Journal of Multimedia Information System VOL. 10, NO. 2, June 2023 (pp. 163-168): ISSN 2383-7632 (Online)
http://doi.org/10.33851/JMIS.2023.10.2.163

Brief Paper:
Knowledge-Based Post-Processing of On-Line Hangeul Short-HandWriting Recognition

Taeho Lee[1], Bong-Kee Sin[2*]

Abstract: In this paper we propose a technique of correcting online handwriting recognition errors using morphemic lexicons and a set of confusion lists for mistaken characters collected from classification errors. Given a string of Hangeul characters returned by a baseline recognizer, we raise a set of alternative characters out of a confusion dictionary, one for each character in the returned string. Then we conduct a post-processing via a dynamic programming-based lattice search for a sequence of legal words while consulting morphemic lexicons. The proposed methods can handle short broken phrases in natural online handwriting of sentences. Experiment shows that, for handwriting strings of 3.09 characters long on average, the proposed method improved the phrase recognition accuracy from 55.43% before post-processing to 86.63%, an error reduction of 70.00%, with single hypotheses.

Key Words: Online Handwriting Recognition, Post-Processing, Knowledge-Based, Confusion, Lexicon.

I. INTRODUCTION

Online handwriting recognition is the core technology for generating documents using an electronic stylus and a digitizer tablet. In Korean Hangeul, a large number of research efforts have been reported with varying degree of success in isolated character recognition [1-3]. Handwriting recognition still remains an unsolved problem. Future deep learning may come to the rescue, but it will take a long while because of the vastly diverse styles, and bad or ambiguous writings which make shape-based recognition insufficient. It is a general consensus that we need to incorporate linguistic and domain knowledge to make sense of the shape-only result.

In many online handwriting environments, people usually write a number of characters at a time that tend to make a structural or conceptual chunk. When characters are analyzed in isolation, word or phrasal errors can grow exponentially. For instance when there are three input characters for a classifier with 83% recognition rate, the string accuracy can drop to 0.83^3~0.56. But we know that the characters thereof are not independent. Rather they usually make legal words together and implement local grammatical structures.

The proposed method is based on two kinds of knowledge bases, recognizer-specific confusion dictionary and linguistic morphemic dictionaries. Given a garbled string of Hangeul characters returned by a baseline handwriting recognizer, we create a set of alternative characters out of the confusion dictionary, one for each character in the input string. Then we carry out a dynamic programming-based lattice search over a proposed grammar network for a sequence of legal words while filtering through morphemic lexicons. This algorithm is highly efficient with a linear time complexity which makes it suitable for real time application.

With the proposed method of lattice search guided by the two knowledge bases, the recognition rate has improved from 55.43% to 86.63% with a single hypothesis. It translates to 70.00% error reduction. This effect continues when multiple candidates are counted, and when measured within classified topics.

The rest of the paper consists as follows: Section 2 makes a quick review of related works. Section 3 characterizes the baseline character recognition method. And then Section 4 describes recognizer-specific confusion lists and the domain-specific knowledge-base, the basis for the proposed correction technique. Section 5 presents experimental results. Finally Section 6 will conclude the paper.

II. RELATED WORKS

The literature for post-processing for OCR is rather small [4-5]. In Korean, we find a survey paper published as early as 1993 [6]. This lead to two extensive efforts conducted by

Manuscript received May 28, 2023; Revised June 20, 2023; Accepted June 23, 2023. (ID No. JMIS-23M-05-022)
Corresponding Author (*): Bong-Kee Sin, +82-5-629-6256, bkshin@pknu.ac.kr
[1]Department of AI Convergence, Graduate School, Pukyong National University, Busan, Korea, iks6164@pukyong.ac.kr
[2]Department of Computer and AI, Pukyong National University, Busan, Korea, bkshin@pknu.ac.kr

leading teams using contextual information from character recognition results [7-8], which reported respectively an improvement with recognition of 99% up from 95%, and 95.53% from 71.2% based on multi-level post-processing. Since then, however, the research has virtually discontinued. Unlike OCR which usually pertains to printed characters, the problem of offline handwritten character recognition is much harder and thus has seldom been tried except for highly constrained tasks like postal code or address recognition [9].

In online-handwriting recognition, it is similarly hard to find post-processing research papers but for different reasons. Since the early days, most research efforts aimed at recognizing isolated characters or words without serious post-processing. Post-processing, if any, has been minor as a part of recognition steps and usually limited to lexicon lookup [10]. This is partly due to the characteristic of the pen-based user interface with real-time feedback. But repetition of similar errors could cause inconvenience, fatigue, or even lead to stress and frustration.

III. ONLINE HANDWRITING RECOGNITION

3.1. Baseline Recognizer

In order to recognize handwriting text recognition, we have to go beyond isolated character recognizers to accept a sequence of an unknown number of characters. We distinguish two approaches. One is to separate the steps of character segmentation and character recognition. The other approach is to combine them into a simultaneous segmentation and recognition as shown in Fig. 1.

For an alphabet-based character recognizer, a reasonable approach is to model individual letters (or graphemes in Korean) and then concatenate them to model all of the 2,335 whole Hangeul characters. In this paper this architecture is further extended into a circular network to accept a string of characters as shown in Fig. 1. This design allows us to determine the optimal number of characters, their boundaries, and the character labels simultaneously.

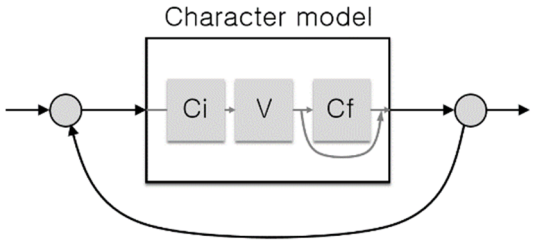


Fig. 1. Hangeul character string model. Ci and Cf stand for the initial and final consonants, and V for the word.

Let $X = X_{1:T} = x_1, \dots, x_T$ be a time-series data corresponding to a sequential input handwriting, and $C_{1:n} = (C_1, \dots, C_n)$ be a sequence of characters and $S_n = (s_1, \dots, s_n)$ the corresponding temporal segmentations with respect to X where $s_r = (t_{r-1} + 1, \dots, t_r)$ denotes r-th time segment; t_r represents the end point of the r-th character and satisfies $t_0 = 0 < t_1 < \cdots < t_n = T$. Let $\Theta = (\theta_1, \dots, \theta_L)$ be the parameter of the recognition model for an L character vocabulary. Then the recognition is defined as the task of computing the triple optimization:

$$(\hat{n}, \hat{C}_{\hat{n}}) = \underset{n, C_n}{\operatorname{argmax}} \max_{S_n} P(\mathrm{X}, C_n, S_n|\Theta), \quad (1)$$

where $\hat{C}_{\hat{n}}$ denotes the best sequence of character labels of the optimal length $\hat{n}$. This can be implemented by a hierarchical extension of the dynamic programming algorithm [11]. In this paper the integrated model Θ is defined as a network of Hangeul grapheme Markov chains θ_l where $l = 1, \dots, L$ [3]. According to Equation (1), we get a string of $\hat{n}$ characters $\hat{C}_{\hat{n}} = (\hat{c}_1, \dots, \hat{c}_{\hat{n}})$ along with, if desired, an optimal set of boundary points $\hat{S}_{\hat{n}} = (\hat{s}_1, \dots, \hat{s}_{\hat{n}})$. There is no word-level language model involved. One will be included in Section IV.

3.2. Recognizer Performance

Among the 2,335 Hangeul characters, many are rarely used in our daily lives. The same is true of many final consonant graphemes. Refer to Table 1. This is a natural consequence of collecting samples from ordinary text. For this reason the training set is so unbalanced across graphemes that it will severely limit the performance. In fact the baseline performance in isolated character recognition is around 83%.

IV. POSTPROCESSING OF ONLINE HANDWRITING

4.1. Overview of the Proposed Method

This section introduces a method of generating alterna-

Table 1. Hangeul graphemes from small training sets. Roman key labels are given in braces. Figures in braces denotesample size.

	Small-sample graphemes	Most popular graph emes
Initial consonant	ㅱ(Q, 2), ㅆ(T, 6), ㅋ(z, 10)	ㅇ(d, 841), ㄱ(r, 355)
Vowel	ㅒ(O, 1), ㅖ(P, 6), ㅞ(np, 2), ㅙ(ho, 0)	ㅏ(k, 631)
Final consonant	ㄷ(e, 2), ㅈ(w, 3), ㄹㅎ(fg, 3), and twelve other graphemes(0)	ㅇ(d, 567)

tive legal strings by incorporating prior knowledge that includes statistics-based confusion dictionary and categorical lexicons. The actual correction is built upon the top-level dynamic programming search of Equation (1).

4.2. Building Confusion Dictionary

Any pattern classifier is imperfect and tends to make classification errors which are characteristic of its model family, architecture, size, and the amount and quality of training set. We will take advantage of this classifier-dependent information for reversing the potential mistakes and proposing legal words or phrases that make sense. Note that an input string is assumed to contain character-wise classification errors which defy conventional morphological analysis.

Table 2 shows a few sample lists of confusions out of training set. The lists are derived statistically from the given recognizer over the training set. Given an input character in the first column, we suggest possible corrections from the second column which will then be checked for legality using dictionaries in the next section. Fig. 2 shows an example of suggestions, each row in the rightmost box built from a combination of character-wise alternatives.

4.3. Lexicons

In this research we assume a sequence of Hangeul-only handwriting characters. And the recognized sequence is usually a broken or corrupt phrase which does not make sense because of one or more misclassified characters. Each alternative string in the right box of Fig. 2 filters character-by-character through lexicons. There are six separate lexicons each of which corresponds roughly to a morphemic category. This distinction, however, is just for convenience in building lexicons but functionally irrelevant since we are not doing any morphological analysis. The structure of a lexicon is usually a multi-way search but we opted out of it for an equivalent but faster binary search [12].

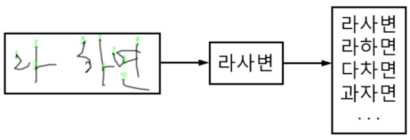


Fig. 2. Alternative strings in the right box for the recognition result in the middle.

Table 2. Confusion list samples (Roman key labels in braces).

Mistaken result	Alternative/original characters
이(dl)	어(dj), 에(dp), 입(dlq), 히(gl), 기(rl), 하(gk),
으(dm)	의(dml), 을(dnf), 은(dms), 므(am), 을(dmf)
라(fk)	과(rhk), 다(ek), 개(ro)

4.4. Post-Processing Correction Algorithm

We model a handwriting string as a sequential concatenation of words or morphemes that corresponds to a short phrase. This concept is organized into a circular network over the six lexicons as shown in Fig. 3. The design implies the search space that can be implemented as a lattice structure over all possible sequence of words. The post-processing correction searches the circular network for the optimal sequence of words. This allows an arbitrary concatenation of legal words or suffixes in any order.

The actual correction is a dynamic programming-based computation of Equation (1) as presented in Fig. 4. It consists of two phases: the forward pass that computes

$$\mathrm{P}(X_{1:t_n}, \hat{C}_n, \hat{S}_n|\Theta) = \max_{C_n \in V, S_n} P(X_{1:t_{n-1}}, C_{n-1}, S_{n-1}|\Theta) \times p(X_{s_n}, c_n, s_n|c_{n-1}, \theta_c), \quad (2)$$

recursively for $t_n = t_{n-1} + 1, \ldots, T$, and $t_0 = 0$. The in-

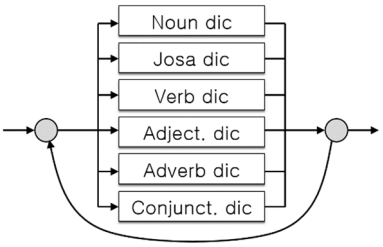


Fig. 3. String correction network with six lexicons.

```
Algorithm: correctHangeulString(X, ConfDic, Lexicon)
input: X = X1,...,XT // recognition string
output: Wordv[5] : array of 5 corrected strings

Hypotheses = {''}
for t = 1:T
    Alt_chars = generate_ConfusedChars(Xt, ConfDic)
    Hypotheses_new = {}
    for a in Alt_chars
        for h in Hypotheses
            h' = modifyHypothesis(h, a, t)
            h'' = expandHypothesis(h', Lexicon)
            Hypotheses_new.append(h'')
        end
    end
    Hypotheses = Hypotheses_new
end
for h in Hypotheses
    [score, wordv] = backtraceHypothesis(h)
    Scorev.append(score)
    Wordv.append(wordv)
end
[score5, top5] = sort5(Scorev) // find top 5 only
return Wordv(top5)
```

Fig. 4. Search algorithm for the best legal word sequences.

nermost for-loop computes Equation (2) and consults lexicons for a new generation of hypotheses. The computation is linear in time, efficient enough for real time application. The second phase is quick backtracking for answer Wordv, the best sequences of legal words and suffixes.

V. EXPERIMENTS

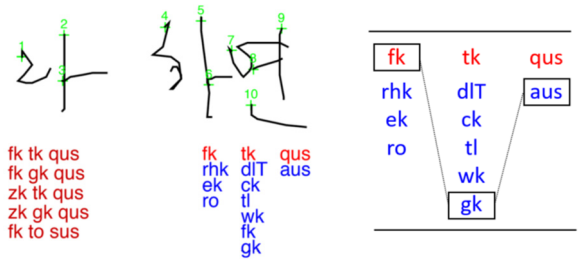
5.1. Experimental Setup

The handwriting test set collected from four subjects consists of 47 documents containing Hangeul phrases as well as mathematical expressions which are ignored in this research. There are in total 351 phrases, each consisting of 1 to 8 characters and 3.12 characters on average. The baseline character recognizer has been prepared separately. Its performance in isolated character recognition task is 83.0%, which implies frequent misclassifications in phrasal writing scenarios.

All the test modules and the search algorithm have been written in MATLAB® core of any version, say 2020R, without any toolbox, for the post-processing as well as handwriting recognition. The key functions include textbook-level dynamic programming and binary search algorithms along with Hashtable supported from Java (*i.e.*, java.util.Hashtable).

5.2. Post-Processing of Recognition Result

Post-processing comprise two steps; first, we generate a list of alternative characters for each of the characters returned from the recognizer. Fig. 5(a) shows an example. The bottom left block of texts are five recognition strings none of which make sense, while the bottom right texts denote three vertical list of alternatives for the first string at the left, 'fk tk qus'(라사변). The second step refers to the search for the best possible legal sequence of words and suffixes. Fig. 5(b) shows one such string 'fk gk aus'(라하면) that makes sense. This is found by the search algorithm given in Section 4.4. This algorithm consults the lexicons through the phrase network of Fig. 3. Given a candidate



(a) (b)

Fig. 5. (a) Recognition result, and (b) three lists of alternative characters from which a legal string 'fk gk aus' is found.

dh fms Whr rm fla rhk rkx ek

(a)

[dh fms Whr]	[rm fla]	[rhk]	[rkx ek]
오른쪽	그림	과	같다
[Noun]	[Noun]	[Josa]	[Adjec]

(b)

Fig. 6. Analysis of a sample input string. Josa is a type of agglutinative suffix in Korean language.

string in Fig. 6(a), it returns a sequence of words or suffixes labeled by lexicon types like Fig. 6(b).

5.3. Performance Evaluation

In the first set of performance measurements we highlight the effect of the post-processing in terms of recognition accuracies compared to those of the baseline recognizer without knowledge base. Table 3 tells us that the proposed method improved the hit rate of 55.43% to 86.63% with a single best hypothesis, a 70.00% reduction of errors. The effect is consistent as we increase the number of returned hypotheses to top five (the bottom row).

In the previous evaluations, character segmentation is critical. When the segmentation fails or the number characters is wrong, we simply count the phrase or all the characters simply as a misclassification since there will be no chance of recovering the correct string. Table 4 shows the accuracy figures when the number of characters are correctly inferred. This hints at the value of proposed method post-processing correction; the recognizer is highly reliable once the number of characters are known or correctly guessed.

The handwriting data set is a collection of documents on six mathematical topics or areas as shown in Table 5. The table shows the performance of the post-processing method for each of the topics. For comparison columns 2 and 3 show the character recognition accuracy with the single best and top five answers, respectively, from the baseline recognizer without post-processing. With post-processing, the accuracy improves to 78.08% to 88.89 % for the topics with the single best, as shown in the fourth column. They

Table 3. Handwriting phrase recognition rates compared and the effect of post-processing in terms error reduction rates (%).

#Hypotheses	1	2	3	4	5
Baseline rec.	55.43	63.79	66.30	67.69	67.69
Proposed mtd	86.63	88.58	90.53	91.09	91.09
Error red. rate	70.00	68.46	71.90	72.42	72.42

Table 4. The performance when the number of characters are known (or guessed correctly) vs unknown (%).

#Hypotheses	1	2	3	4	5
#Chars unkwn	86.63	88.58	90.53	91.09	91.09
#Chars kwn	92.01	94.08	96.15	96.75	96.75

add up to 85.63% to 93.33% with best five hypotheses. The average or the overall performance figures are shown at the bottom. They are 41.86% and 84.29% with the single best candidates.

Fig. 7 illustrates the effect of the proposed method, a simple visualization of Table 5. The broken lines at the bottom represent the baseline recognition rates for the six topics with increasing number of hypotheses. Whereas the group of solid lines at the top correspond to the accuracies of the proposed method.

Table 5. Performance of the post-processing (%).

Topics \ #Hypotheses	Baseline	Recognizer	Proposed	Method
	1	5	1	5
Algebra	28.45	40.52	84.48	93,10
Calculus	38.36	47.95	78.08	87.67
Geometry	47.41	58.52	87.04	90.37
Prob/Stat	37.72	43.11	80.84	85.63
Math I	53.33	57.78	88.89	93.33
Math II	58.62	75.86	86.21	93.10
Overall	41.86	51.43	84.29	89.91

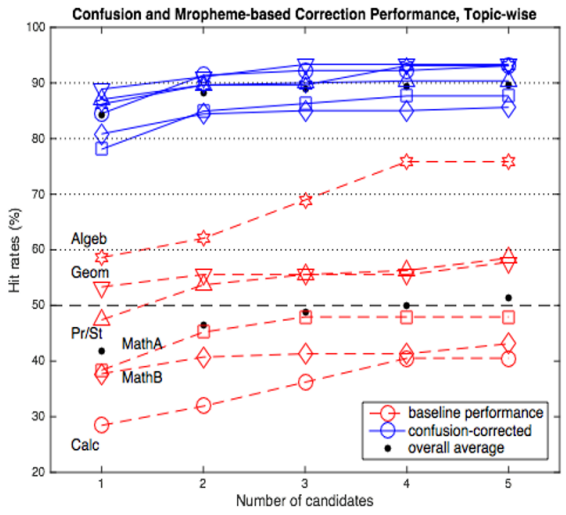


Fig. 7. The effect of the post-processing correction. For performance figures of the first and last columnar points, refer to Table 5.

VI. CONCLUSION

This paper presents a knowledge-based method of post-processing correction of online Hangeul handwriting recognition strings. The knowledge-base includes the confusion dictionary derived from the character recognizer and domain-dependent lexicons of words and suffixes. It has been tested in detail and has confirmed that (1) the proposed post-processing is highly effective in reducing the error rate by 70.00%, (2) the effectiveness continues when tested across six domains in secondary school mathematics. Finally, (3) an added bonus is that an input string is segmented into a sequence of legal Hangeul words which, although possibly not accurate, is good enough for our goal of post-processing correction and is made to make sense.

ACKNOWLEDGMENT

This work was supported by a Research Grant of Pukyong National University (2021).

REFERENCES

[1] B. H. Kim and B. T. Zhang, "Hangul handwriting recognition using recurrent neural networks," *KIISE Transactions on Computing Practices*, vol. 23, no. 5, pp. 316-321, 2017.

[2] H. Kim and Y. Chung, "Improved handwritten Hangeul recognition using deep learning based on GoogLeNet," *The Journal of the Korea Contents Association*, vol. 18, no. 7, pp. 495-502, 2018.

[3] B. K. Sin, "Augmentation of hidden Markov chain for complex sequential data in context," *Journal of Multimedia Information System*, vol. 8, no. 1, pp. 31-34, 2021.

[4] C. Rigaud, A. Doucet, M. Coustaty, and J. P. Moreux. "ICDAR 2019 competition on post-OCR text correction," *The 15th International Conference on Do-cument Analysis and Recognition*, 2019, pp. 1588-1593.

[5] T. Nguyen, A. Jatowt, M. Coustaty, and A. Doucet, "Survey of post-OCR processing approaches", *ACM Computing Surveys*, vol. 54, no. 6, pp. 1-37, 2021.

[6] B. Min, S. W. Lee, and H. Kim, "A study on post-processing for character recognition," in *Proceedings of the Character Recognition Workshop*, 1993, pp. 91-103.

[7] H. C. Kwon, H. J. Hwang, M. J. Kim, and S. W. Lee, "Contextual post-processing of a Korean OCR system by linguistic constraints," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Aug. 1995, vol. 2, pp. 557-562.

[8] G. Lee, J. H. Lee, and J. Yoo, "Multi-level post-processing for Korean character recognition using morpho logical analysis and linguistic evaluation," *Pattern*

Recognition, vol. 30, no. 8, pp. 1347-1360, 1997.

[9] S. W. Lee and E. S. Kim, "An efficient post-processing algorithm for error correction in Hangul address re-cognition," in *Proceedings of the 4th Conference on Hangeul and Korean Language Information Processing*, 1992, pp. 555-566.

[10] S. Jaeger, S. Manke, J. Reichert, and A. Waibel, "Online handwriting recognition: the NPen++ recognizer," *International Journal of Document Analysis and Recognition*, vol. 3, pp. 169-180, 2001.

[11] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260-269, 1967.

[12] E. Fredkin, "Trie memory," *Communications of the ACM*, vol. 3, no. 9, pp. 490-499, 1960.