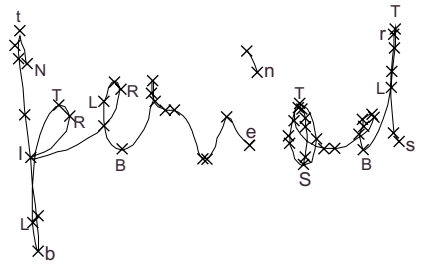
2.1. Ink phase

The page is analyzed to isolate word traces that are to be recognized (*target words*). Some words may consist of multiple strokes (pen-down to pen-up).

Simple features, consisting of horizontal and vertical extrema, initial stroke directions, and stroke crossings, are extracted. Improved feature extraction is not an objective of our current research, but of course the more stable and discriminating the features, the better.

The features are sorted according to horizontal position or time. (The tablet records 10 dots per mm at a frequency of 100Hz.). Each target word is represented as a *target string* of variable length corresponding to the number of features, defined over the alphabet of feature types (Figure 1). Partial-word matching against a list of labeled feature strings, called *reference strings*, is then based on a string-edit computation [10,11]. The feature string is compared to some or all of the reference strings. The label of each reference string that matches at least a two-letter segment of the target string is saved for lexical processing. The collection of these labels is called the *match list*. The length of the match list grows with the length of the reference list, resulting in one means of improved recognition with use.

Examples of some reference words that appear in a simulated match list for the target word "period" are shown in Figure 2. The "ink" information contained in the target feature string has been transformed into the lexical information of the labels of the match list.



NtrLLRbBLLTRlBRRTLLBRrLTLBTBBTTBennS
BLRLRRLRTTLLBRTLLBBRTTLRLLBBRTrLRLLs

Figure 1. Feature string extracted *period*. Some features are labeled on the trace.

The system will be initialized with each user's favorite words. The first few dozen entries cannot be recognized because the reference list is almost empty. In most applications, they will be transcribed using a keyboard when the file is uploaded to a computer. In others, they may be entered with bootstrap ICR software for block letter recognition, via speech input, or by pointing the stylus to a facsimile keyboard on the tablet.

2.2. Lexical phase

The lexical phase makes use of a large lexicon of common words. The lexicon is augmented with recognized writer-specific words, typically proper nouns and abbreviations, much like a spelling lexicon. The lexical phase operates on ASCII reference and lexicon words to select a unique lexicon word that corresponds to the labels of the reference strings that matched part of the target feature string. In Figure 2, the match list for the word "*period*" consists of six words. These words are all the words in the reference list that share at least one bigram with the target.

Reference List		Match List	Lexicon	Rule 1	Rule 2
body	body	*	attitude	have	
civil	civil		aviation	civil	
Erie	Erie	*	body		people, ...
have	have		civil	civil	
it	it		consequences	civil	
lever	lever	*	Erie	state	
over	over	*	have	have	
people	people	*	it	it	
position	position	*	lever	have	
state	state		masters	state	
			officer		body, ...
			over	over	
			people	have	
			period		
			position	it	
			preferably		body, ...
			proceedings		Erie, ...
			prohibition	it	
			several	have	
			state	state	

Figure 2. Reference words are shown with their lexical transcripts. The six that match *period* are indicated by asterisks. Rules 1 and 2 are applied to the Lexicon, the Reference List and the Match List to recognize *period*.

Every word in the lexicon is tested using the following rules. (1) Lexicon words that match at least one bigram of *any* word on the reference list that is *not* on the match list are eliminated (e.g., "*aviation*"

in the example of Figure 2). This rule ensures that the chosen lexicon word does not contain any bigram that is not in the target word.
(2) Lexicon words that fail to match at least one bigram of *every* word on the match list are eliminated. (e.g., "*body*"). This rule ensures that the selected lexicon word contains every bigram that appears in the target word. In the example of Figure 2, only "*period*" was selected from the lexicon.

3. Simulation Results

We present simulation results on our hand-labeled single-writer database of 4952 words (1498 unique words) and on the Brown Corpus. Initial and terminal blanks were included with each word and taken into consideration in forming the match list. We haven't yet run experiments on the combined feature extraction and lexical phases.

The fraction of target words that are correctly identified is shown, as a function of the length of the reference list, in Table 1. These lists are subsets of our lexicon of 1498 words and they mirror the frequency distribution of words in the single-writer database.

Table 1. Fraction of correctly recognized target words (test results on all1498 words).

Reference List Size	Correct	Error	Reject
100	93.12 %	0.0 %	6.88 %
300	99.07 %	0.0 %	0.93 %
1000	99.60 %	0.0 %	0.40 %
1498	99.87 %	0.0 %	0.13 %

Table 2 shows the number of correctly recognized words from the 1,013,253 word Brown Corpus. Even on the full collection of 43,300 unique words the number of rejects is low, although the only order information we use so far is what is implicitly contained in overlapping bigrams.

Table 2. The Brown Corpus recognition rates for different Lexicon and Reference List sizes.

Lexicon / Reference	1000 (69.6 % of corpus)	10,000 (92.8 %)	43,300 (100 %)
100	84.00 %	75.12 %	66.94 %
300	98.40 %	97.58 %	95.25 %
1000	99.80 %	99.57 %	98.66 %

All rejects are due to multiple candidates in the lexicon that match exactly the same reference words, like: *sly-slyly*; *possess-possesses*; *kaola-kola*; *mm-mmm-mmmm*; *rocco-rococo*; *award-awkward*; *shed-she'd*; *willfully-wilfully*; *bee-beebe*; *lewellyn-llewellyn*; *latter-later*. With the growth of the reference list, the only rejects that can not be eliminated are words that have identical bigrams, such as *asses* and *assess*. An error can occur only if the target word is not in the lexicon.

4. Discussion

We believe that the main advantage of the proposed method over character-based models is that it exploits the consistency of inter-character ligatures. Furthermore, it is easier to match longer and more complex traces than the elementary and often vestigial strokes of single letters.

The advantage over whole-word recognition is that the vocabulary is limited only by the size of the expandable lexicon, rather than by the size of the reference set or training sample. Many spelling dictionaries already include common proper names and abbreviations.

In the "ink" phase, we cannot expect accurate matches. We could, however, take into account the estimated length of the target word to select lexical candidates. We could also eliminate lexical candidates if their match positions, estimated from average character lengths, are off by more than one or two characters. We plan to implement distance-sensitive approximate string matching.

Unreliable feature extraction and string matching will also require relaxing the very strict conditions of rules (1) and (2). The results of the lexical recognition procedure are not altered by the presence of multiple instances of the same word in the reference list. In matching electronic ink, however, redundancy provides increased opportunity for recognition.

To resolve multiple candidates, we will eventually make use of standard word frequencies as *a priori* probabilities. These will be gradually modified to take into account the writer's own word-usage statistics as represented by the current reference list. We will also consider using dynamic word-transition models.

Aside from the customization of the lexicon, improved recognition will result simply from the growth of the reference list. A longer reference list will increase the probability of covering a new target word. Multiple matches with common character segments (e.g. *the, des, ing*) will increase the

probability of a good match by modeling variations in the writer's style.

Our method does not require complex estimation procedures, such as expectation maximization or neural networks. It might appear that the number of computations in string matching will grow rapidly with the size of the reference list. We will attempt to recognize each target word before the entire match list is determined. Words identified unambiguously will be accepted at an early stage. Intensive computation will be performed only for "difficult" words. Fortunately, the lexical phase presents no problem: excellent data structures and algorithms, such as those used in spell checkers, are already available.

Most the standard databases for on-line cursive writing, such as Unipen [2,7], contain short passages by multiple writers. Most speech databases also contain many speakers, rather than long samples by individual speakers. We need longer samples but not necessarily by many writers. The reference list must grow to several thousands and contain nearly one thousand unique words, before recognition reaches acceptable levels (Table 1). In practice, this may take anywhere from a few hours of constant use (e.g. writing a term paper) to several weeks of intermittent use (taking notes at meetings).

We expect the frequency distributions of individual reference lists to be as skewed as collections of published material (e.g. the Brown Corpus). Many of the most frequent words - names of people, organizations, and local abbreviations - will be specific to the writer, and absent from the initial lexicon. However, the number of successful comparisons should grow more quickly with the size of the reference list than would comparison with a non-writer-specific word list of the same length.

It remains to be shown that feature-level processing can identify bigrams. By the time ICPR'00 is convened, we may be able to demonstrate some results on actual handwriting.

Acknowledgment

We thank Yarmouk University, Jordan, for their financial support.

We are grateful to Dr. Mahesh Viswanathan and the Pen Technologies Group at the IBM T.J. Watson Research Center for providing the data. We are also grateful to Frank LeBourgeois, Harsha Veeramachaneni and Tetsushi Wakabayashi for their valuable comments and suggestions.

5. References

[1] Chan K-F., Yeung D-Y., "Elastic Structural Matching for On-Line Handwritten Alphanumeric Character Recognition", *Proc. Int. Conf. On Pattern Recognition*, vol. 2, 1998, pp. 1508-11.

[2] Guyon I., Schomaker L., Plamondon, R., Liberman M., Janet S., "UNIPEN Project of On-Line Data Exchange and Recognizer Benchmarks", *Proc. 12th IAPR Int. Conf. on Pattern Recognition*, vol. 2, 1995, pp. 29-33.

[3] Heutte L., Barbosa-Pereira P., Bougeois O., Moreau, J.V., Plessis, B., Courtellemont P., Lecourtier Y., "Multi-bank Check Recognition System: Consideration on the Numeral Amount Recognition Module", *Int. J. Pattern Recognition and Artificial Intelligence*, vol. 11, no. 4, June 1997, pp. 595-618.

[4] Hong T., Hull J., "Visual Inter-Word Relations and Their Use in OCR Post-Processing", *Proc. Int. Conf. on Document Analysis and Recognition*, vol. 1, Aug. 1995, pp. 442-5.

[5] Hong T., Hull J., "Algorithms for Post-Processing OCR Results with Visual Inter-Word Constraints", *Proc. Int. Conf. on Image Processing*, vol. 3, Aug. 1995, pp. 312-15.

[6] http://www.crosspad.com/

[7] http://hwr.nici.kun.nl/unipen/

[8] Hu J., Brown M.K., Turin W., "HMM Based Online Handwriting Recognition", *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, 1996, pp. 1039-45.

[9] Mahadevan U., Srihari S.N., "Parsing and Recognition of City, State, and ZIP Codes in Handwritten Addresses", *Proc. Fifth Int. Conf. on Document Analysis and Recognition*, 1999, pp. 325-328.

[10] Stephen G.A., " String Searching Algorithms", *Lecture Notes Series on Computing*, vol. 3, World Scientific, Singapore, 1994.

[11] Wagner R., Fischer M., "The String-to-String Correction Problem", *J. ACM*, 21(1), 1974, pp. 168-173.

[12] Watanabe T., Luo Q., Sugie N., "Layout Recognition of Multi-Kinds of Table-Form Documents", *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, April 1995, pp. 432-45.

[13] Yaeger L., Webb B., Lyon R., "Combining Neural Networks and Context-Driven Search for Online, Printed Handwriting-Recognition in the Newton", *Lecture Notes in Computer Science*, vol. 1524, 1998, pp. 275-98.