USE OF EXTERNAL INFORMATION IN ZIP CODE RECOGNITION

by

Jonathan J. Hull and Sargur N. Srihari

Department of Computer Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, New York 14260

ABSTRACT

The recognition of zip codes in a postal address can utilize many sources of external information. City and state names are obvious examples that can be used in conjunction with a city-state-zip directory to provide evidence about digits in a zip code. This paper describes an extension of this methodology that uses information about legal *street names* and *suffixes* to constrain the digits in a zip code. The design of such a technique is presented and some preliminary experiments with the ZIP+4 database are discussed. A method for extracting useful information from an address is discussed that does not require complete recognition of all the characters in words. Rather, a feature description is computed and used to retrieve a set of zip codes from a dictionary that correspond to words with the same description. A statistical simulation explores the effect of several alternative feature sets. It is shown that even a relatively simple description of two words in the street line of an address can significantly reduce the number of zip codes that could appear on a piece of mail.

1. Introduction

Reading the zip code within the destination address area of a mailpiece is of central importance in automated mail sorting. The problem is not always amenable to straightforward alphanumeric character recognition — particularly when the address is handwritten, or composed of unconventional fonts, or the print is of poor quality, etc. Fortunately, the zip code in a destination address does not usually occur in isolation. It occurs in the context of a city name, a state, a street address, the name of a destination, and perhaps an attention line. Also, there is usually a return address and sometimes advertising material on many mailpieces. All this information has some potential to contribute to recognition accuracy. The most obvious use of such *external* information is for the city and state to confirm a five digit zip code. State information constrains the first digit whereas city information constrains the second and possibly the third digits. Thus if the city and state information is known, the number of alternatives is reduced for zip code recognition. There exist multi-line reading equipment today that are capable of using city and state information as well as street information in determining a nine-digit zip code.

Besides the information in the city/state/zip line (or lines), there is a wealth of information in the street line that could be utilized. Figure 1 illustrates several sources of external information that could be used to constrain the digits in a zip code. For example, if mail for a particular city is being sorted, recognizing the pre-directional code as *N* might limit the zip code to one that occurs north of some known boundary. A better use of external information would be to recognize the post-directional code (e.g., NW, NE, etc.) when sorting mail for cities that use such codes (e.g., Washington, D.C.). This could dramatically reduce the number of zip codes that could match a mail piece. Recognizing the street name or the organization would have a similar effect. If combined with a suitably arranged dictionary, this would in many instances specify the zip code.

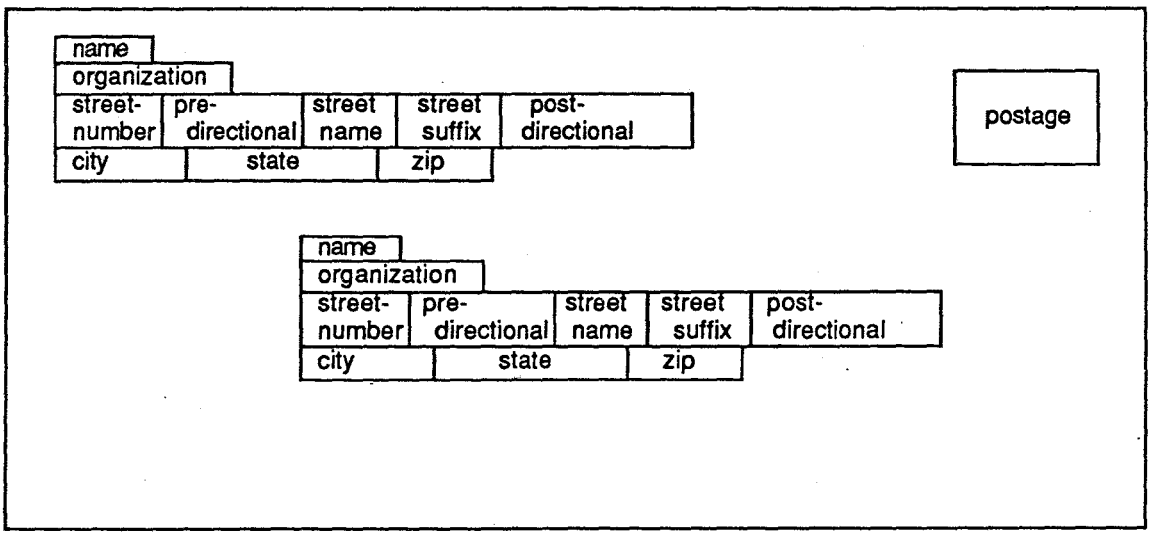

Figure 1. Template for the face of a mailpiece.

This paper concentrates on a technique of using external information to constrain the digits in a five-digit zip code. Usually, an accurate recognition of all the characters in a word, such as the city name, is required to use external information. There is usually little tolerance for broken, touching, or smeared characters. Although this insures an accurate recognition procedure, it does not utilize all the information that is present in an address. Instead, a technique is desired that is robust in the presence of noise and can extract some useful information from textual portions of a destination address. Such a technique should be able to constrain the digits of a postal code even if it cannot fully recognize the text.

It is proposed that a technique be used that computes a noise-tolerant feature description of a specific word or words in an address. This feature description is then used to access a dictionary and return a number of zip codes that correspond to words with that feature description. This either produces a unique identification of the zip code, or, by constraining the digits of the zip code, provides information that could be used to advantage in recognition. This methodology has the advantage that it does not require an exact recognition of all the characters in the words it examines. Only some features have to be calculated and these features may provide only a gross description of the word. However, such a feature description may provide useful information even though it is tolerant to noise and easy to compute.

The remainder of this paper examines this methodology for using external information to constrain the digits in a zip code. The technique is defined precisely and a detailed example of its operation is given. Several statistics are then defined that project its performance in a mail sorting environment. These statistics are then used to simulate the performance of this methodology for a city with twenty six unique five-digit zip codes. The performance of this simulation is discussed and future improvements are mentioned.

2. Method for Constraint Generation

A method is presented for constraining the digits in a zip code. The digital images of isolated words in an address are input. It is assumed that the words come from known categories such as the first words in a street name. A feature description of an input word serves as an index into a dictionary that returns one or more zip codes associated with the input word. This set of zip codes is referred to as the *neighborhood* of the words with that feature description. The objective of this procedure is to find as small a neighborhood as possible. Ideally, a simple feature extraction routine could be used to find a neighborhood that contains a single zip code.

An example of this procedure is presented in Figure 2. Figure 2(a) shows the twenty six five-digit zip codes that could appear on a piece of mail with the city and state names of "Wilmington, Delaware" on it. Thus the city and state information constrain the number of possibilities to twenty six from a theoretical maximum of 100,000 or the practical maximum of about 41,000 (about this many zip codes are in use). The map in Figure 3 shows the geographic distribution of ten of the twenty six codes. The remaining sixteen are assigned to unique businesses or to Post Office boxes. Under the assumption that a mail stream is being sorted for the city of Wilmington, it is not possible to route a piece of mail to its correct five-digit zip code without reading information in some area of the address besides the city/state/zip line. One possibility is to use the number of characters as a feature description for the second word in the line above the city/state/zip line. Under the assumption that this word is either a pre-directional code (N,S,E,W,NW,NE,SW,SE) or the first word in a street name, a suitably organized dictionary could be used to determine the zip codes associated with all the

19801 - 19810, 19850, 19885 - 19899
(a)

Mr. John Q. Public
376 Van Buren Street
Wilmington, Delaware 19802
(b)

feature	zip codes	ns
1	19801-10, 19893, 19895	12
2	19807-10, 19850, 19887	6
3	19801-10, 19850, 19885-9, 19896-7, 19899	19
4	19801-10	10
5	19801-10, 19888, 19892, 19893	13
6	19801-10, 19890, 19898	12
7	19801-10, 19889, 19896, 19897	13
8	19801-10, 19891, 19892, 19894	13
9	19801-10	10
10	19801-10, 19890, 19891	12
11	19804, 19807, 19808, 19810	4
12	19804, 19806-10	6
13	19805, 19807	2
15	19850	1

(c)

feature1, feature2	zip codes	ns
2, 6	19809	1
3, 2	19802-3, 19805-10	8
3, 3	19801, 19803-5, 19808, 19810	6
3, 4	19803, 19805-10, 19850, 19885-9, 19897	14
3, 5	19802-5, 19807-10	8
3, 6	19801-2, 19804-5, 19807-9, 19896, 19899	9
3, 7	19803, 19810	2
3, 9	19808	1
4, 2	19801-10	10

(d)

Figure 2. Example of constraining zip codes using feature descriptions: (a) the five-digit zip codes for Wilmington, Delaware, (b) an example address, (c) the dictionary for one feature: length of the first word in the street name. (d) an alternate dictionary that uses the length of the first word and the length of the street suffix as its features.

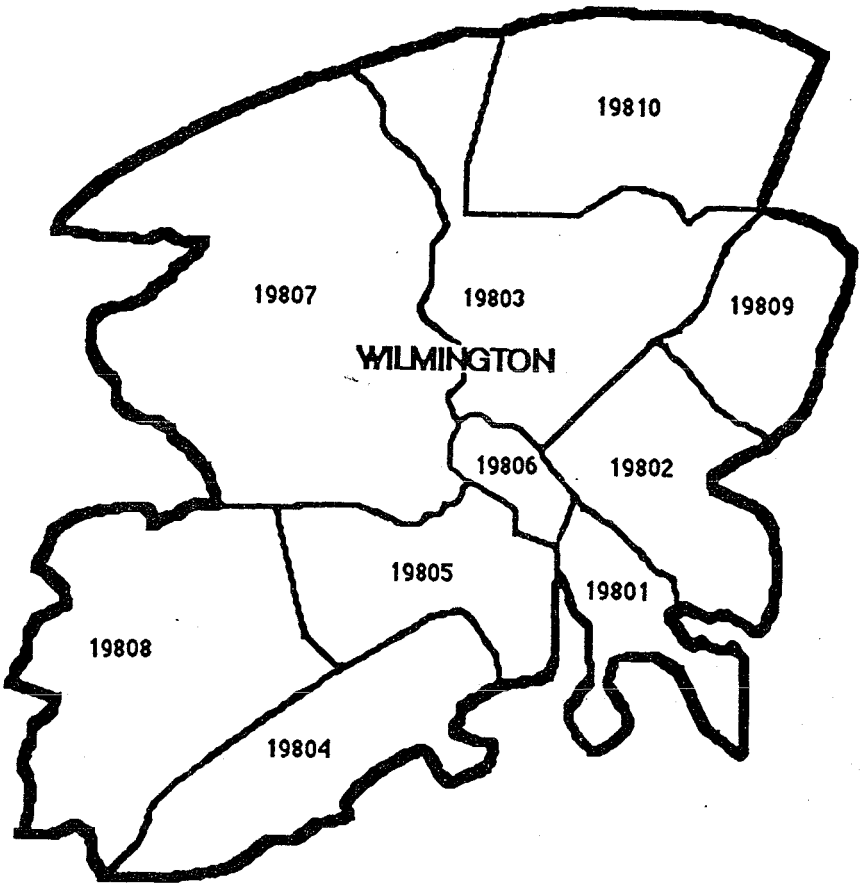

Figure 3. Zip code map of Wilmington, Delaware.

words having that feature description. Such a dictionary structure is shown in Figure 2(c). The dictionary is indexed by the number of characters in a word. The neighborhoods of zip codes are shown and the size of each neighborhood (ns) is indicated. Using this dictionary, the example address in Figure 2(b), and the assumption that the number of characters in the first word of the street address, *Van*, is correctly determined, the number of zip codes that could match the address is reduced from twenty six (Figure 2(a)) to nineteen.

An improvement of the procedure presented in this example uses a feature description of more than one word to index into the dictionary. An example of a dictionary for this modification is shown in Figure 2(d). The dictionary is indexed by the number of characters in the second and last words in the line above the city/state/zip line. It is assumed that this is the street address line. The words in question are assumed to be the first word of the street name and the street suffix (Street, Avenue, Lane, Boulevard, etc.), or, if it exists, the post-directional code (N, S, NW, etc.). The example of Figure 2 shows that if the number of characters in *Van* and *Street* can be correctly computed and if these words are the first word of a street name and the street suffix, then the number of five-digit zip codes that could be in the address is reduced to nine. This is a significant improvement over the nineteen zip codes that

could be present if just the first word in the street name was considered.

3. Measures of Performance

There are several statistical measures of performance for this methodology. They all focus on the number of zip codes associated with an address. However, the statistics are differentiated by the characteristic they measure. Each statistic gives an idea of how a different aspect of the approach will perform in practice. The Average Neighborhood Size (ANS) is one such statistic. It is defined as:

$$ANS = \frac{1}{N_f} \sum_{i=1}^{N_f} ns_i$$

where N_f is the number of different feature descriptions present in the dictionary, and ns_i is the size of the i^{th} neighborhood. That is, the number of different zip codes associated with the i^{th} feature description. In the example of Figure 2(c), $N_f = 14$ and $ANS = 9.5$.

ANS is a static measure of performance. It gives the expected number of zip codes in a partition of the dictionary. It is best if this statistic is as close to 1.0 as possible. Then the expected number of zip codes associated with any feature description will be low enough to provide nearly perfect recognition. No other features would have to be calculated. However, a disadvantage of this statistic is that it does not measure any dynamic characteristics of the approach. It is possible that even though a low value of ANS is obtained, there could exist a small number of large neighborhoods. If there are many pieces of mail that contain words with the feature description corresponding to that neighborhood, then there will be many cases where the number of zip codes matched to a mailpiece is much larger than that indicated by ANS. Therefore, a statistic is needed that reflects such dynamic characteristics of the procedure.

The Average Neighborhood Size per Text word (ANS_t) is such a statistic. It is defined as:

$$ANS_t = \frac{\sum_{i=1}^{N_f} ns_i * n_i}{N_t} ;$$

where n_i is the frequency of the i^{th} feature description in the source text and $N_t = \sum_{i=1}^{N_f} n_i$. ANS_t provides a way of incorporating information about the frequency of words in a mail stream into a model of performance for the technique presented here. ANS_t provides a dynamic measure of performance that indicates the expected number of zip codes associated with a piece of mail in a modeled mail stream.

Another statistic ($\%uniq$) provides the percentage of words in the source text with a feature description that uniquely specifies a zip code:

$$\%uniq = 100 * \frac{N_{uniq}}{N_t}$$

where N_{uniq} is the number of words in the text that uniquely specify a zip code.

The three statistics presented here measure the performance of the proposed method of constraining zip codes. A similar analysis was used in another environment [1,2]. ANS is a static measure that can be computed from a given feature description and dictionary. ANS_t and $\%uniq$ require a representation for a mail stream that includes the frequency of occurrence of the target words and the zip codes associated with those words. In the example of section 2, this would be the frequency of every word that could be the first word of a street name as well as the frequency of every combination of the first and last words.

4. Statistical Simulation

A statistical simulation of the technique proposed in this paper for constraining the digits in zip codes was constructed to test the effect of several alternative feature sets. The objective of the simulation was to show that many different feature sets could be used to describe the words in an address. Each feature set has different characteristics in terms of computability and tolerance to noise. The feature set used in practice would depend on these considerations as well as how tightly constrained the zip code should be. The computability and noise tolerance factors are very implementation-dependent. However, the constraint factors can be estimated by the statistics defined earlier.

The database for the simulation was a subset of the ZIP+4 National Directory file for the State of Delaware. The ZIP+4 file contains all the information needed to assign a 9-digit zip code to any address in the United States [3]. This is done by storing, among other information, the ranges of numbers on every street that correspond to each 9-digit zip code. A representation for a portion of the database is shown below:

range	Odd/Even	Complete Street Name	9-digit zip
101-199	O	N French Street	19801-2505
100-198	E	N French Street	19801-2506
201-299	O	N French Street	19801-2507
200-298	E	N French Street	19801-2508

The complete street name can contain up to four fields (explained in section 2): pre-directional, street name, street suffix, and post-directional. The Odd/Even field usually indicates a specific side of a street.

It was desired to test the proposed methodology on a subset of the database where the usual strategy of using external information to constrain the digits in a zip code by examining the city and state name would not work. This is a subset that has the same city and state names. Therefore, the largest such subset in the given database was chosen. This yielded a file of all the street names and five-digit zip codes in the City of Wilmington. As explained in the example in section 2, there were twenty six unique five-digit zip codes in this database.

The Wilmington database was pre-processed so that it included the full spelling for the street suffixes as well as its usual USPS abbreviation. This was done by substitutions such as BOULEVARD for BLVD, LANE for LN, CIRCLE for CIR, STREET for STR, etc. In many cases this yielded two records for every one input: the original and a copy with the substitution for the suffix. There were 2316 records in the original database and 4622 in the preprocessed version. The preprocessed database should represent a large percentage of the ways the names of

streets would be written for destinations in Wilmington.

Several experiments were conducted for different feature descriptions. Two constraints were used in each experiment: either the first or the first and last words from the complete street name. The first word was the pre-directional, if it existed, otherwise it was the first word in the street name. The second word was either the post-directional, the street suffix, or the last word in the street name, chosen in this order. It was assumed that these words could be located in an address by finding the second word and the last word in the line immediately above the city/state/zip line. Presumably the first word in the line is the number of the street address. In the example address of Figure 2(b), the first and last words in the complete street name are *Van* and *Street*.

Dictionaries similar to those shown in Figure 2(c) and (d) were constructed for several different feature sets. The ability of these feature sets to determine which of the twenty six possible five-digit zip codes matched addresses in the database was simulated by computing ANS, ANS_1, and $\%uniq$. The results for several different feature sets are given below.

4.1. Length

The number of characters in the two words were used as features. This is the same feature set used in the example discussed in section 2. The results using this feature set were:

constraint	N_f	ANS	$\%uniq$	ANS_1
first word only	14	9.5	0	12.3
first and last words	84	6.4	1	9.8

4.2. Length and First Letter

The number of characters in both words and the exact identity of the first character in each word were used as features. The results were:

constraint	N_f	ANS	$\%uniq$	ANS_1
first word only	202	4.8	2	7.3
first and last words	1758	2.1	22	3.2

4.3. First and Last Letters

The identities of the first and last letters of both words were used as features. No information about the number of characters was used. The results were:

constraint	N_f	ANS	$\%uniq$	ANS_1
first word only	412	2.8	14	5.7
first and last words	2256	1.7	45	9.0

4.4. First and Last Letters and Length

The identities of the first and last letters in both words and the number of characters in both words were used as features. The results were:

constraint	N_f	ANS	$\%uniq$	ANS_t
first word only	545	2.3	23	5.1
first and last words	2472	1.6	51	8.9

The results of these experiments illustrate several aspects of this approach. A very simple feature of just the number of characters in two words produced an ANS_t value of about 10. This is much better than the value of twenty six that would be encountered if no constraints were used. If only one letter and the number of characters in one word can be recognized, ANS_t is reduced to 7.3. If the first character and the lengths of two words can be computed, a value of 3.2 is obtained for ANS_t. When the first and last characters in two words can be recognized, the ANS value is only 1.7. Also, the zip code for up to 45% of the mail stream is determined with no other feature tests. This is encouraging but comes at the cost of an increase in ANS_t to 9.0. A similar trend occurred with the fourth feature set where ANS_t was about the same at 8.9, but $\%uniq$ increased to 51%. If only the first word in the street address is considered, ANS_t is only 5.1, but a zip code is uniquely determined for only 23% of the pieces.

5. Discussion and Conclusions

A method for using external information in an address, other than the city and state names, to constrain the digits in a five-digit zip code was discussed. This method requires that a small number of features be calculated for a few pre-determined words. These feature descriptions are then used to access a dictionary and return a set of zip codes that could be present in the image. It is best if this set is as small as possible. Three statistics were defined that measure the projected performance of this technique.

A statistical study was conducted in which these statistics were computed over a database that represented the city of Wilmington, Delaware. Two constraints (the feature description of one or two words) and four feature descriptions were tested. It was shown that the best overall performance, as measured by the ANS_t statistic, was achieved for the two-word constraint and a simple feature description of the identities of the first letters and the lengths of the words. An average of only 3.2 zip codes matched each address. This is quite an improvement over the twenty six zip codes eligible if no constraints were used.

Future work in this area should include an improved simulation of the mail handling environment. The population of addresses used for the experiments discussed here contained at most two entries for each street in each five-digit zip code zone. The simulation thus assumes an equal amount of mail is destined for each street. An improvement would incorporate data that more accurately reflects the amount of mail going to each street.

Acknowledgements

The authors gratefully acknowledge the assistance of Glen Davis of Arthur D. Little, Inc. in acquiring several of the databases used for experiments discussed in this paper. We also acknowledge the support of the United States Postal Service under BOA contract 104230-86-M3990.

References

1. J. J. Hull, "Word shape analysis in a knowledge-based system for reading text," *The Second IEEE Conference on Artificial Intelligence Applications*, Miami Beach, Florida, December 11-13, 1985, 114-119.
2. J. J. Hull, "Hypothesis generation in a computational model for visual word recognition," *IEEE Expert*, August, 1986, 63-70.
3. *Zip+4 national directory technical guide*, United States Postal Service Address Information Center, September 1, 1985.