Pergamon

Pattern Recognition, Vol. 27, No. 2, pp. 311–320, 1994
Elsevier Science Ltd

0031-3203/94 $6.00 + .00

PROPER NOUN DETECTION IN DOCUMENT IMAGES

GINIGE L. DE SILVA and JONATHAN J. HULL†
Center of Excellence for Document Analysis and Recognition, Department of Computer Science, 226 Bell Hall, Buffalo, NY 14260, U.S.A.

(*Received* 10 *December* 1992; *in revised form* 20 *September* 1993; *received for publication* 5 *October* 1993)

Abstract—An algorithm for the detection of proper nouns in document images printed in mixed upper and lower case is presented. Analysis of graphical features of words in a running text is performed to determine words that are likely to be names of specific persons, places, or objects (i.e. proper nouns). This algorithm is a useful addition to contextual post-processing (CPP) or whole word recognition techniques where word images are matched to entries in a dictionary. Due to the difficulty of creating a comprehensive list of proper nouns, a methodology of locating such words prior to recognition will allow for the use of specialized recognition strategies for those words only. Experimental results demonstrate that about 90% of all occurrences of proper nouns were located and over 97% of the unique proper nouns in a document were found using this algorithm.

Proper noun detection Character recognition Word recognition Feature extraction
Capitalized word detection Nearest neighbor classifier

1. INTRODUCTION

The location and recognition of proper nouns in a document image is an important part of any text recognition algorithm. Proper nouns provide the names of people, places, and specific objects. As such, much specialized information is conveyed by these words and their recognition is thus an essential part of understanding the message conveyed by the document.

A problem with the recognition of proper nouns is that it may be costly (in run-time) to utilize a contextual post-processing (CPP) algorithm[1] in which character recognition decisions are matched to the most similar word in a dictionary. This is because of the large number of entries in a comprehensive list of proper nouns. For example, over 500,000 surnames have been identified as being used in the United States.‡ The run-time of a CPP algorithm on such a list may be prohibitive, especially if there is no information available to distinguish proper nouns and every word had to be post-processed vs. that dictionary.

Such a large dictionary would also provide a significant impediment to the use of whole word recognition algorithms.[2,3] These methods compensate for image noise by recognizing whole words as units. An input word image is matched to a dictionary and a ranking of the most visually similar words is produced. The segmentation of a word into characters before recognition is not required. This strategy has been shown to provide a significant improvement in recognition performance in noisy images.[4]

This paper proposes a solution to proper noun recognition in which the word images found by a document segmentation algorithm (such as that proposed by Baird[5]) are filtered to locate potential proper nouns. Features are calculated from each word image as well as the words before and after it. This feature vector is classified to determine whether the word in question is a proper noun. The success of the proposed methodology is determined by how well proper nouns are distinguished from other words.

The word images identified as proper nouns by this procedure could then either be recognized by a character or word recognition algorithm. A large dictionary would thus be used for only a small proportion of the word images in a document. When a proper noun is recognized that does not occur in the dictionary, the character recognition decisions alone could be output. This would be a necessary alternative strategy because of the difficulty of maintaining a comprehensive list of proper nouns.

The rest of this paper presents a proper noun location algorithm. The background for the algorithm is first discussed, including various statistical characteristics of the features that are used. An algorithm for proper noun location based on extracting those features is then presented. This is followed by a discussion of experimental results that demonstrate the performance of the proposed technique. The paper is concluded with a summary and discussion of future directions.

2. ALGORITHM BACKGROUND

To identify the features that distinguish proper nouns from other words, a statistical study was conducted of a large body of text known as the Brown Corpus.[6] The Brown Corpus is composed of over one million words of running text divided into 500 samples of about 2000 words each. The 500 samples were selected from 15 subject categories or genres. Each word is

† Author to whom all correspondence should be addressed.
‡ Derived from the United States Postal Service file of forwarding addresses.

annotated with an indication whether it was capitalized in the source document. The original punctuation was also retained. A part-of-speech (POS) tag was also assigned to each word that indicates the grammatical function of that word in the original passage of text.

The usefulness of seven features in distinguishing proper nouns was measured on the Brown Corpus. The following discussion summarizes the results of this investigation. The objective of this analysis was to determine a feature set that could be measured reliably from the image of a document without requiring a word to be recognized first.

Capitalization. 99.5% of all proper nouns are capitalized. Hence, for practical purposes we can assume that all proper nouns are capitalized. Of the capitalized words, 35% are proper nouns. Also, approximately 64% of the time, at least one word in a group of sequential capitalized words was a proper noun. From this analysis it was concluded that the identification of capitalized words is an essential feature for proper noun location.

Location in a sentence. 45% of the capitalized words occur at the beginning of a sentence. Of these, only 8% are proper nouns. This is only 10% of all proper nouns. Thus, finding a capitalized word at the beginning of a sentence implies a 92% probability that it is *not* a proper noun.

Length of words. The average length of a proper noun is six characters as compared to five for capitalized non-proper nouns. The percentage of capitalized words of a particular length that are proper nouns are given in Table 1. It is seen that more than 50% of the words of lengths six, seven, and eight are proper nouns while very few of the short words with less than four characters are proper nouns.

Length of the previous word. The probability that a proper noun occurs among the capitalized words that follows words of a particular length is also given in Table 1. It is noteworthy that almost 70% of the capitalized words that follow a word of length two are proper nouns while 85% of the capitalized words that follow a word of length one are not proper nouns.

Length of the following word. The probability that a proper noun occurs among the capitalized words that precede words of particular lengths is shown in Table 1.

Table 2. Percentages of proper nouns among capitalized words before and after words with the indicated grammatical tag

Grammatical tag	% proper nouns before	% proper nouns after
Noun	22.2	39.0
Adjective	15.9	37.9
Article	6.5	36.0
Preposition	29.3	70.3
Adverb	22.4	50.9
Verb	34.7	56.8

It can be seen that while about 54% of the capitalized words that occur prior to words of length one are proper nouns, almost 70% of capitalized words that occur prior to words of length more than two are not proper nouns.

Syntactic category of the previous word. The probability that a proper noun occurs among the capitalized words that follow words of particular syntactic categories is presented in Table 2. It can be seen that while about 70% of the capitalized words that follow a preposition are proper nouns, more than 60% of the capitalized words that follow common nouns and adjectives are not proper nouns.

Syntactic category of the following word. The probability that a proper noun occurs among the capitalized words that precede words of particular syntactic categories is also given in Table 2. It is apparent that only a few capitalized words that occur before nouns, adjectives, articles, prepositions, and adverbs are proper nouns.

3. ALGORITHM STATEMENT

Based on the statistical analysis of features presented in the previous section, an algorithm consisting of the following two steps was designed: (1) capitalized word detection; (2) proper noun classification. The detection of capitalized words is a separate step in the algorithm because of its importance, i.e. over 99% of proper nouns are capitalized and capitalized words account for about 10% of an overall text. Thus, the number of

Table 1. Proper nouns and word lengths

Length of the word	% of PNs in such cap. words	% of PNs among succeeding cap. words	% of PNs among prior cap. words	Length of the word	% of PNs in such cap. words	% of PNs among succeeding cap. words	% PNs among prior cap. words
1	5.6	15.4	54.2	8	52.3	36.4	34.0
2	11.8	69.1	31.3	9	49.9	47.9	32.2
3	14.5	50.5	30.0	10	46.7	32.0	31.0
4	34.7	64.6	27.6	11	39.9	37.9	26.4
5	43.4	52.7	30.4	12	38.8	27.8	24.6
6	56.9	48.8	35.2	13	41.7	36.0	25.0
7	53.0	52.6	34.8	14	25.0	30.2	26.9

word "tokens" that need to be examined is reduced by about 90% by first locating words whose first letter is capitalized.

3.1. *Capitalized word detection*

An objective in the design of the capitalized word detection algorithm was to develop a technique that did not require accurate character segmentation. This was to provide tolerance to image noise. One solution would have been to segment and recognize the first character of every word and use that result to derive a decision. However, this was judged to be unacceptable since it would be sensitive to character segmentation and recognition performance. Instead, feature analysis was preferred that required less specific information.

The detection of capitalized words is performed by first locating the baselines (lower, middle, and upper) in an input word. A zone at the beginning of each word is then located that should contain a large portion of the first character. Global features common to many capitalized letters and local features common to specific capitalized letters are then extracted within this zone. The discrimination between capitalized and non-capitalized words is performed by a rule-based system.

3.1.1. *Baseline estimation.* The lower, middle, and upper baselines in a word image are shown in Fig. 1. The lower baseline is the line on which all characters without descenders rest. The middle baseline is the top of all lower case characters without ascenders and the upper baseline is the highest point reached by all capitalized letters. The distance between the lower and upper baselines is referred to as the *cap-height*, while the distance between the lower and middle baselines is called the *x-height* or the *middle zone*. The area between the middle and upper baselines is known as the *upper zone*. An area known as the *cap-width* is denoted at the beginning of the word that approximates the average width of a capitalized letter. The *cap-width* area is used for capitalized letter feature extraction.

Baselines are estimated using a method called projection profile analysis. For a given word image, the number of black pixels in each row is represented in a histogram (Fig. 1). The middle and lower baselines correspond to the rows in the histogram at which the maximum row-to-row discontinuity is encountered when the histogram is scanned from the top and bottom, respectively. The upper baseline is the first row in the histogram in which pixels are encountered when looking from the top. When baselines are estimated using isolated words, this method is accurate for long words, but is failure-prone for words with fewer than three or four characters. Stable estimates of the baselines are achieved by calculating the histogram on the image of an entire text line.

3.1.2. *Global feature analysis.* A word image is first checked for the presence of a hole (a hole is defined as an area totally enclosed by black pixels) in the *upper zone* within the *cap-width* (Fig. 2(a)), which is a certain indicator that the word is capitalized. Also, if the image

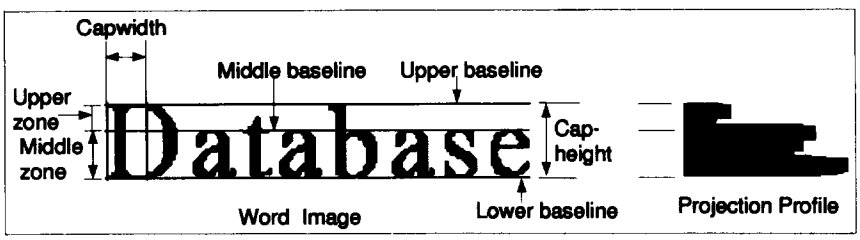

Fig. 1. Lower, middle and upper baselines.

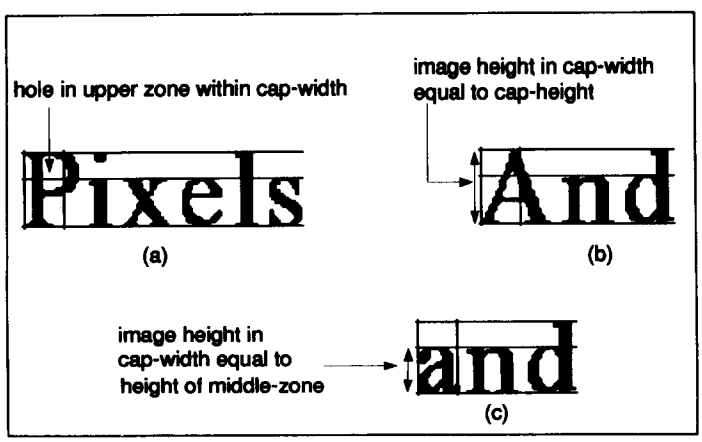

Fig. 2. Examples of global features.

height within the *cap-width* is nearly equal to the *cap-height* (Fig. 2(b)), this indicates the word might be capitalized. If the image height within the *cap-width* is about the height of the *middle zone* (Fig. 2(c)), this indicates the word is lower case.

If the word image cannot be positively identified as capitalized or lower case by the above global features (features shown in Fig. 2), then it is checked for multiple occurrences of other global features within the *cap-width* that might indicate capitalization. These features are: high number of black pixels in the *upper zone*, high number of black pixels in the *upper zone* contributing to strokes in the northeast–southwest and northwest–southeast directions, high number of black pixels in the *upper* and *middle zones* contributing to strokes in the northeast–southwest direction, and the presence of a period in the word immediately prior to the word being processed.

Typically, the analysis of global features is sufficient to locate the letters B, D, O, P, Q, R, W, X, and Y. The other upper case letters are distinguished by the presence of letter-specific features.

3.1.3. *Letter-specific feature analysis.* If a word image does not pass any of the global feature tests, it is inspected for the presence of letter-specific features. If an image matches any of the letter-specific feature tests, it is classified as capitalized. Otherwise, the word is classified as non-capitalized.

This analysis depends on the extraction of several features. The *slope* feature applies to the first black pixels encountered when an image is scanned either from the left or the top. The slope is estimated by measuring the difference between the x coordinates of the two boundary pixels found both two rows above and below each pixel. Only the sign (positive or negative) of the slope is retained (Fig. 3(a)). The detection of *strokes* is performed by analysis of a vertical (for vertical strokes—Fig. 3(b)) or horizontal (for horizontal strokes—Fig. 3(c)) projection profile taken over a fixed distance. A stroke is present if a minimum of two contiguous rows or columns each contain at least a number of black pixels that is approximately equal to *cap-width* for rows and *cap-height* for columns.

The following letters are detected primarily by analysis of the slope feature (A, C, G, S, V, Z). These are letters that contain no vertical strokes. The percentage of pixels that have either a positive or negative slope within a given horizontal area scanned from the left or vertical area scanned from the top are computed. Table 3 summarizes the feature tests on these measurements. For example, letter A is located if the number

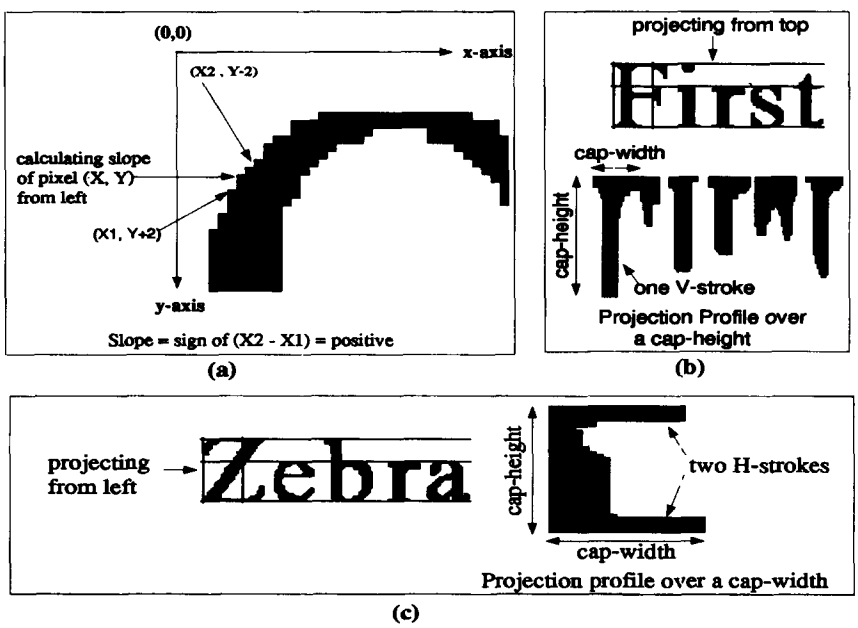

Fig. 3. Slope and stroke estimation.

Table 3. Slope feature rules

Letter	Proj.	Sign	Location	Proj.	Sign	Location
A	H	+	cap-height	V	–	between 1 and 2 cap-widths
C, G	H	+	top half cap-height	H	–	bottom half cap-height
S	H	+	top third cap-height	H	–	middle third cap-height
V	H	–	cap-height			
Z	H	+	cap-height			

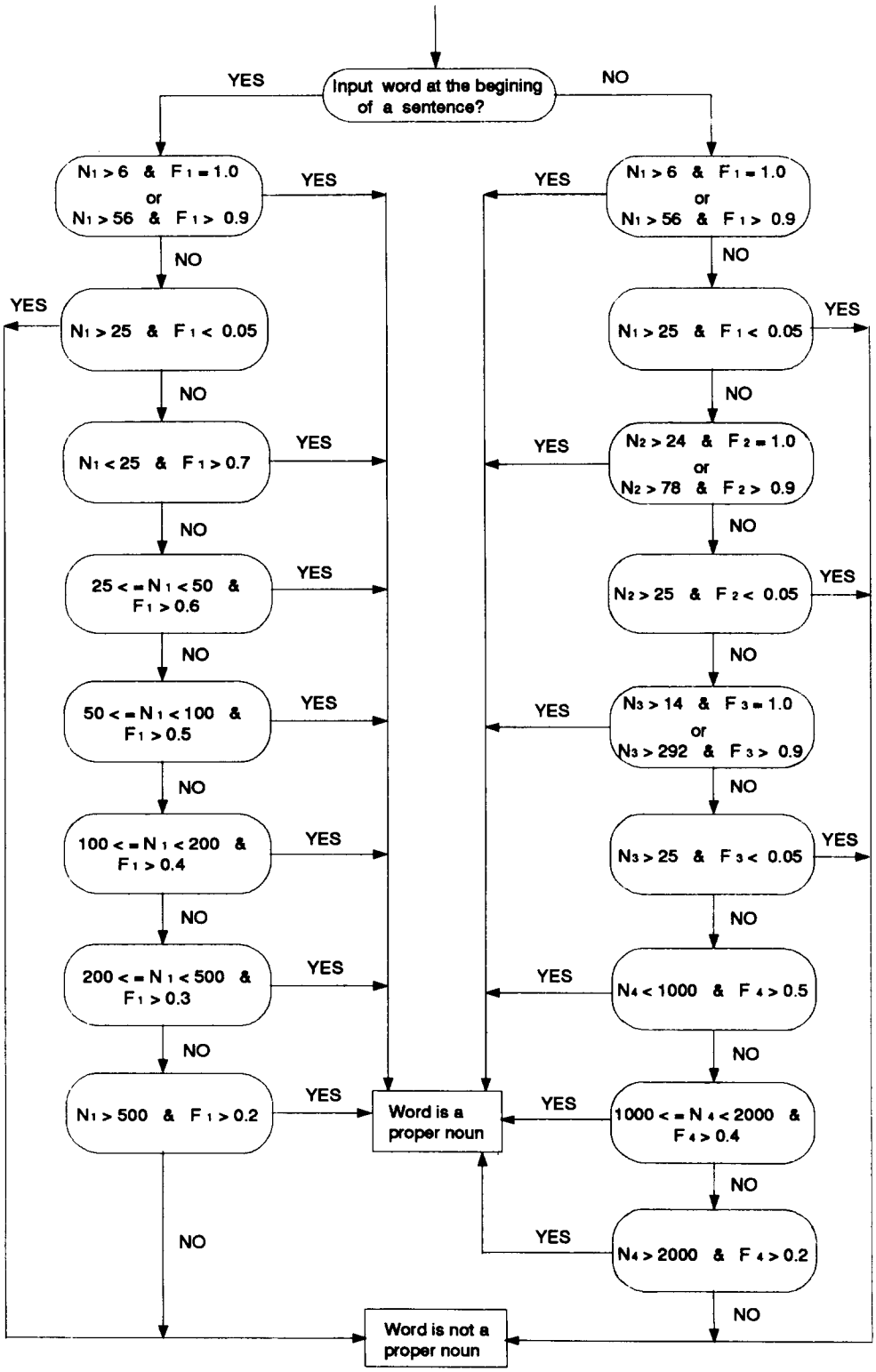

Fig. 4. Decision tree.

of pixels projected from the left that have a positive slope is nearly equal to the cap-height. Also, the number of pixels projected from the top with a negative slope that occur between one and two cap-widths from the left side of a word should be above a threshold.

The detection of the following letters depends primarily on the location of strokes within the image (E, F, H, I, J, K, L, M, N, T, U). If a vertical stroke about the height of *cap-height* is found within the first cap-width of the image, then the image is tested for each of the following letters in turn. The additional tests needed for each letter are as shown. Horizontal strokes are denoted *H-strokes* and vertical strokes *V-strokes*.

E: three *H-strokes* within the first cap-width;
F: two *H-strokes* within the first cap-width;
H: one *H-stroke* in middle 40% and one *V-stroke* in second cap-width;
I: short *H-strokes* on top and bottom and nothing attached to middle 60% of *V-stroke*;
J: *V-stroke* on the right and nothing attached to middle 30% of *V-stroke*;
K: percentage of pixels with positive slope projected from the top within one cap-width above threshold;
L: *H-stroke* on bottom and nothing attached in middle 60% of *V-stroke*;
M: one *V-stroke* in second cap-width as well as positive and negatively sloped pixels as projected from the top;
N: one *V-stroke* in second cap-width and negatively sloped pixels projected from the top;
T: one *H-stroke* in top 10% of the image and number of white pixels to left of *V-stroke* is low;
U: one *V-stroke* in second cap-width and *H-stroke* in bottom 10% and nothing attached to middle 60% of *V-stroke*.

3.2. *Proper noun classification*

Once an image is identified as a capitalized word and its position within a sentence is determined, a feature vector is calculated from the surrounding text. Elements of this vector are the number of characters in the word, number of characters in the previous word, number of characters in the following word, part-of-speech (POS) tag of the previous word (e.g. Noun, Verb, etc.), POS tag of the following word, whether the previous word is capitalized, and whether the following word is capitalized.

The number of characters in a word image can be estimated using a combination of connected component and projection profile analyses.[1] The POS tags of the previous and following words can be identified with high accuracy in a word recognition application using a hidden Markov model.[7]

The classification of a feature vector for an unknown word is performed with a modified nearest neighbor classifier structured as a decision tree. A database of prototype feature vectors for use in the classifier was built from a large set of capitalized words (100,000+) that were extracted from the Brown Corpus. Each vector is labeled as to whether the word associated with it is a proper noun.

Table 4. Variations in lengths and POS tags for each matching level

Level No.	Previous length	Current length	Following length	POS tags
1	0	0	0	exact
2	1	1	1	exact
3	1	3	1	exact
4	2	3	6	any tag

Four levels of matching are calculated between an input vector and the prototype set by varying the distance allowed between the input and a prototype for a "match" to occur. The total number of prototype vectors that match (N_i) and the number that correspond to proper nouns (P_i) are calculated for each level i ($i = 1, \ldots, 4$). Table 4 summarizes the parameter variations at each level. For example, the entry for level 2 specifies that matches are counted for prototypes where the length of the current, previous, and following words vary by at most one but the POS tags of the previous and following words match exactly.

The decision tree (shown in Fig. 4) uses the location of the word in a sentence and the N_i and P_i values to determine a structured series of tests. Word location is a significant determiner of performance because of the small number of proper nouns that occur at the beginning of sentences. The values for the thresholds in the tree were determined by inspection of performance on a small training set (three articles from the Brown Corpus).

The thresholds at each level are based on the concept of the *quality* and the *strength* of the match between the input vector and the prototype set. When the number of prototypes in the training set that match the input at each level is low (e.g. $N_i < 25$), even locating a high percentage of prototypes that are proper nouns (i.e. the fraction $F_i = P_i/N_i$ is high) may indicate little about whether the input vector represents a proper noun. So the number N_i indicates the "quality" of the match. The fraction F_i indicates the "strength" of the match, so that a higher value for this fraction indicates that most of the prototypes that match the input vector are proper nouns.

4. EXPERIMENTAL RESULTS

4.1. *Database generation*

Databases of word images for training the algorithm were generated using two software tools. The first is a device-independent text formatting and typesetting tool called *ditroff* that is used to generate postscript images of ASCII text in several fonts. The second is an image format conversion tool that is used to convert the postscript into bit-maps with a specified resolution.

Table 5. Performance of the capitalized word detector

Type of test	Total No. of words	No. of cap. words	Number identified as caps	% correctly identified as caps	% of all caps located
Unique words from the Brown Corpus (generated)	53,024	13,625	15,001	90.83	99.76
Randomly selected magazine article (scanned)	717	120	122	95.1	96.67

Table 6. Performance of the proper noun classifier

Type of test images	Total No. of words	No. of cap. words (CW)	Total No. of proper nouns (PN)	No. of PNs identified	% of PNs identified	No. of CWs misclassified	% accuracy of prediction	% of CWs misclassified
Generated with POS tags	26,358	3254	1081	973	90.0	113	89.6	3.5
Generated with no POS tags	26,358	3254	1081	960	88.8	92	91.3	2.8
Scanned with no POS tags	717	120	59	52	88.1	9	85.2	7.5

This process of database generation has the advantage that the word images are automatically *truthed* as they are generated.

4.2. *Algorithm performance evaluation*

Evaluation of algorithm performance was done in three stages. The strategy was to test the capitalized word detector and proper noun classifier independently and then to test the combined system. This way the contribution of each part of the algorithm to the overall performance could be determined. Another variation was to use two types of testing data: word images *generated* from the ASCII text of the Brown Corpus (using the process mentioned above) and word images *segmented* from an existing document. The advantage of using generated word images is the availability of a wide variety of word types with different letter combinations and fonts. This is especially important for testing the capitalized word detector. The word images from an existing document provide a more realistic test of the entire algorithm as it would be used in a text recognition system.

The generated data were used to develop the parameters of the algorithm. The word images segmented from a magazine article† were used as an independent test set, separate from all training processes. Thus, results derived from this test set validate the performance of the algorithm.

4.2.1. *Performance of the capitalized word detector.* Results of testing the capitalized word detector on the full set of unique words from the Brown Corpus are presented in the first row of Table 5. Overall, there are 53,024 unique words in the Corpus, of which 13,625 are capitalized. The main group of misclassified words was numerals (8.75% of all words classified capitalized). The percentage of words that are numerals in this test set over-represent the occurrence of numerals in the running text. Thus, the 8.75% of words misclassified will not cause a significant problem in processing a running text. Numerals present special problems by having a significant number of pixels present in the upper left corner of their images, thus leading to misclassification as capitalized words. One way to get around this would be to include numeral recognition as a part of the capitalized word detector, so numerals can be successfully separated from capitalized words. This set of experiments was performed with images generated in a 10-point Times-Roman font. Other experiments, not discussed here, were also conducted with various fonts and point sizes that provided similar results.

Results of testing the capitalized word detector on the 717 word images from the scanned document are presented in the second row of Table 5. One type of problem encountered here was the presence of a different font within the article. Three of the capitalized words missed by the detector were in a different font than the rest of the article. Among misclassified non-capitalized words were three words that were numerals. Altogether, only four out of the 120 capitalized words (3.3%) were not found.

4.2.2. *Performance of the proper noun classifier.* Results of testing the proper noun classifier on words generated from 13 randomly selected articles from the

† Memoirs: Andrei Sakharov, "The Poisonous Legacy of Trofim Lysenko", page 61, *Time*, 14 May 1990.

The Poisonous Legacy of Trofim Lysenko

Under Stalin and Khrushchev, the biologist Trofim Lysenko terrorized Soviet scientists. A ruthless political infighter, Lysenko rejected Mendelian genetics, favoring the ideas of Ivan Michurin, who held that modifications acquired by one generation of plants and animals could be passed on to future generations. Lysenko's notions poisoned Soviet agriculture—and science—for decades. Sakharov, who considered Lysenko a crackpot and a bully, unfortunately confronted him and his Mafia.

In 1950 a commission visited the Installation to check up on senior scientists. I was called in and asked what I thought of the chromosome theory of heredity: after Stalin's endorsement of Lysenko, belief in Mendelian genetics was regarded as an indication of disloyalty. I replied that the theory seemed scientifically correct. The commission members exchanged glances but said nothing. But Lev Altshuler, who had played a major role in the development of atomic charges, gave the same answer and faced dismissal.

When Avraami Zavenyagin, a KGB lieutenant general and a top nuclear weapons program official, visited the Installation, I urged him to appeal the decision. Zavenyagin paid close heed to scientists and understood their role in the project. He said, "I'm aware of Altshuler's hooligan conduct. You say he's done a lot and will be useful in the future. Fine. We won't take action now, but we'll watch how he behaves."

How did Lysenko and his gang maintain their positions through the Khrushchev era, when it was no longer a simple matter of using the tactics of denunciation and pseudo philosophy that had served them so well in the 1930s and 1940s? Lysenko was always ready with a new idea that promised the sort of quick fix for Soviet agriculture that Khrushchev found irresistible. (And when that fell through, Lysenko would be ready with a new, equally surefire idea.) Even more important: the party agriculture bureaucracy was full of people who supported Lysenko and bitterly opposed proper experiments as a threat to their vested interests.

In June 1964 regular elections for membership in the academy were held. The biologists had voted to elevate Nikolai Nuzhdin to full member. Nuzhdin was one of Lysenko's closest associates, an accomplice in his persecution of genuine scientists. As I recalled the tragedy of Soviet genetics and its martyrs, my indignation boiled up. When Nuzhdin was placed in nomination, I raised my hand. I said:

THE BIOLOGIST AT A COLLECTIVE FARM IN 1949

"The academy's charter sets very high standards for its members with respect to both scientific merit and civic responsibility. Nuzhdin does not satisfy the criteria. He and Lysenko bear the responsibility for the shameful backwardness of Soviet biology and of genetics in particular, for the dissemination of pseudoscientific views, for the degradation of learning and for the defamation, firing, arrest, even death of many genuine scientists. I urge you to vote against Nuzhdin."

There was a deafening silence followed by cries of "Shame!"—but also by applause in the greater part of the hall. Lysenko exclaimed in fury, "People like Sakharov should be locked up and put on trial!"

The physicist Pyotr Kapitsa told me later that Leonid Ilyichev, head of the Central Committee's agitation and propaganda department and a member of the academy's presidium, had been upset by my speech and wished to take the floor. He asked, "Who's that speaking?" "That's the father of the hydrogen bomb," Kapitsa replied. Ilyichev apparently decided it would be more politic to remain silent.

Nuzhdin's bid to become a full member of the academy failed.

I heard that my speech against Nuzhdin had enraged Khrushchev to the point that he stomped his feet and ordered the KGB to gather compromising material on me. Khrushchev supposedly said, "First Sakharov tried to stop the hydrogen bomb test, and now he's poking his nose again where it doesn't belong."

Soon afterward, in October 1964, Khrushchev was vacationing by the Black Sea when he was summoned to an urgent meeting of the Presidium. He rushed to the Kremlin and stalked into the room where the Presidium was in session. "What's going on here?" he demanded. Told that the members were discussing his removal from office, he cried, "Are you crazy? I'll have you all arrested right now!" Khrushchev phoned Rodion Malinovsky, the Defense Minister. "As Commander in Chief, I order you to arrest the conspirators at once." Malinovsky replied that he would carry out the decision of the Central Committee. Vladimir Semichastny, the KGB chairman, also refused to help.

Khrushchev's fall led to the final rout of Lysenko and his supporters. The previously "disgraced" geneticist Nikolai Dubinin was soon elected to the academy and was made director of the Institute of Genetics in 1966. For the next few years, Dubinin sent me New Year's cards recalling how valuable my intervention had been. ■

(a)

KGB	Nuzhdin
Lysenko	KGB
Khru-shchev	Khrushchev's
Valadimir	KGB

(c)

Stalin	Lysenko	Mendelian	Lev	Alt-shuler	Avraami	Za-venyagin	Zavenyagin	Altshuler's	You	How	Lysenko
Khrushchev	Lysen-ko	Soviet	Khru-shchev	Lysenko	Lysenko	June	Nikolai	Nuzhdin	Nuzhdin	Ly-senko	I
Soviet	When	Nuzhdin	He	Lysenko	Sovi-et	Nuzhdin	Shkharov	Pytor	Kapitsa	Leonid	Ilyichev
Committee's	Kapisa	Ilyichev	Nuzhdin's	Nuzhdin	Khrushchev	Sa-kharov	Soon	October	1	964	Khru-shchev
Black	Sea	Kremlin	Told	Khrushchev	Rodion	Malinovsky	Commander	Malinovsky	Se-michastny	Lysenko	Nikolai
Dubinin	For	Du-binin									

(b)

Fig. 5. Performance on the scanned document: (a) page image of the article; (b) words recognized as proper nouns; (c) proper nouns not recognized.

Table 7. Performance of the whole system

Type of word images	Total No. of words	No. of cap. words (CW)	Total No. of proper nouns (PN)	No. of PNs identified	% of PNs identified	No. of CWs misclassified	% accuracy of prediction	% of CWs misclassified
Generated	6121	945	419	387	92.4	49	88.8	0.8
Scanned	717	120	59	51	86.4	12	81.0	1.7

Brown Corpus that were not contained in the training data are given in the first and second rows of Table 6. This sample of text contained 26,358 words of running text. The results of testing on this text both with and without the use of POS tags are shown. When POS tags are not used, the criteria of exact POS tag matches at levels 1, 2, and 3 in Table 4 are ignored. It is apparent that performance decreases slightly when POS tags are not used in the classification. But still, close to 90% of all proper nouns were identified. One observation made during testing, although not apparent from the table, is that certain articles deviate significantly from the norm (e.g. one article had a list of 10–15 proper nouns strung together in one sentence—a list of city names occurring in a report format). Reduced performance on such articles is expected, since the algorithm is based on the statistics of proper noun occurrence in normal English text.

Results of testing the classifier on the scanned word images are shown in the third row of Table 6. Again the algorithm provided nearly a 90% correct rate. This is important because the database of prototype vectors used in the nearest neighbor classifier was built from the Brown Corpus and the classifier may be biased in some way towards the articles in the Corpus. This test demonstrates that the classifier performs as well for a randomly selected article as for an article selected from the Corpus.

4.2.3. *Performance of the whole system.* Results of testing the combined capitalized word detection and proper noun classification system on words generated from three randomly selected articles of the Brown Corpus as well as the scanned word images are given in Table 7.

The system identified more than 92% of the proper nouns among the generated images and more than 86% of the proper nouns among the scanned images. Shown in Fig. 5 is the image of the magazine article and the list of proper nouns the system located as well as the list of proper nouns it missed. It can be seen that among the proper nouns missed by the system, three (Nuzdin, Lysenko and Khru-shchev) were recognized elsewhere. Another three (KGB) were in a font different from the dominant font of the article. Only two proper nouns (Valadimir, Khrushchev's) were truly missed by the system. Thus, the algorithm correctly located nearly 97% of the unique proper nouns in the document.

5. SUMMARY AND CONCLUDING REMARKS

An algorithm was presented that locates proper nouns in a document image by analyzing various features of words in a running text. The algorithm has two stages: capitalized word detection and proper noun classification. It was shown that the graphical contexts of word images were helpful in capitalized word detection, while the context in which these capitalized words occur was important for proper noun classification. Overall, the algorithm located over 90% of all occurrences of proper nouns and 97% of all the unique proper nouns within a test document.

Future work on this topic should include more extensive testing of the algorithm. Also, the algorithm should be extended to recognize numerals and the use of classifiers that are easily retrained such as a classification and regression tree (CART)[8] or a neural network should be considered. Finally, this algorithm should be integrated with other language level analysis methods for document images.

REFERENCES

1. S. Mori, C. Y. Suen and K. Yamamoto, Historical review of OCR research-and-development, *Proc. IEEE* **80**, 1029–1058 (1992).
2. J. J. Hull, Hypothesis generation in a computational model for visual word recognition, *IEEE Expert* **1**, 63–70 (1986).
3. T. K. Ho, J. J. Hull and S. N. Srihari, A computational model for recognition of multifont word images, *Mach. Vision Applic.* **5**, special issue No. 3 on Document Image Analysis, 157–168 (1992).
4. T. K. Ho, J. J. Hull and S. N. Srihari, A word shape analysis approach to lexicon based word recognition, *Pattern Recognition Lett.* **13**, 821–826 (1992).
5. H. S. Baird, Background structure in document images, *Advances in Structural and Syntactic Pattern Recognition*, H. Bunke, ed., pp. 253–269. World Scientific, Singapore (1992).
6. H. Kucera and W. N. Francis, *Computational Analysis of Present-day American English*. Brown University Press, Providence, Rhode Island (1967).
7. J. J. Hull, A hidden Markov model for language syntax in text recognition, *Proc. 11th IAPR Int. Conf. on Pattern Recognition*, The Hague, The Netherlands, pp. 124–127 (1992).
8. L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*. Wadsworth, Belmont, California (1984).

About the Author—GINIGE L. DE SILVA received a B.S. degree in electrical and computer engineering from the State University of New York at Buffalo in 1991. He is currently pursuing a Master's degree in computer engineering at the State University of New York at Buffalo. Since April 1991 he has been a Graduate Research Assistant at the Center of Excellence for Document Analysis and Recognition. His areas of research interests include image processing and pattern recognition. Mr De Silva is a member of Tau Beta Pi and Eta Kappa Nu.

About the Author—JONATHAN J. HULL received the BA degree in computer science and statistics (double major) in 1980, as well as the M.S. and Ph.D. degrees in computer science in 1982 and 1987, respectively, from the State University of New York at Buffalo. From 1984 to the present he has been a full-time member of the research staff at SUNY Buffalo and is presently a Research Associate Professor in the Department of Computer Science and the Associate Director of the Center of Excellence for Document Analysis and Recognition. His research interests include pattern recognition and various aspects of document analysis and recognition.