Handwritten Address Interpretation System Allowing for Non-use of Postal Codes and Omission of Address Elements

Tatsuo Akiyama, Daisuke Nishiwaki, Eiki Ishidera
Media and Information Research Laboratories, NEC Corporation
1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa, 211-8666, Japan
{t-akiyama@df, d_nishiwaki@da}.jp.nec.com, ishide@ccm.cl.nec.co.jp

Katsuhiko Kondoh, Masato Hayashi, Toshifumi Yamauchi
Social Information Solution Division, NEC Corporation
1-10, Nisshincho, Fuchu, Tokyo, 183-8501, Japan
{k-kondoh@bk, m-hayashi@ax, t-yamauchi@cw}.jp.nec.com

Abstract

This paper reports a new address interpretation system which allows both non-use of postal codes and omission of address elements, such as the omission of county or state when a city name has been given. While lexicon-driven recognizers show good recognition performance when they are input with adequate word images and a lexicon containing correct word strings, it is difficult to design one which would be of practical use when postal codes are not in use and elements of addresses may have been omitted. That is to say, an inadequate design in this area is likely to result in an impractically high erroneous recognition rate. In response to this problem, we propose here an advanced address interpretation system that utilizes both an improved address interpretation method and improved word recognition methods. The improved address interpretation method has been designed to satisfy as completely as possible the need to accommodate non-use of postal codes and the omission of address elements, while the improved word recognition methods have been designed to achieve low erroneous recognition rates in cases in which that need has not been fully satisfied. When we applied our new system to approximately 2000 actual address images for which that need would be relevant, we achieved a 52% *rate of correct outward sorting with only a* 0.7% *rate of erroneous outward sorting. These rates are good enough for practical applications.*

1. Introduction

Handwritten-address interpretation systems have been under development since the early 1990's. In spite of extensive research effort, however, no system is yet able to interpret postal addresses as precisely as a human being, and efforts are still being made to improve performance.

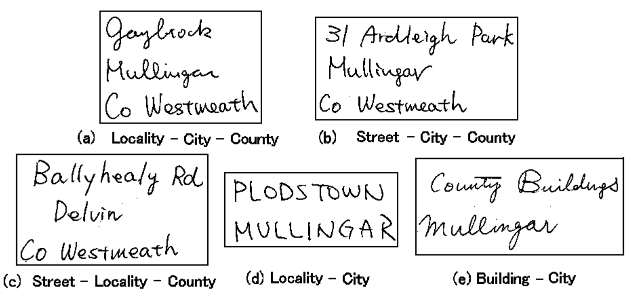


Figure 1. Handwritten Irish Addresses

Current systems are commonly applied to automatic mail sorters, which classify individual mail items according to their destinations. This classification involves two types of sorting: outward sorting, by which a mail item is classified to the main post office nearest its destination, and inward sorting, by which each mail item is classified to the area assigned to an individual delivery person. In outward sorting, interpretation is necessary with respect to postal code, city name, or the name of some other general locality. In inward sorting interpretation needs to be more detailed, so as to include more specific localities and street names.

Most conventional address interpretation systems have been proposed for US address interpretation [5], [1]. In such systems, when a postal code is recognized, the street name lexicon corresponding to that code is called up. When a postal code cannot be fully recognized, additional address information (state and/or city name, etc.) is used in an attempt to recover the full code.

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 **IEEE**

By way of contrast, the system we propose here has been designed for Irish postal addresses, which differ in two important ways from the US system. The most striking difference is that there is no postal code in Irish postal addresses. This means that some address classes, such as county (corresponding to state in the US) names, city names, and other locality names will be required for outward sorting. Another difference is that some address class names are customarily omitted. Figure 1 shows examples of Irish mail addresses; as may be seen, they contain no postal codes. Figure 1(a) is an example in which a county name, city name and locality name have all been written (i.e., "no-omission" with respect to the task of outward sorting). Each of the addresses in Figures 1(b) to 1(e) has one or more omission with respect to outward sorting.

In developing our system, we set out two conditions. The first was that interpretation should be completed in a short time: for example, around two seconds with a Pentium III 1.0 GHz PC. The second condition was an ability to achieve low erroneous sort rates (e.g., 1.0% or less for both outward sorting and inward sorting), with correct sorting rates of at least 50% for both outward sorting and 25% for inward sorting.

One of the first problems we faced was that of the design for a lexicon-driven word recognizer. Such recognizers are generally used in address interpretation systems because of their excellent performance when they are input with appropriate word images and their lexicons contain correct word entries. Use of the lexicon-driven word recognizer is the most successful approach yet developed for handwritten-word recognition. Conventional recognizers could not, however, be expected to perform particularly well with Irish postal addresses because of the lack of postal codes and the omission of address elements.

For this reason, we found it necessary to develop an advanced interpretation system that features both an improved address interpretation method and improved word recognition methods. The improved address interpretation method has been designed to accommodate the omission of an address element by "jumping over" any omission and going directly to the next address class for its interpretation result.[1] The improved word recognition methods employ word verification methods based on multiple classifiers and on multiple segmentation methods.

In the following sections of this paper, for purposes of simplicity, we focus particularly on outward sorting. In Section 2, we discuss problems that had to be overcome. In Section 3, we introduce our improved address interpretation method, and in Section 4 we present improved word recognition methods. In Section 5, we give simulation results and briefly summarize our work.

[1] Addresses of which some elements are customarily omitted may be found in not only Ireland but many other countries. The address interpretation method is applicable also for these countries, with designing what omission to be accommodated depending on circumstances of each country.

2. Analysis and Ideas

2.1. Interpretation for Domestic Outward Sorting

In designing an address interpretation system for outward sorting, we set the following conditions:

1. Addresses would contain no postal code and would consist of three elements (county, city, and locality), of which one or two might have been omitted.
2. The address interpretation system would have a lexicon-driven word recognizer for identifying county, city, and locality.
3. The recognizer must be able (1) to judge recognition-result candidates on the basis of determined confidence values, and (2) to accept or reject for output any given "candidate" on the basis of predetermined parameters.
4. Locality recognition cannot be based simply on the use of a single lexicon containing the names of all localities because the required searches would be too time-consuming.
5. Recognition results will fall into one of two categories: "Sorting" and "Non Sorting".
    - "Sorting" refers to that condition in which either the city name, or the locality name, or both have been recognized.
    - "Non sorting" refers to that condition in which neither the city name nor the locality name has been recognized.
6. Addresses may be assumed to have a tree structure.
7. The performance of the address interpretation system will be evaluated on the basis of its "correct sorting" rate and "erroneous sorting" rate, where
    - "correct sorting" refers to the condition in the sorting category in which no erroneous city or locality name has been accepted for output and at least one correct city or locality name has been accepted.
    - "erroneous sorting" refers to the condition in the sorting category in which either an erroneous city name, an erroneous locality name, or both have been accepted for output.

2.2. Breadth-First Search Approach

One of the first approaches we considered was a breadth-first search of the tree structure. Interpretation with this approach would flow in the following way:

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

Breadth-first Search

Step 1 Using a lexicon that contains entries for all counties, obtain a recognition result for the county name.

Step 2 If the result is rejected, end. Otherwise, using a lexicon that contains entries for all cities in the identified county, obtain a recognition result for the city.

Step 3 If the result is rejected, end. Otherwise, using a lexicon that has entries for all localities in the identified city, obtain a recognition result for the locality, and end.

Figure 2(a) shows combinations of word recognition results and their consequent sorting categories. "C", "R," and "E" in this figure represent correct accepted result, rejected result, and erroneous accepted result, respectively. As this figure illustrates, the Breadth-First strategy would potentially be useful for addresses which contain no omissions.

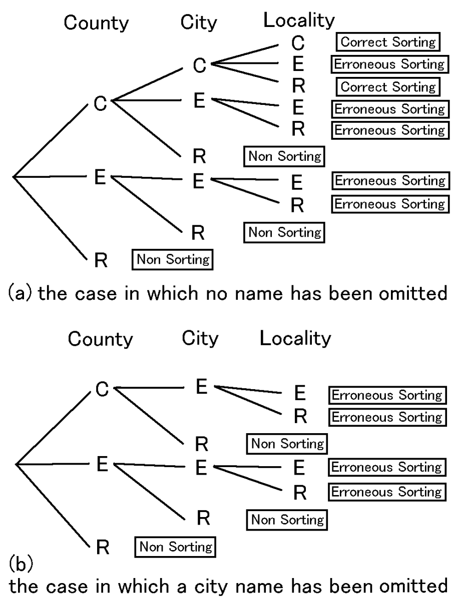


Figure 2. Relationships between word recognition results and sorting results

As may be seen in Figure 2(a), however, in order to achieve in the "no-omission" case the same outward sorting performance as can be obtained with the use of postal code recognition, we would have to improve word recognition performance. That is, with postal code recognition, only a single correct recognition (of the code itself) is required to achieve outward sorting. In contrast to this, for the case illustrated in the figure, two correct accepted results, with no erroneous accepted result, are required. In other words, recognition performance itself would need to be improved in order to match postal code performance levels. Our methods for achieving such improvements are presented in Section 4.

2.3. Handling Address-element Omissions

Our system also needs to handle outward-sorting situations in which one or more address element has been omitted. As an example of this, let us consider the case in which a city name has been omitted. Assuming the Breadth-First approach were simply applied as is to this case, the results would be as shown in Figure 2(b). Specifically, any name appearing in the location assumed to be that of the city name could only either be accepted in error or rejected. In either case, no useful result would be obtained. This could only be avoided if the system were to know in advance of a specific omission, but obviously that would never be possible.

The issue, then, becomes one of how to design a system capable of accommodating such omissions.

3. Address Interpretation

We began by assuming that outward sorting should be made possible for the following five combinations of address elements: (County, City, Locality), (County, City), (County, Locality), (City, Locality), and (City). Note here that (County) would not be specific enough for outward sorting and (Locality) would require the search of a lexicon containing all localities, a prospect that we have previously rejected because of its excess time-consumption.

3.1. Address Relation Model

In order to understand how we have designed our interpretation system, let us consider it in terms of an Address Relation Model (see Figure 3). An ordinary Breadth-First search can be illustrated as shown in Figure 3, with the model's nodes simply being connected by arrows. In our design, shown in Figure 3(b), we are able to jump individual address elements, going directly, for instance, from County to Locality.

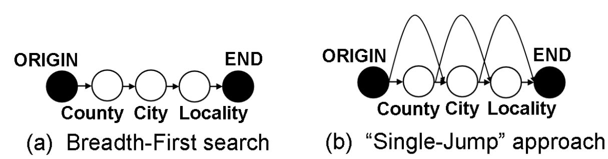
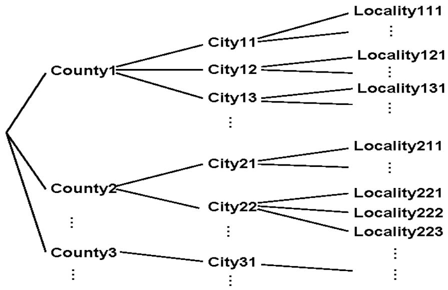


Figure 3. Address Relation Models

3.2. How to make lexicons and address candidates

In order to implement this "single-jump" approach, we have created address candidate tables (illustrated in Figure 4). Let us consider how an address candidate table would be employed in the case in which a city name has been omitted. In the first step, the lexicon of counties is checked for candidates, and two, County1 and County2, are accepted and entered into the candidate table (see Figure 4(b), along



Address Tree

A. C. S. – Address Candidate Score

Result (County)	Score (County)	Result (City)	Score (City)	Result (Locality)	Score (Locality)	A. C. S.

(a)

County Recognition

Order	Recognition Result	Recognition Score
1	County1	200
2	County2	100

Result (County)	Score (County)	Result (City)	Score (City)	Result (Locality)	Score (Locality)	A. C. S.
County1	200					200
County2	100					100

(b)

City Recognition

Order	Recognition Result	Recognition Score
1	City12	300
2	City21	100

Result (County)	Score (County)	Result (City)	Score (City)	Result (Locality)	Score (Locality)	A. C. S.
County1	200					200
County2	100					100
		City12	300			300
		City21	100			100
County1	200	City12	300			500

(c)

Locality Recognition

Order	Recognition Result	Recognition Score
1	Locality131	400

Result (County)	Score (County)	Result (City)	Score (City)	Result (Locality)	Score (Locality)	A. C. S.
County1	200					200
County2	100					100
		City12	300			300
		City21	100			100
County1	200	City12	300			500
County1	200			Locality131	400	600

(d)

Figure 4. Address Candidate Tables

with their respective "Address Candidate Scores" (confidence values).

Next, the all-cities lexicon is checked for matches with the subsequent address element (which we know here to be not actually a city name). As may be seen in the "City Recognition" portion of the figure, two city candidates are accepted, City12 and City21. Referring these results to the Address Tree in the figure, we notice that City12 actually exists in County1, which suggests this combination to be a reliable candidate, and we enter this combination in the table (Figure 4(c)). In our system, this second address element is not, however, assumed to be a city name. We also need to check the possibility of its being a locality name. To do this, we check the lexicon of all-localities for that county which shows the best Address Candidate Score (here, County1). This check reveals the acceptable result of Locality131, and we enter its score in the Table (Figure 4(d)). As may be seen in this final table, the combination County1-Locality131 outscores the combination of County1-City2 by 600 to 500, and it is chosen as the recognition result.

We should note that we assume here a rule under which each address is to be written in order of increasing generality, i.e., locality → city → county, and where results indicate that this rule has been violated, the system automatically determines a "non sorting" decision.

4. Improved Word Recognition

In our address interpretation system, we started with a lexicon-driven word recognizer based on the algorithm introduced in [2], and we enhanced its performance by applying the GLVQ described in [4] with 392-dimensional orientation histogram [2] to the verification of preliminary results (see Figure 5(a)).

4.1. Rule-Base Word Verifications

We also developed word verification methods that employ production rules. Some of the rules are used for a general verification, while others are targeted at address elements that appear especially frequently.

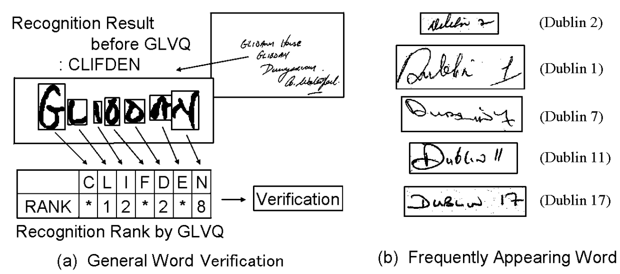


Figure 5. Verification and Frequently Appearing Word

Framework for General Word Verification

A lexicon-driven word recognizer is only properly able to handle character strings that have previously been registered in its lexicon. When an address has been written in a mistaken or incomplete manner, it will either fail to find an acceptable match or it will output an erroneous result. The General Word Verification method that we have developed helps mitigate this problem by applying an added, stricter standard for acceptance. It considers the individual characters in character strings that have been accepted in the original recognition process, and it evaluates such strings on the basis of GLVQ recognition of individual characters, the size of individual characters, and gaps between characters. If a given string does not meet a certain standard with respect to any one of these three areas of evaluation, it will be rejected.

Verification with character sizes is a method by which a result is rejected if the sizes of characters (the size of circumscribed quadrangles) obtained from word recognition are judged to be too large or too small. Verification with character gaps is a method by which a result is rejected if the character gaps are too wide. Both of the above two methods of verification are able to reduce erroneous word recognition caused by connection of primitive segments in one-dimensional Dynamic Programming.

In character-size based verification, the size of a quadrangle circumscribed about a character is judged to be either too large, too small, or of acceptable size. In character-gap verification, gaps between characters in a string are judged to be either too large or of acceptable size. In GLVQ recognition, as may be seen in Figure 5(a), a ranking is assigned to each individual character in a word string. These rankings are then compared to candidate results obtained in the original recognition process. If the first letter in a candidate is found among the five highest ranking characters in the GLVQ results, it is awarded two points. If any of the subsequent characters in that candidate is found among the five highest ranking characters, it is awarded one point. If the average of character-points for the candidate does not exceed a certain pre-determined value, the candidate will be rejected.

High-frequency Word Verification with Target-Oriented Rules

Additionally, in High-frequency Word Verification, we apply target-oriented rules to words which appear frequently. As one example, let us consider the case illustrated for Dublin, an obviously high-frequency word, in Figure 5(b). The Dublin area is divided into districts each of which is designated by a number (1 through 26) or a special designation, 6W. As may be seen in the figure, even when the Dublin portion has been clearly recognized, individual strings may be so similar as to result in a high rate of erroneous recognitions. One approach might be to separate them into main and suffix portions, but the separation process itself (as may be seen in the case of "Dublin 7") would present its own special difficulties. Our solution was, rather, to enhance the verification logic for the specific target of "Dublin + district suffix." To our General Word Verification, we added the application of 400-dimensional MQDF([2]), using as input a 400-dimensional orientation histogram.

4.2. Multiple Segmentation Method

In our character segmentation, we employ two distinct methods, those of Kimura et al. ([2]) and Nishiwaki et al. ([3]). While [2] segments at valley points, [3] bases segmentation on variations in the lengths of individual vertical runs. [2] tends to perform well with cursive script but can sometimes fail when block letters abut each other, a problem that [3] avoids. In our segmentation, we first use [2], and if a result is rejected, we then apply a combined form of [2] and [3].

5. Experimental Results

We experimentally evaluated our approach on the basis of the Address Relation Model illustrated in Figure 6(a) for the distribution of the Address Class Combinations listed in Table 1. Performance levels for the various tasks are shown in Figure 6(b).

Table 1. Distribution of Address Class Combinations

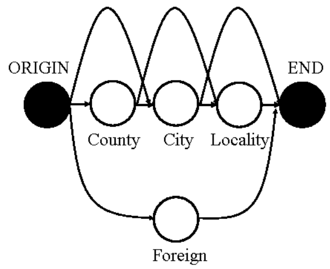
	rate
(County, City, Locality)	22.7%
(County, City)	17.8%
(County, Locality)	20.5%
(City, Locality)	13.7%
(County)	0.1%
(City)	18.3%
(Locality)	4.8%
(Foreign)	2.4%

We created two Address Candidate Tables for each target address, assuming that the address might be foreign as well as domestic. We then compared the Country score of the best Foreign candidate with the County and City scores of the best Domestic candidate. If the Country score was higher than either of the other two, the Foreign candidate was chosen; if lower than either, the Domestic candidate was chosen. In (very rare) case of a tie, neither Foreign nor Domestic would be chosen.

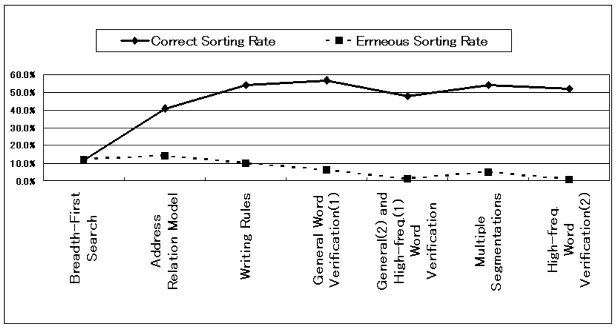
With the Address Interpretation Method being used in combination with the Address Relation Model, the system is able to handle omissions consistent with the model. The system is particularly successful in handling the situation in which a city name the same as a county name is used and the county name is omitted. This might often be the case for the City of Carlow, which is in the County of Carlow. In a conventional interpretation system, if the county name Carlow were omitted and only the city name Carlow were written, that lowest line would be interpreted as county name and would subsequently fail to conduct a search based on the City of Carlow. With our system, as has been previously noted, this problem does not arise.

With respect to Rule-Base Word Verification, we may note the example shown in Figure 5(a). Here, the original recognition result was rejected by GLVQ (in this figure, "*" represents ranks not among the top sixteen) . The decision

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE



(a) Address Relation Model for Outward Sorting



(b) Respective Performance Levels

Figure 6. Experiment

to reject was made because the number of characters with at least a fifth rank was proportionally too low.

In an evaluation of system performance, we used 2088 samples of images collected from actual Irish mail and achieved a 52% rate of correct outward sorting with a 0.7% rate of erroneous outward sorting. Figure 6(b) shows performance for respective tasks. Note that at the beginning of the process, the initial correct and erroneous rates were both 12%.

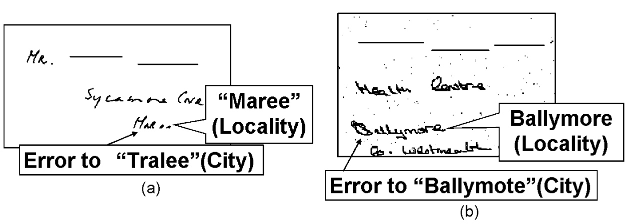


Figure 7. Remaining Errors

With respect to the erroneous results, we may usefully consider the examples shown in Figure 7, which shows examples of remaining errors. One of the main causes was the writing of addresses inconsistent with the Address Relation Model, as shown in Figure 7(a). In this example, neither a county name nor a city name has been written; its lowest line is a locality name. Our system was unable to accommodate this severe of a divergence.

Another cause was words with extremely similar spellings, such as "Ballymote" and "Ballymore," as shown in Figure 7(b). Ballymote is a city name and Ballymore is the name of a locality name in County Westmeath. In such a case, if an acceptable county name could not be determined, the system would have no choice but to look for a city name resembling the next line above, and it would output a very high score for an erroneous "Ballymote." This problem represents a fundamental difficulty that we have yet to overcome.

6. Summary

In this paper, we have proposed an advanced address interpretation system that utilizes both an improved address interpretation method and improved word recognition methods. It is designed to be used on addresses which contain no postal codes and from which certain address elements may have been omitted. The improved address interpretation method helps the system to use the most appropriate lexicon for any given individual word-string search. The improved word recognition methods help to keep erroneous recognition rates low even when the most appropriate lexicon is not always used, and they also help to achieve more accurate character segmentation. Performance evaluations show our system have achieved a 52% rate of correct outward sorting and a 0.7% rate of erroneous outward sorting.

References

[1] A. Filatov and V. Nikitin and A. Volgunin and P. Zelinsky, "The AddressScript™ Recognition System for Handwritten Envelopes", *DAS1998*, pp. 222-236, 1998.

[2] F. Kimura and S. Tsuruoka and Y. Miyake and M. Shridhar, "A Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words", *IEICE Transactions on Information and Systems*, Vol. E77-D, No. 9, pp. 785-793, 1998.

[3] D. Nishiwaki and K. Yamada, "A New Numeral String Recognition Method Using Character Touching Type Verification", *IWFHR-VI*, pp. 567-576, 1998.

[4] A. Sato and K. Yamada, "A formulation of learning vector quantization using a new misclassification measure", *ICPR'98*, Vol. I, pp. 322-325, 1998.

[5] S. Srihari and E. Kuebert, "Integration of Hand-Written Address Interpretation Technology into the United States Postal Service Remote Computer Reader System", *ICDAR'97*, pp. 892-896, 1997.

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 **IEEE**