Firm Name Recognition for Automatic Address Interpretation

Ping-Kit Wong, Tin Kam Ho, Sargur N. Srihari

Center of Excellence for Document Analysis and Recognition
State University of New York at Buffalo
226 Bell Hall
Buffalo, New York 14260

email: pwong@cs.buffalo.edu

Abstract

Firm name recognition provides a useful source of information for automatic postal address interpretation. This paper presents two approaches to firm name recognition. The word-based approach treats a firm name as a list of words each providing an index to the database. The character-based approach treats a firm name as a sequence of characters and matches them using a similarity measure. Contextual information at different levels is utilized by the two methods. Both methods were implemented and tested using a firm name database containing over one million records and images scanned on a postal OCR. Preliminary experimental results are promising and they reveal problems that need to be addressed in further development of the methods.

1 Introduction

Firm name recognition can be a very useful source of information for automatic postal address interpretation. A recognized firm name with a ZIP+4 Code retrieved from the firm name database can be used to select among several choices of ZIP+4 Codes obtained in a contextual analysis of an address block [1]. It can also be used to derive the ZIP+4 Code independently, thereby avoiding errors in interpreting the street address.

A firm name typically contains more than one words separated by white spaces. Besides the difficulties in processing and recognizing the images, variations in the forms of a firm name create special problems that need to be solved. Some of the more common name variations include:

- misspellings and alternative spellings — MONY CENTRES as MONEY CENTERS
- keyword omissions and insertions — JC PENNY as JC PENNEY COMPANY INC

- abbreviations — AMERICAN INS CO as AMERICAN INSURANCE COMPANY
- acronyms — IBM as INTERNATIONAL BUSINESS MACHINES
- aliases — DONALD INS AGENCY as DONALD LIFE INS

In this paper, we present two approaches to firm name recognition using an optical character reader (OCR). We assumed that a firm name has been located by a global contextual analysis on an address block image. The firm name is first segmented into individual word images, which are further segmented into character images. Each of the character images is input to an OCR and a list of decisions is returned. The two approaches differ in the way that contextual post-processing is applied to these character decisions. The first approach is based on processing the words as individual units and then using the decisions on the words as indices into the database. The second approach skips the word recognition stage by concatenating the character decisions for all the words in a name and matching them as a single sequence. Both methods output a ranked list of candidate firm names to be used by higher level contextual analysis.

In Section 2 we describe the organization of the firm name database. The two matching methods are presented in Sections 3 and 4. Preliminary experimental results on a set of postal images are discussed in Section 5. With reference to these results, the two methods are compared in Section 6.

2 The Firm Name Database

Our methods assume that a database of all valid firm names in a specific area is available. An example database is used which covers firm names in an area of 25 Sectional Center Facilities (SCFs) and contains more than 1.2 million records. To reduce the scope of matching for accuracy and efficiency, the database is partitioned first by the SCFs and then by the 5-digit ZIP area where the firm is located. Each firm record is composed of a record number, a firm name field, a street property field, and a 4-digit add-on ZIP Code, if exists.

A partition of the firm name database is activated at run time by a set of hypotheses of 5-digit ZIP Codes. The hypotheses are given by a preliminary analysis of the address block. It is observed that the last two digits of the ZIP Codes on the address block do not often agree with those from the database. Therefore only the first three digits of the hypothesized ZIP Codes, which determine an SCF area, are used to constrain the match. Only the first choice of SCF is used if more than one

are hypothesized. The database partition corresponding to this SCF is activated. Figure 1 shows an overview of our firm name recognition algorithm. *

An additional data structure is built from the firm name database for use in the word-based matching algorithm. The data structure consists of (1) a lexicon including all words that appear in any valid firm name in the partition; and (2) for each entry in the lexicon, a list of record numbers of the firm names containing that particular word. This data structure allows retrieval of a firm name by any of the words that constitute the name.

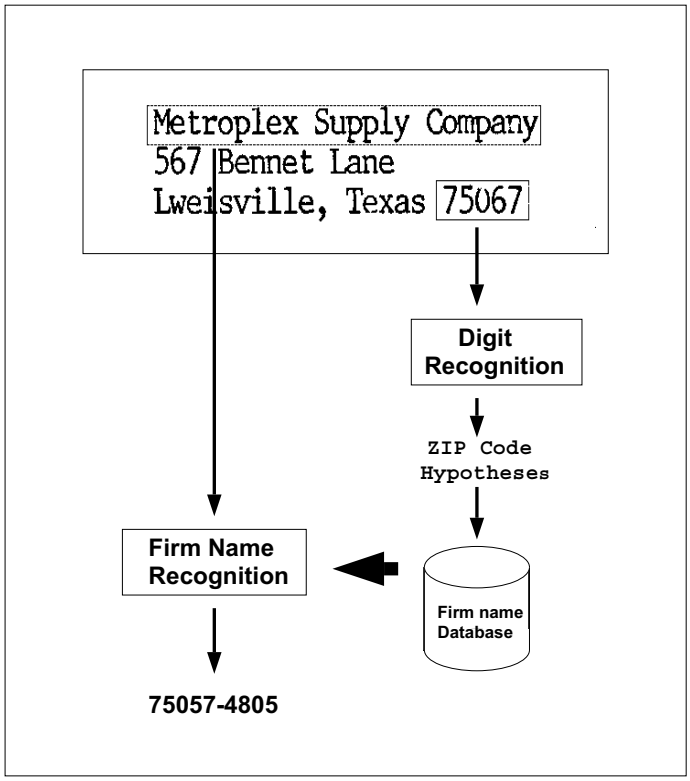


Figure 1: Firm name recognition overview.

*Portions of addresses in this paper have been blocked out to preserve confidentiality.

3 The Word-Based Approach

The word-based approach focuses on the words as individual units in processing the images. Word segmentation is essential in this approach. Each of the segmented word images is input to a word recognition algorithm that determines a ranked list of candidate words from the lexicon. The list is referred to as a *neighborhood* of the target word. The relevant firm names are then retrieved using the indices associated with these candidate words. A score is used to rank the retrieved firm names. Figure 2 shows an overview of the word-based approach.

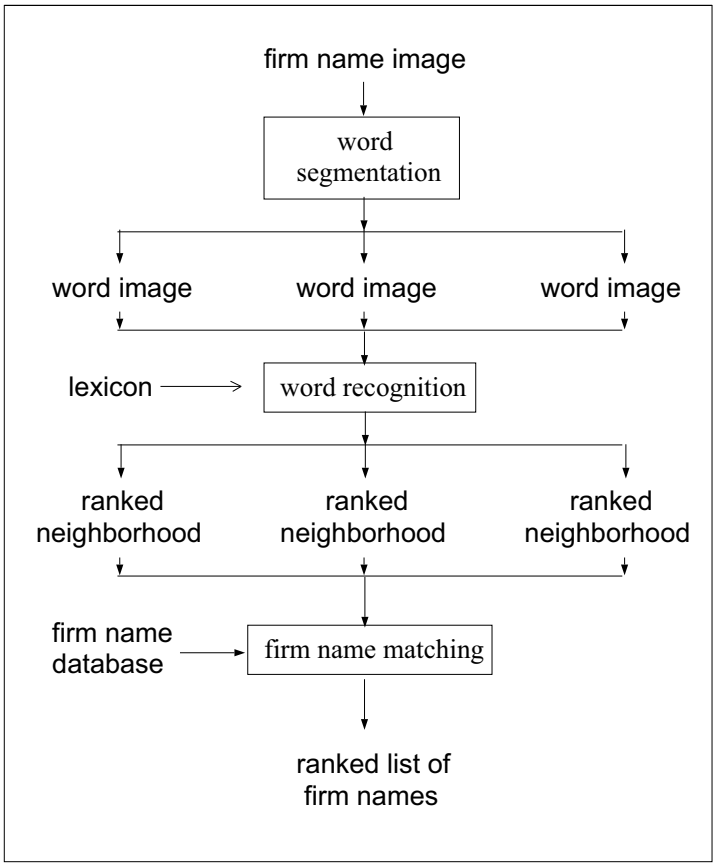


Figure 2: A firm name is matched at the word level in the word-based approach.

The retrieved firm names are ranked by a score based on the ranks of the words in their neighborhoods. A multiple-word firm name may be retrieved more

than once by the candidates for each word in the name. In such cases the score for the name is accumulated. Obviously if several words in the same name are ranked high in their neighborhoods, that name will be ranked high in the output list.

Though the recognition rankings of the words are the primary basis for the scores, other factors can be included into consideration in computing the score. Examples of such factors are the semantic significance of the words and the order of the words in the name. We now describe how the ranked neighborhoods for each word are computed and the adjustment of the scores according to the semantic significance of the words.

Rank Assignment by Word Recognition

The word recognition algorithm takes in a word image and a lexicon. Based on an analysis of the word image, it determines a set of candidate words from the lexicon which best match the image. Word recognition techniques fall into three major categories: character recognition and contextual postprocessing, segmentation-based methods, and word shape analysis. A combination of these techniques can be applied to achieve the best results [3].

In this implementation, we use character recognition and contextual postprocessing as an example word recognition technique. Characters are extracted from the word image and recognized by a polynomial discriminant based classifier [6]. Top five choices for each character along with their confidence scores are taken from the decisions of the classifier. Words from the lexicon that are of length similar (within plus and minus one) to the length of the string of the segmented characters are matched to these decisions. Each word receives the sum of the recognition confidences of its characters as its confidence score. For the words that are not of the same length of the character strings, one character is removed at a time from the longer string, and the match with the maximum score is retained. The candidate words are then ranked by these confidence scores.

Keywords, Aliases and Abbreviations

Many firm names contain common suffixes such as "Company" and "Inc." As a result, when these words are present in an image, they often retrieve a large number of records with relatively little contribution to their discrimination. We would not want the scores of the names be dominated by the successful recognition of the less important suffix instead of the more important parts of the name. Moreover, very often the patrons would confuse the suffixes with one another. However, omitting

these keywords is not a good solution to the problem since there are cases, especially for shorter names, where these keywords do help in discriminating between the names.

To cope with this problem, a small database of unimportant keywords with their aliases is compiled. If a recognized word is identified as a keyword, the name records associated with it are assigned a special low score. The special scores for these keywords can be determined by their frequencies of occurrences in the database or by experimentation. For instance, in our implementation, we found it satisfactory to set this special score to 5, as compared to the maximum score 20 used for other words in neighborhoods of size 20.

Aliases, abbreviations, and alternate spellings are also common in firm names. Aliases like "*Nursing Home*" as "*Caring Center*", and abbreviations like "*13TH*" as "*Thirteenth*" are unlikely to be matched by simple approximate string matching algorithms. To solve this problem, special alias files are created to store common aliases, abbreviations, and acronyms. A recognized word is first checked against those alias files to see if it is a varied form of one of the canonical words. If it is, the firm records indexed by the canonical word would get the score assigned to the recognized word according to its rank in the neighborhood.

Recognition Example Using the Word-Based Approach

The word-based approach to firm name recognition is illustrated by an example shown in Figure 3. Table 1 shows the 20-word neighborhoods produced by the

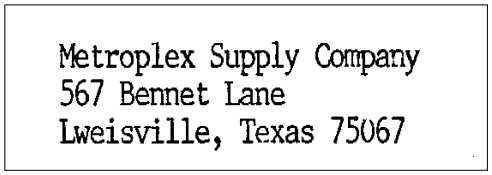


Figure 3: An example address block image containing a firm name.

word recognition algorithm. Scores are assigned to the firm records retrieved by the words in these neighborhoods according to the ranks of the words by descending joint confidences. More than one records may be retrieved by a single word, and in such cases they all share the same score. For instances, all the records retrieved

by "Metroplex" get 20 as their score, and all the records retrieved by "Supply" get 20 as their score. The records retrieved by "COMPANY" get a special low score 5, since "COMPANY" is identified as an unimportant keyword. Accordingly, a firm name record which contains "Metroplex", "Supply", and "Company" scores 45 (20+20+5).

Table 1: Word recognition results and scores assigned to corresponding firm records.

Rank	Word 1	Joint Conf	Record Score	Word 2	Joint Conf	Record Score	Word 3	Joint Conf	Record Score
1	Metroplex	0.741	20	Supply	0.634	20	Company	0.856	*5*
2	Metrolex	0.643	19	Suply	0.529	19	Companys	0.755	*5*
3	Metrotek	0.643	19	Supper	0.508	18	Compact	0.673	18
4	Metronet	0.559	17	Supplie	0.507	17	Compante	0.666	17
5	Metrodeck	0.549	16	Snappy	0.503	16	Command	0.662	16
6	Metroneon	0.533	15	Spain	0.472	15	Compass	0.644	15
7	Tytronix	0.521	14	Surety	0.467	14	Compaq	0.635	14
8	Metrostar	0.521	14	Simply	0.467	14	Tympany	0.629	13
9	Margolies	0.514	12	Yuppie	0.458	12	Botany	0.614	12
10	Macrotek	0.511	11	Septic	0.456	11	Compuad	0.566	11
11	Methodist	0.508	10	Snooty	0.454	10	Conway	0.561	10
12	Getronix	0.503	9	Superm	0.447	9	Complex	0.556	9
13	Cytronics	0.497	8	Sneaky	0.443	8	Comptax	0.556	9
14	Metrocrest	0.497	8	Sundry	0.442	7	Comfac	0.542	7
15	Autoplex	0.496	6	Poppin	0.437	6	Coupac	0.541	6
16	Petrospec	0.492	5	Split	0.427	5	Comcast	0.541	6
17	Microelec	0.491	4	Snipper	0.426	4	Commonw	0.540	4
18	Mccroskey	0.491	4	Slipper	0.420	3	Commons	0.540	4
19	Metrocare	0.490	2	Snipes	0.419	2	Romano	0.539	2
20	Metrology	0.490	2	Snipps	0.419	2	Womans	0.539	2

After every word in the neighborhood is matched with the database index, retrieved firm records are ranked by their total scores and output in that order. Table 2 shows the top 20 firm name records output by the word-based algorithm. The deepest level of sort (LOS) "75057–4805" for this address is successfully obtained at the top choice of the neighborhood.

4 The Character-Based Approach

The character-based approach treats a firm name as a sequence of characters. A sequence of character decisions is matched to the character sequences of the firm

Table 2: Top 20 firm name records retrieved by the word-based algorithm.

Rank	Score	5-digit ZIP	Add-on	Firm name
1	45	**75057**	**4805**	METROPLEX SUPPLY CO.
2	40	75051	1849	METROPLEX AUTO SUPPLY
3	40	75051	5308	METROPLEX AUTO SUPPLY
4	25	75090	5983	SHIPP MOTOR SUPPLY CO.
5	20	75090	5826	TEXTILE SUPPLY CO.
6	20	75090	7020	JOHNSON BURKS SUPPLY CO.
7	20	75081	6007	BILLS OFFICE SUPPLY CO.
8	20	75006	4605	AUDETTE CAD & SUPPLY CO.
9	20	75038	3511	PROTEIN SUPPLY CO.
10	20	75057	4825	POSCO SUPPLY CO.
11	20	75081	5334	UNIVERSAL SHELLAC & SUPPLY CO.
12	20	75038	5318	HOLST SUPPLY CO.
13	20	75057	4825	SUMMIT DRYWALL SUPPLY CO.
14	20	75006	5517	VINSON SUPPLY CO.
15	20	75060	3036	IRVING AUTO SUPPLY CO.
16	20	75051	3540	ACCURATE SUPPLY CO.
17	20	75081	4014	MEDICAL AIR SUPPLY CO.
18	20	75060	3148	ENGLISH COLOR & SUPPLY CO.
19	20	75081	4001	TRANS TEX SUPPLY CO.
20	20	75051	3540	HARPUR SUPPLY CO.

names in the database. The similarity measure is the length of the longest common subsequence between two character strings normalized by the average length of the two strings. The length of the longest common subsequence is computed by a dynamic programming method [7]. Finding the longest common subsequence is a special case of computing the edit distance between two character strings, which is defined as the minimum cost of a sequence of editing operations (insertion, deletion, substitution, ..., etc.) that transforms one string into the other. Edit distance is a useful measure of string similarity for approximate string matching [2]. However, it is computationally expensive and therefore we use a simpler measure instead.

Let A and B be two character strings. A is a subsequence of length n of B if there exist $r_1, r_2, ..., r_n$ $(1 \leq r_1 < \ldots < r_n \leq |B|)$ such that $A_i = B_{r_i}$ $(1 \leq i \leq n = |A|)$. A is a longest common subsequence of U and V if A is a subsequence of both U and V and there is no subsequence of both U and V that is longer than A [7].

Let $lcs(S1, S2)$ be the length of the longest common subsequence of two character strings $S1$ and $S2$. We can define the *similarity* between $S1$ and $S2$ as

$$similarity(S1, S2) = \frac{lcs(S1, S2)}{average_length_of(S1, S2)}$$

For example,

lcs ("word", "word") = 4
lcs ("word", "od") = 2
lcs ("word", "image") = 0

and

$similarity$ ("word", "word") = 1.0
$similarity$ ("word", "od") = 0.67
$similarity$ ("word", "image") = 0.0

The character-based matching algorithm begins by extracting the characters from a name image. A classifier is then applied to the character images and the top decision for each character is obtained. A sequence is formed using these top decisions and is matched to each firm name in a relevant partition of the record database. The records are ranked by their similarity to the input sequence according to the above defined measure.

In our implementation we use a polynomial discriminant based classifier [6] as an example. White spaces and special characters are ignored in matching. The matching is insensitive to characters of different cases. Using the same address block image in Figure 3 as an example, the top decisions given by the classifier are shown in Table 3.

Table 3: Example Character Recognition Decisions.

True word in image	Top choices by classifier converted to upper case
Metroplex	MDTROPLEX
Supply	SNPPLY
Company	COMPANY

By concatenating all these top choices, a character sequence for this image is obtained: "MDTROPLEXSNPPLYCOMPANY" (S1). The similarity calculation is illustrated below taking the firm name "METROPLEX SUPPLY CO" as an example. The concatenation of all the characters in that name gives the sequence

"METROPLEXSUPPLYCO" (S2). The longest common subsequence of $S1$ and $S2$ is "MTROPLEXSPPLYCO". Thus,

$$lcs(S1, S2) = 15$$

Since $length(S1) = 22$ and $length(S2) = 17$, we have

$$similarity(S1, S2) = \frac{15}{19.5} = 0.75$$

The ranking of the 20 best matched names for this example is shown in Table 4.

Table 4: Top 20 firm name records retrieved by the character-based algorithm.

Rank	Similarity	5-digit ZIP	Add-on	Firm name
1	0.75	**75057**	**4805**	METROPLEX SUPPLY CO
2	0.63	75051	1849	METROPLEX AUTO SUPPLY
3	0.63	75051	5308	METROPLEX AUTO SUPPLY
4	0.63	75060	4915	METROPLEX INSPECTION
5	0.62	75061	6355	METROPLEX SALES CO.
6	0.61	75087	4248	CAMERON COMPANY
7	0.61	75061	7193	METROPLEX INSURANCE AGENCY
8	0.60	75067	6811	METROPLEX GENERAL CONTR
9	0.60	75061	7193	METROPLEX INS AGENCY
10	0.59	75062	6328	METROPLEX ACCEPTANCE
11	0.58	75081	0	METROPLEX PSYCHOTHERAPY INC.
12	0.58	75041	3732	METROPLEX OPTICAL
13	0.58	75042	6621	METROPLEX OPTICAL
14	0.58	75057	4132	METRO PIPE & SUPPLY
15	0.58	75074	5526	MULTIPLEX DISPLAY
16	0.58	75041	2407	METROPLEX PROP MANAGEMENT
17	0.57	75041	5824	METROPLEX DENTAL GROUP
18	0.57	75050	4757	METROPLEX ELECTRONICS
19	0.57	75057	3072	METROPLEX SPECIALTIES
20	0.57	75057	3950	APEX SUPPLY CO.

The deepest level of sort (LOS) "75057–4805" for this address is successfully obtained at the top choice of the neighborhood.

5 Experimental Results

Both the word-based and the character-based matching algorithms were implemented and tested using a set of 786 address block images each containing a firm name. The address block images were pre-processed, segmented into lines and words and parsed. Lines containing a firm name were located by the parser. Performance was measured on only those cases where the parser correctly located the firm name and that firm name has a corresponding entry in the database. Table 5 summarizes the recognition performance using each approach. Table 6 summarizes the reasons

Table 5: Summary of Performance

Category	No. of Images	% of **411**	% of **786**
Images containing firm names	**786**	xxx	100.0
Rejected	152	xxx	19.3
Accepted	634	xxx	80.7
Firm name lines correctly located among accepted	537	xxx	68.3
Correctly located firm names having corresponding entries in database	**411**	100.0	52.3
Results by the word-based method:			
Matched in top 20 choices	337	**82.0**	42.9
Missed in top 20 choices	74	18.0	9.4
Provided ZIP+4 Code	169	**41.1**	21.5
Results by the character-based method:			
Matched in top 20 choices	354	**86.1**	45.0
Missed in top 20 choices	57	13.9	7.3
Provided ZIP+4 Code	173	**42.0**	22.0

of the rejections. Some of the images rejected are firm direct images, which means that the 5-digit ZIP Code on those images are already the deepest level of sort, therefore those firms do not need further processing.

Table 6: Reasons for Rejects

Firm direct images	76
No 5-digit ZIP hypotheses given	16
Invalid SCF given by ZIP hypotheses	36
Database errors causing no output	24
Total	152

A correct rate of 82% in the top 20 choices was achieved using the word-based approach, given that the firm names were correctly located on the mailpiece

and that they have corresponding entries in the record database. Given the same conditions, the character-based approach was able to include the correct records in the top 20 choices for 86% of the cases. This performance is considered useful given that the firm name record database contains about 1.2 million entries in total and 48,000 records per SCF partition on average. Close to half of the correct matches by either approach provided the ZIP+4 Code. This performance could be better if the firm name database was more complete. Among the 786 images, 351 (44.7%) are addresses directed to P. O. Boxes. Many of those P. O. Box addresses do not have the ZIP+4 Codes in the firm name database we used.

Table 7 further breaks down the performance within the top 20 choices. For the 411 images with corresponding records existing in the database, the word-based method matches 56.0% of the correct firm names at the top choice, and the character-based method matches 75.4% of the correct firm names at the top choice.

Table 7: Performance in the top 20 neighborhood

	Word-Based				*Character-Based*			
On or above rank	Firm name found	% of 411	ZIP+4 found	% of 411	Firm name found	% of 411	ZIP+4 found	% of 411
1	230	56.0	90	21.9	310	75.4	101	24.6
5	291	70.8	140	34.1	335	81.5	149	36.3
10	319	77.6	155	37.7	341	83.0	163	39.7
15	333	81.0	165	40.1	351	85.4	170	41.4
20	337	82.0	169	41.1	354	86.1	173	42.0

Table 8 and Table 9 summarize the major causes of the failures where the correct firm names were missed from the top 20 choices.

Table 8: Errors by the word-based method

Reason of failure	Number	% of Total
Irrecoverable character recognition errors	36	48.6
Excessive irrelevant records retrieved by single characters	10	7.4
Word segmentation errors	7	9.5
Recognition errors caused by punctuation in image	4	5.4
Wrong SCF given by ZIP Code hypotheses	4	5.4
Others (abbreviations, acronyms, ..., etc.)	13	17.6
Total	74	100.0

Table 9: Errors by the character-based method

Reason of failure	Number	% of Total
Irrecoverable character recognition errors	44	77.2
Unimportant keywords existing in image or database records	4	7.0
Wrong SCF given by ZIP Code hypotheses	4	7.0
Important keywords missing in image	3	5.3
Alias problems	2	3.5
Total	57	100.0

6 Comparison of the Two Approaches

The biggest advantage of the character-based approach is that errors caused by word segmentation are avoided, since the characters of all the words are concatenated. Also it does not depend on the availability of a lexicon that includes all forms of the valid words. It requires no index files and therefore storage and preparation of the database are minimized.

One disadvantage of the character-based approach is that it relies heavily on the accuracy of character recognition output. As seen in the error analysis, misrecognized characters caused more errors than they did when the word-based approach was used. Improvement is possible if a more sophisticated string matching algorithm is used, such as the n-gram techniques [8], the edit distance methods [7] [5], and probability models [4]. Those algorithm are more expensive but may be able to correct more recognition errors.

The word-based approach needs resources to create the keyword files, alias files, lexicons and indices from the database. However, once these data structures are created, aliases and keywords can be better handled.

The word-based approach allows more flexibility in image recognition methods. Other than the recognition of isolated characters, other word recognition techniques such as word-shape analysis, segmentation-based methods, or a combination of several techniques can be applied [3]. Improvements can also be made by using a more sophisticated scoring scheme, which may include adjustments by word orders and word lengths. For example, a higher score can be assigned to the first word of the firm name to compensate for its relative significance, and a lower score can be assigned to single-character words to degrade the ranks of the excessive irrelevant records they retrieve. Keywords can be assigned more appropriate scores according to their frequencies of occurrences in the firm name database. The correctness of the word-based method can also be improved by creating a more accurate and complete list of aliases and unimportant keywords.

7 Summary and Conclusions

Recognizing the firm names on address block images can be an effective way to derive a correct ZIP+4 Code. Two approaches for firm name recognition are described which use contextual information at different levels. The word-based approach matches segmented and recognized firm words with index words of a firm name database. A heuristic scoring scheme is applied to evaluate the matches. The character-based approach uses the decisions of an OCR directly to match with firm name strings in the database. The matches are evaluated by finding the longest common subsequences.

Both approaches were tested using firm name images extracted from address blocks scanned on a postal OCR. Experiments with 786 images show that the word-based approach successfully matched in the top 20 choices 82 % of the 411 images which have possible matches, while the character-based approach matched 86 % of all such cases.

Each of the two approaches has its own merits and drawbacks. The character-based approach is independent of word segmentation, and the word-based approach allows more possibilities for improvements. The two approaches give independent match results. A further step will be to combine the decisions by the two methods. This will allow rejection of the uncertain cases when the two methods disagree, and improvement of the rankings by consensus. Efficiency in run-time and storage can also be improved by using more advanced data structures in the organization of the database.

Acknowledgements This work was supported by the Office of Advanced Technology of the United States Postal Service under contract 104230-84D-0962, Task Order No. 104230-88D-2576. The authors wish to thank Carl O'Connor of the Office of Advanced Technology of the USPS, Gilles Houle and Gerardo Garcia of Arthur D. Little Inc. for their advice and support for this project.

References

[1] P.B. Cullen, T.K. Ho, J.J. Hull, M. Prussak, S.N. Srihari, "Contextual analysis of machine printed address", *Proceedings of the USPS Fourth Advanced Technology Conference*, Vol 2, Washington, D.C., Nov 1990, 779-793.

[2] P.A. Hall, G.R. Dowling, "Approximate string matching", *ACM Computing Surveys*, Vol 12, Dec 1980, 381-402.

[3] T.K. Ho, *A theory of multiple classifier systems and its application to visual word recognition*, Doctoral Dissertation, Department of Computer Science, State University of New York at Buffalo, 1992.

[4] R.L. Kashyap, B.J. Oommen, "Spelling correction using probabilistic methods", in *Computer text recognition and error correction, tutorial*, edited by Sargur N. Srihari, IEEE Computer Society, 1985.

[5] W.J. Masek, M.S. Paterson, "A faster algorithm computing string edit distances", *Journal of Computer Systems Sciences*, Vol 20, Feb 1980, 18-31.

[6] J. Schuermann, "A multifont word recognition system for postal address reading", *IEEE Transactions on Computers*, C-27, 8, Aug 1978, 721-732.

[7] R.A. Wagner, M.J. Fischer, "The string-to-string correction problem", *Journal of the ACM*, 21, 1, Jan 1974, 168-173.

[8] J.R. Ullman, "A binary n-gram technique for automatic correction of substitution, deletion, insertion, and reversal errors in words", *Computer Journal*, 1977, Vol 20, 141-147.