27. Chen, M.-Y. et al.: Off-line handwritten word recognition using hidden Markov model, Proc. of the 5th USPS Advanced Technology Conf., 563-577 (1992).
28. Tao, C.: A generalization of discrete hidden Markov model and of Viterbi algorithm, *Pattern Recognition* 25(11), 1381-1387 (1992).
29. Lari, K., Young, S.J.: The estimation of stochastic context-free grammars using the Inside-Outside algorithm, *Computer Speech and Language* 4, 35-36, 1990.
30. Levinson, S. E., Continuously variable duration hidden Markov models for automatic speech recognition, *Computer Speech and Language* 1, 29-45 (1986).
31. Brugnara, F., et al.: A family of parallel hidden Markov models, Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'92), 377-380 (1992).
32. Kundu, A., Bahl, P.: Recognition of handwritten script: a hidden Markov model based approach, Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'88), 928-931 (1988).
33. Kundu, A., He Y., Bahl, P.: Recognition of handwritten word: first and second order hidden Markov model based approach, *Pattern Recognition* 22(3), 283-297 (1989).
34. Gillies, A. M.: Cursive word recognition using hidden Markov models, Proc. of the 5th USPS Advanced Technology Conf., 557-562 (1992).
35. Gilloux, M., Leroux, M.: Recognition of cursive script amounts on postal cheques, Proc. of the 5th USPS Advanced Technology Conf., 545-556 (1992).
36. Chen, M.-Y., Kundu, A., Srihari, S.N.: Unconstrained handwritten word recognition using continuous density variable duration hidden Markov model, Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'93) (1993).
37. Park, H.S., Lee, S.-W.: Off-line recognition of large-set handwritten Hangul (Korean script) with hidden Markov models, Proc. of the Third International Workshop on Frontiers in Handwriting Recognition (IWFHR-3), 51-61 (1993).
38. Caesar, T., et al.: Recognition of handwritten word images by statistical methods, Proc. of the Third International Workshop on Frontiers in Handwriting Recognition (IWFHR-3), 409-416 (1993).
39. Chen, M.-Y., Kundu, A.: An alternative to variable duration HMM in handwritten word recognition, Proc. of the Third International Workshop on Frontiers in Handwriting Recognition (IWFHR-3), 82-91 (1993).
40. Gilloux, M., Bertille, J.-M., Leroux, M.: Recognition of handwritten words in a limited dynamic vocabulary, Proc. of the Third International Workshop on Frontiers in Handwriting Recognition (IWFHR-3), 417-422 (1993).
41. Ha, J.-Y., et al.: Unconstrained handwritten word recognition with interconnected hidden Markov models, Proc. of the Third International Workshop on Frontiers in Handwriting Recognition (IWFHR-3), 455-460 (1993).
42. Bertille, J.-M., El Yacoubi, M.: Global cursive postal code recognition using hidden Markov models, Proc. of the 1st European Conf. on Postal Technologies (JetPoste'93), 129-138 (1993).
43. Leroux, M., Salomé, J.-C., Badard, J.: Recognition of cursive script words in a small lexicon, Proc. of the 1st International Conf. on Document Analysis and Recognition (ICDAR'91), 774-782 (1991).
44. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 6, 721-741 (1984).

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

Language-Level Syntactic and Semantic Constraints Applied to Visual Word Recognition

Jonathan J. Hull

Center of Excellence for Document Analysis and Recognition
Department of Computer Science
State University of New York at Buffalo
Buffalo, New York, 14260
hull@cs.buffalo.edu

Abstract. Various aspects of using language-level syntactic and semantic constraints to improve the performance of word recognition algorithms are discussed. Following a brief presentation of a hypothesis generation model for handwritten word recognition, various types of language-level constraints are reviewed. Methods that exploit these characteristics are discussed including intra-document word correlation, common vocabularies, part-of-speech tag co-occurrence, structural parsing with a chart data structure, and semantic biasing with a thesaurus.

1. Introduction

Information above the level of individual words can significantly improve the performance of algorithms that transform images of text into their ASCII equivalent. This area of research uses algorithms and techniques from graphical text layout, information retrieval, and natural language processing, to modify recognition decisions so that they are consistent with the contextual information provided by a coherent passage of text.

The paradigm for handwriting recognition assumed by the techniques presented in this paper is shown in Figure 1. Word images segmented from a passage of text are passed to a word recognition algorithm that computes a ranked set of n hypotheses or potential word decisions that are chosen from a dictionary. Referred to as the *neighborhood* of the word, this hypothesis set should rank the correct decision at the top position as often as possible. In cases where the most highly ranked choice is not the correct decision, it should appear somewhere in the neighborhood. The word recognition algorithm can use a combination of isolated character recognition, postprocessing of the character decisions versus the dictionary, and wholistic word recognition [12]. The methods

for language-level analysis discussed in this paper utilize global contextual information extracted from the entire document to improve the performance of the word recognition technique.

The definition of *improved performance* in word recognition has many aspects. The most basic is an overall improvement in correct rate, i.e., rearranging the neighborhood so that the top decision is correct more often. Also, a reduction in the size of the neighborhood improves performance as long as an error is not introduced. This happens when the correct word is removed. Another important aspect of improving performance is determining a subset of words from the input document that have a higher probability of correctness than that provided by the recognition algorithm. Such a subset of high confidence decisions can be used as "islands" to drive further processing of the other neighborhoods.

Many techniques have been used to improve the performance of text recognition algorithms. Baird used the semantics of chess to select a sequence of character decisions that produced a legal series of chess moves [1]. The simultaneous use of character transitional probabilities and word collocation statistics has also been addressed [16]. Natural language information, including orthographic constraints, syntactic and semantic knowledge have been discussed [5]. Language-level syntactic knowledge is also an important part of many approaches to speech recognition [27]. Several methods have been used that incorporate syntactic information including probabilities [25], and chart parsing [3,26].

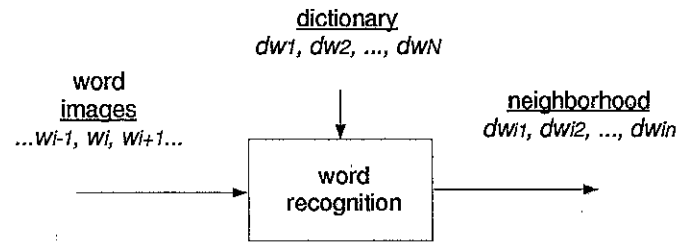

Figure 1. Paradigm for text recognition.

Five contextual constraints that are useful for improving the performance of a text recognition algorithm are discussed in this paper. *Graphical* constraints refer to the context between words within a document prepared by a single writer. The consistency of physical writing style within a document allows for the determination of clusters of equivalent word images. The clusters are then mapped onto language statistics and function words (short, frequently occurring words such as *the, of, and, a,* and *to.*) are recognized. Also, clusters are located that contain potential keywords or *content words* that indicate the topic of the passage of text. This method improves performance by recognizing function words with high confidence. Also, clusters of potential content words are used to generate feature representations that improve their recognition performance.

Vocabulary constraints are the commonality between the words used by authors when writing a passage of text about a specific topic. These constraints narrow the gap between the large dictionary that is needed for general purpose word recognition (on the order of 100,000 or more words) and the limited vocabulary that may be used in a given document (maybe as few as 200 to 500). Techniques from information retrieval are used to match the neighborhoods of word decisions calculated from an image to ASCII documents in a free-text database. The vocabularies from the most similar documents are used to locate a subset of word decisions that are correct with high confidence.

Statistical constraints quantify the predictive ability of words or other grammatical characteristics in a text passage. The most common of these are word collocation data. Another technique is to use the information given by the part-of-speech tag (e.g., noun, verb, etc.) for a word to constrain adjacent part-of-speech tags. Only word decisions with those tags are "legal" in the given context. This technique improves the overall performance of a word recognition algorithm by removing words from neighborhoods that do not have the estimated grammatical characteristic.

Structural-syntactic constraints refer to the information provided by a full parse for a sentence. Typically calculated by a structural method such as a chart parser, the parse for a sentence expresses syntactic information from several contiguous words. A modified chart parser that operates directly on the neighborhoods provided by a word recognition algorithm can choose decisions from the neighborhoods that are consistent with the entire sentence. This technique improves text recognition performance by choosing the single best decision for each word. Modifications that use multiple parses to increase the number of choices for each word are also possible.

Structural-semantic constraints refer to the "glue" that binds together words within a document. Because of the commonality of theme within a coherent

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

passage, it has been observed that groups of words can be identified that are about the same subject. For example, a passage of text about a river is likely to contain the words *water, boat, dock, bank,* and so on. The semantic relationship between words can be calculated from a data structure such as a thesaurus. This information is used to improve the performance of a text recognition algorithm by constraining decisions for selected words to be similar in topic.

The rest of this paper discusses each of the categories mentioned above in more detail. Example algorithms are given and the ability of those techniques to improve text recognition is discussed.

2. Graphical Constraints

The consistency of writing style within a handwritten document is a valuable source of information that can improve recognition performance. Given a document written by a single person, the commonality of appearance between images of the same word can allow for the determination that two images are equivalent. That is, they represent the same word.

Statistics about the frequency of word occurrence within documents are a valuable constraint on the repetition of distinct words. This has been used for solving substitution ciphers and OCR processes [20]. Two especially useful word occurrence characteristics are the frequency of *function* words and the internal frequency of *content* words within a document.

Function words are short determiners or prepositions that supply syntactic information about nearby words. For example, over 75 percent of the words that follow a determiner are either common nouns or adjectives. Examples of function words include *the, of, and, a,* and *to*. In fact, these five words as well as the other five most common function words account for about 23 percent of the word tokens in the over one million words of running text known as the Brown Corpus [18].

Content words are usually nouns that convey information about the topic of an article. Often, the same content words recur several times within an article. This effect is utilized in document classification and information retrieval techniques that select keywords based on their frequency. Often the first step in locating a keyword (index term) is to discard the most frequent words, which tend to be function words, and choose the words from the remaining set that have high internal frequency within the document [24]. These methods have also been extended to phrasal indexing that uses repeated groups of words to improve the effectiveness of document classification.

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

The accurate recognition of both function and content words is essential in a text recognition system. The syntactic information provided by function words can be used to constrain the choices for nearby words. Because of their importance for document classification and subsequent information retrieval processes, content words must also be correctly recognized.

An algorithm has been proposed that uses information about the repetition of words within a document to improve the recognition of both function and content words [17]. Clusters of *equivalent* word images are determined by *matching* images to one another and improved protctypes are generated for the words in a cluster by using inter-image redundancy to eliminate noise. Function words are then recognized by a combination of intra-cluster and inter-cluster statistical characteristics as well as matching to stored prototypes.

Advantages of the technique include its limited use of an explicit recognition algorithm. The basic operation of word image matching implies that relatively high noise levels can be tolerated. Touching, broken, or degraded characters that would render typical recognition algorithms useless are easily compensated for by inter-word redundancy.

Figure 2 outlines the steps of the word image matching algorithm. Word images from an entire sample of text are segmented, and sent to a clustering algorithm sequentially. The algorithm iteratively groups the images into clusters of equivalent words. The ideal result is a number of clusters, each containing all the occurrences of a specific word in the document.

Several operations are then performed on the resulting word image clusters. *Cluster identification* refers to the process of identifying the word contained in a given cluster, possibly without direct recognition of the word images. This is done by using both language statistics and cluster properties. This method of identifying the content of clusters is more effective with clusters containing the function words because their frequency of occurrence in the language is more likely to match their frequency of occurrence in a document.

The clusters containing potential content words are located by inspecting the number of words in each cluster and the lengths of those words. Using the redundancy between images in these clusters, improved (less noisy) prototypes for the content words are generated. A word recognition algorithm (not described here) could then be applied to the improved prototypes to yield better performance than would be possible on the individual images.

This technique has been successfully applied to machine-printed documents. Over 95 percent of the function words in a set of documents were recognized and 93 percent of the important keywords were located [17]. These experiments were conducted on documents that exhibited moderate to extreme noise.

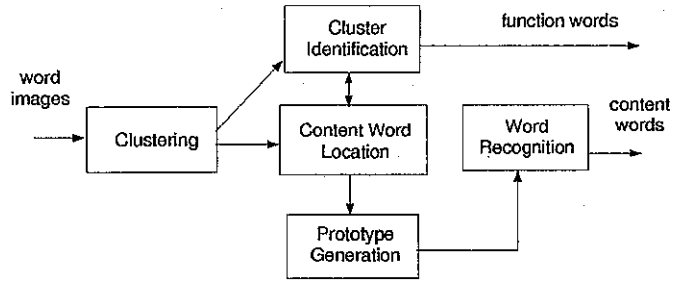

Figure 2. Word image matching algorithm.

The extension of this and other techniques that use graphical context to handwritten documents would depend on the success of the algorithm that determines whether two words are equivalent. This poses an interesting problem for future research. An important consideration would be whether on-line or off-line samples would be used.

3. Vocabulary Similarity

Word recognition algorithms utilize the dictionary directly in the recognition process, effectively employing word-level context in processing image data. Representations for words from a dictionary are matched to word images in documents. The result is a ranking of the dictionary for each word image where words that are ranked closer to the top have a higher probability of being correct. A consideration in using word recognition is that a large dictionary (on the order of 100,000 or more words) may be needed to guarantee that almost any word that could be encountered in an input document would exist in the dictionary.

Errors in the output of a word recognition system can be caused by several sources. When a noisy document image is input, the top choice of a word recognition system may only be correct a relatively small proportion of the time.

However, the ranking of the dictionary may include the correct choice among its top N guesses (N=10, for example) in nearly 100 percent of the cases.

An observation about context beyond the individual word level that is used here concerns the vocabulary of a document. Even though the vocabulary over which word recognition is computed may contain 100,000 or more words, a typical document may actually use fewer than 500 different words. Thus, higher accuracy in word recognition is bound to result if the vocabulary of a document could be predicted and the decisions of a word recognition algorithm were selected only from that limited set.

A technique has recently been proposed in which the N best recognition choices for each word are used in a probabilistic model for information retrieval to locate a set of similar documents in a database [15]. The vocabulary of those documents is then used to select the recognition decisions from the word recognition system that have a high probability of correctness. A useful side effect of matching word recognition results to documents from a database is that the topic of the input document is indicated by the titles of the matching documents from the database.

3.1. Algorithm Description

The algorithmic framework discussed in this paper is presented in Figure 3. Word images from a document are input. Those images are passed to a word recognition algorithm that matches them to entries in a large dictionary. *Neighborhoods* or groups of words from the dictionary are computed for each input image. The neighborhoods contain words that are *visually* similar to the input word images.

A matching algorithm is then executed on the word recognition neighborhoods. A subset of the documents in a pre-classified database of ASCII text samples are located that have similar topics to the input document. The hypothesis is that those documents should also share a significant portion of their vocabulary with the input document.

Entries in the neighborhoods are selected based on their appearance in the matching documents. The output of the algorithm are words that have an improved probability of being correct based on their joint appearance in both the word recognition neighborhoods as well as the matching documents. These are words that are both visually similar to the input and are in the vocabulary of the documents with similar topics.

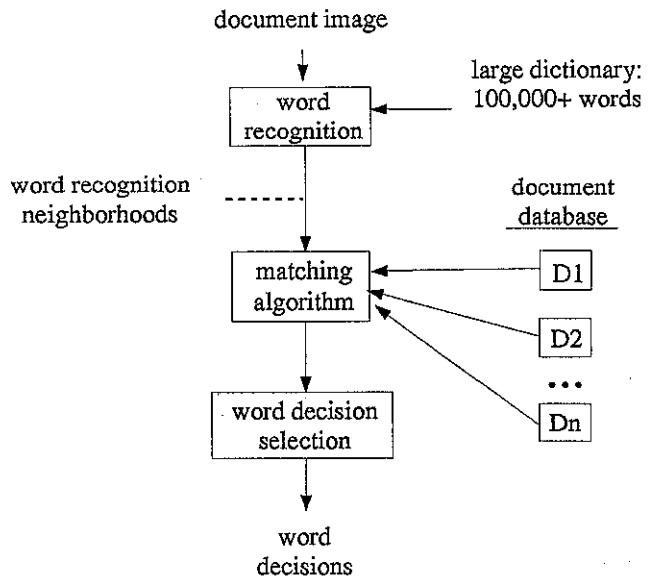

Figure 3. Vocabulary matching algorithm.

3.2. Document Matching Algorithm

The matching algorithm that determines the documents in the database that are most similar to the input is based on the *vector space model* for information retrieval [23]. In this approach, a document is represented by a vector of index terms or keywords. The similarity between a query and a document or between two documents is calculated by the inner product of the term vectors where each entry in the vectors is equal to one if the corresponding term is present and zero otherwise.

An alternate formulation is to replace the binary values by weights that represent the importance of the corresponding word or phrase in representing the content of the document. One method for calculating the weight assigned to word k in document i is:

$$w_{ik} = \frac{tf_{ik} \log\left(\frac{N}{n_k}\right)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [\log\left(\frac{N}{n_k}\right)]^2}} \quad (1)$$

where, tf_{ik} is the frequency of term k in document i, n_k is the number of documents in the database that contain term k, and N is the total number of documents in the database.

This formulation assigns higher weights to terms that occur frequently inside a document but less frequently in other documents. This is based on the assumption that those terms should be more important for representing the content of the document. Thus, the co-occurrence of terms with high weights in two documents should indicate that their topics are similar.

The direct use of the weight calculation expressed in equation (1) would assume the presence of a word recognition system with 100 percent accuracy. A modification is proposed in calculating the term frequency for a word in an input document that accounts for the imprecision in the recognition results. Instead of accumulating a unit weight for each occurrence of a word, the term frequency for a word is taken as the sum of the probabilities assigned to that word by the recognition algorithm.

The calculation of the external frequency of words in other documents in the database is unchanged since their true ASCII representation exists.

3.3. Similarity Calculation

The similarity between two documents i and j, as mentioned above, can be calculated as the inner product between their weight vectors:

$$sim(D_i, D_j) = \sum_{k=1}^{t} w_{ik} \cdot w_{jk}$$

for the t index terms that occur in either one or both of the documents.

In the application discussed here, the index terms in an ASCII document are calculated from the non-stop words and non-proper nouns. Stop words occur frequently in a normal text passage and convey little meaning. Proper nouns are

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

names of specific persons, places, or things. Every other word is assumed to be an "index term" for the purpose of matching.

3.4. Experimental Results

The word decision selection algorithm discussed above was demonstrated on the Brown corpus [18]. The Brown corpus is a collection of over one million words of running text that is divided into 500 samples of approximately 2000 words each. The samples were selected from 15 subject categories or genres and the number of samples in each genre was set to be representative of the amount of text in that subject area at the time the corpus was compiled.

One of the samples in the Brown corpus was selected as a test document to demonstrate the algorithm discussed above. This sample is denoted G02 (the second sample from genre G: *Belles Lettres*) and is an article entitled *Toward a Concept of National Responsibility,* by Arthur S. Miller that appeared in the December, 1961 edition of the Yale Review.

There are 2047 words in the running text of G02. After removing stop words and proper nouns, there were 885 words left. Neighborhoods were generated for those words by a simulation of a word recognition algorithm that used the 53,000 unique words from the entire Brown corpus as its dictionary.

The ten most visually similar dictionary words were calculated for each input word. This provided 8850 neighbors overall. The word shape calculation had performance of 87 percent correct in the top choice and 99 percent correct in the top ten choices.

The training data for the matching process and the word decision selection algorithm was the other 499 samples in the Brown corpus besides G02. The document matching algorithm described earlier was used to rank the other 499 samples for their similarity to G02.

The ten most similar samples in the Brown corpus, as determined by the matching algorithm, are listed in Table 1. It is interesting to observe how similar their titles are to that of G02. For example, the most similar sample is J42 whose title is *The Political Foundation of International Law.* This group of similar articles illustrates the side-effect of the matching algorithm since it essentially classifies the content of a document by indicating the samples that it is most similar to. The effectiveness of the document classification task could be improved by applying further preprocessing to the text samples in the database. More detailed representations of the database documents could be used in a more complex classification algorithm.

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

Table 1. Ten most similar samples to G02

rank	sample	title
1	J42	The Political Foundation of International Law
2	J22	The Emerging Nations
3	G25	The Restoration of Tradition
4	H02	An Act for International Development
5	H20	Development Program for the National Forests
6	G72	For a Concert of Free Nations
7	G35	Peace with Justice
8	H22	U.S. Treaties and Other International Agreements
9	H19	Peace Corps Fact Book
10	G10	How the Civil War Kept you Sovereign

3.5. Word Decision Selection Results

The ability of the most similar samples determined by the matching procedure to select the correct word decisions from the neighborhoods was tested. The top choices of the recognition algorithm were filtered by comparing them to the most similar samples and retaining the words that occurred in those samples. The three selection criteria that were tested included *overall* performance in which all the top recognition choices in G02 that occurred anywhere in the similar samples were retained.

The *G02-nouns* condition refers to the case where only the top choices for the nouns in G02 that matched any of the nouns in the similar samples were retained. The application of this selection criteria in a working system would assume the presence of a part-of-speech (POS) tagging algorithm that would assign POS tags to word images.

In the *matching-nouns* condition, only the nouns in the similar samples were used to filter the top recognition choices. This case was explored because the nouns may be considered to carry more information about the content of a text passage than verbs or words with other parts of speech. Thus, the co-occurrence of nouns in two documents about similar topics should be due less to chance than other word types.

The results of word decision selection when applied to the original word recognition output (with 13% error at the top choice) are summarized in Table 2. When all the words in the most similar sample (J42) were matched to the top recognition decisions for G02 (top left entry in Table 2), it was discovered that 251 of those top decisions also occurred in J42. Of those, only nine words were erroneous matches. This corresponds to an error rate of about four percent. In other words, the correct rate for 28 percent of the input words was raised to 96 percent from the 87 percent provided by the word recognition algorithm alone.

The other results show that as more of the similar samples are used to filter the word recognition output, a progressively higher percentage of the eligible neighborhoods are included and the correct rate remains stable. For example, in the *overall* condition using the four most similar samples, 441 of the 885 (50%) input words were effectively recognized with a correct rate of 97.2 percent. The results for the *G02-nouns* matching condition show that up to 26 percent of the input can be recognized with a 99.16 percent correct rate. In the *nouns-matching* condition, 29 percent of the input words can be recognized with a 97 percent correct rate.

Table 2. Word selection performance on the 885 neighborhoods.

samples used	decision selection criteria								
	overall			*G02-nouns*			*nouns-matching*		
	matches	errs	corr.%	matches	errs	corr.%	matches	errs	corr.%
1	251	9	96	130	2	98	187	6	97
2	345	11	97	177	2	99	206	6	97
3	393	12	97	199	2	99	241	6	98
4	441	12	97	229	2	99	257	8	97
5	451	12	98	234	2	99	258	9	97
6	459	13	97	248	2	99	272	9	96
7	474	16	97	254	3	99	280	11	96
8	483	16	97	254	3	99	284	11	96
9	498	16	97	261	3	99	288	11	96
10	526	22	96	300	4	99	296	12	96

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

4. Statistical Constraints

The statistical transitions between pairs of words are one source of syntactic information that have been used to improve word recognition [10]. Alternatives for the identification of a word were removed from consideration if they never followed the previous word in a large training sample of text. Even though this technique improved performance, it was computationally unacceptable as the transitions were difficult to estimate with even very large samples of text.

An improvement on word-to-word transitions was to model language syntax with binary constraints between a group of words with the same shape and the syntactic classes that could follow them [11]. The constraints were compiled from a training text and applied to restrict the decisions for the syntactic class of a word to be consistent with the shape of the previous word. Even this limited binary information was shown to be effective at reducing the average number of words that could match any image by about 16 percent on average with an error rate of about one percent. An error occurred when a word was erroneously removed from consideration.

This section discusses an algorithm that models English grammar as a Markov process where the probability of observing any syntactic category is dependent on the syntactic category of the previous word or words [14]. This model is applied to text recognition by first using a word recognition algorithm to supply a number of alternatives for the identity of each word. The syntactic categories of the alternatives for the words in a sentence are then input to a modified Viterbi algorithm that determines the sequences of syntactic categories that best match the input. An alternative for a word decision is output only if its syntactic category is included in at least one of these sequences. The Markov model improves word recognition performance if the number of alternatives for a word are reduced without removing the correct choice.

4.1. Syntax Model

The syntax of a sentence is summarized as the sequence of grammatical categories for each of its words. There are several ways to define the grammatical categories. The part-of-speech (POS) tags assigned to each word are one definition. For example, in the sentence "He was at work.", *He* is a pronoun, *was* is a verb, *at* is a preposition, and *work* is a noun.

Since the appearance of a grammatical category probabilistically constrains the categories that can follow it, a Markov model is a natural representation for syntax [19]. An example of such a constraint are the probabilities that certain

POS tags follow an article in a large sample of text. The word following an article is a singular or mass noun in 51 percent of all cases and is an adjective 20 percent of the time. The other 29 percent of occurrences are scattered over 82 other syntactic classes [7].

A hidden Markov model (HMM) can be specified that links the recognition process described earlier and a Markov model for language syntax [21]. The grammatical categories in the English language are assumed to be the N *states* of a discrete r^{th} order Markov process. The states are defined to be POS tags. Many words can be assigned one POS tag and a relatively small number (typically 25 to 100) of POS tags have been used for English.

In the word recognition algorithm, the states are "hidden" because they are not observable at run-time. Rather, the feature vector that describes a word image is the observable event. The number of such feature vectors is finite and provides a fixed number of *observation symbols*.

The transition from one state to another is described by a *state transition probability distribution*. If the Markov process is assumed to be first order, this distribution can be given by an $N \times N$ matrix. A second order assumption would imply the use of an $N \times N \times N$ probability distribution matrix.

There is also a probabilistic mapping function from the set of observations onto the set of states. Each observation is first mapped onto a set of words by the neighborhood generation or word recognition algorithm. Each word is assigned a probability of correctness by this process. Those words are then mapped onto the set of POS tags by a many-to-one function. The probability of an observation given that the process is in a specific state is provided by a combination of the recognition and word-to-state mapping probabilities. This is sometimes referred to as the *confusion probability*.

There are also *initial and final state distributions* that specify the probability that the model is in each state for the first and last words in a sentence. These constraints can be powerful. For example, it has been observed in a sample of newspaper reportage that the first word in a sentence is an article or a proper noun with probability 0.31 and 0.14, respectively. The other 55 percent is divided among 24 other classes.

The HMM is completely specified by the five elements just described (states, observation symbols, state transition probabilities, observation symbol probabilities, and initial probabilities). The HMM is applied to word recognition by estimating the sequence of states with the maximum a-posterior probability of occurrence for a given sequence of observations (feature vectors). The performance of the word recognition algorithm is improved by reducing neighborhoods so that they contain only words that map onto states in the estimated sequence.

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

The estimation of the sequence of states with the maximum a-posterior probability of occurrence is efficiently performed by the Viterbi algorithm [6]. The adaptation of the Viterbi algorithm to this problem is similar to its use in post-processing character decisions [9]. The Viterbi algorithm has also been successfully used for speech recognition [2].

4.2. Example

An example of applying the HMM for syntactic constraints is shown in Figure 4. The original input sentence is shown along the the top of the figure (HE WAS AT WORK.). The complete neighborhoods for each word are shown below the input words. Each word is shown along with an indication of its syntactic class and the confusion probability for that neighborhood given the syntactic class. The first and third words in the sentence have only one word in their neighborhoods. The second neighborhood contains eight different words, two of which have different syntactic classes. The fourth neighborhood contains six different words, one of which has three different syntactic tags.

The transition probabilities are shown along the arcs. It is seen that some transitions, such as PPS-NNS (third person nominative pronoun followed by a plural noun) never occurred in the training text and hence have a probability of zero. Other transitions are much more likely, such as PPS-VBD (third person nominative pronoun followed by a verb in the past tense, e.g., *he ran*) which has a probability of 0.3621.

In this case, the top choice of the Viterbi algorithm was PPS-BEDZ-IN-NP and the second choice was PPS-BEDZ-IN-NN. In the top choice, three of the four classes included the correct identification for each word. The correct answer for the fourth word (NN) was only contained in the second choice of the Viterbi.

4.3. Experimental Investigation

Experimental tests were conducted to determine the ability of the HMM to reduce the number of word candidates that match any image. Given sentences from test samples of running text, a neighborhood was calculated for each word by a simulation of the word recognition process. These data were then processed by the HMM.

Performance was measured by calculating the average neighborhood size per text word before and after the application of syntax. This statistic is defined as:

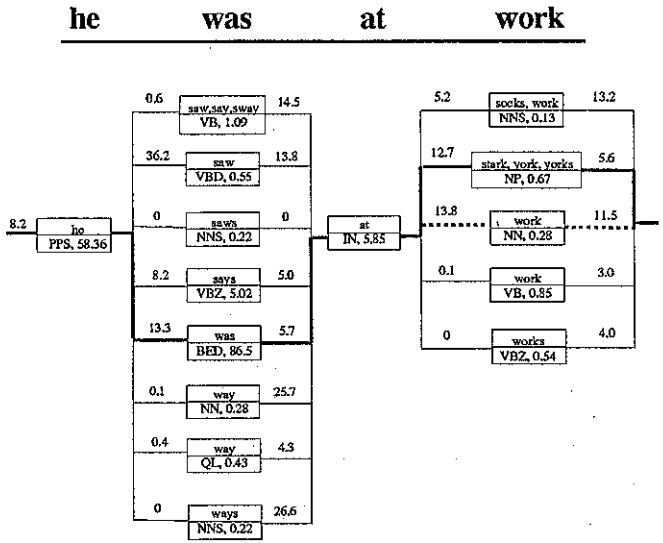

Figure 4. Example of applying the Viterbi algorithm .

$$ANS_t = \frac{1}{N_w} \sum_{i=1}^{N_w} ns_i$$

where N_w is the number of words in the test sample and ns_i is the number of words in the neighborhood for the i^{th} word in the text. The improvement in performance is measured by the percentage reduction in ANS_t. The *error rate* was also measured. This is the percentage of words with neighborhoods that do not contain the correct choice after the application of syntax.

In the example presented in Figure 4, N_w is 4, $ns_1 = 1$, $ns_2 = 8$, $ns_3 = 1$, and $ns_4 = 6$. $ANS_t = \frac{1}{4}$ (1+8+1+6) = 4.0. After the application of both parses, ANS_t was reduced to 1.75. This is an overall reduction of about 44 percent in ANS_t with a zero percent error rate.

4.4. Experimental Results

The HMM was applied to correct the text recognition results for the neighborhoods generated from sample A06. A06 is a sample of newspaper reportage that contains excerpts from the *Newark Evening News* of March 22, 1961.

Five state-sequences were output for each sentence by the HMM and used to filter the neighborhoods. Both first and second-order syntactic class transition probabilities were estimated from the remainder of the corpus.

Results showed that the neighborhoods could be reduced by between 70 and 80 percent with an error rate that ranged from one to five percent. The best performance was achieved with five state sequences and first order transition probabilities. Interestingly enough, using second order transitions had little effect on performance. This might be attributed to a lack of training data.

5. Structural Syntactic Constraints

Structural techniques that produce a complete parse (or a number of parses) for a sentence have been used to improve the performance of speech recognition devices. Typically, such methods require that a grammar and lexicon be supplied that summarize the syntax of the sentences of the language that will be encountered. An input sentence is then parsed and word decisions are output that appear as terminal nodes in the parse tree(s).

Difficulties with such techniques include the need to specify a grammar that will cover all the instances of a language that will be encountered in practice. Recently, statistical and structural approaches have been combined to overcome some of the difficulties in extending parsing techniques to unrestricted text. Automatic acquisition of lexical knowledge also has become a promising approach for building the large-scale dictionaries needed by such techniques.

Recently, parsing techniques have been applied to correct the output of a word-based text recognition system [8]. A probabilistic lattice was described that used syntactic and semantic constraints to find the best candidate for each word image.

Two types of linguistic constraint were used. One is local word collocation in which the identity of a word is used to predict the identity of other nearby words [4]. Global structural constraints are exploited with a chart parsing

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

model. A lattice parser was used that allows for several word candidates at the same position, and therefore can be directly applied to correcting the neighborhoods output by a word recognition algorithm. The parser chooses the words on a path through the lattice that correspond to a legal sentence with the highest probability of being correct, given the sentences represented in the lattice.

Given the word images from an English sentence, a word recognition algorithm generates a neighborhood for each word image; next, a relaxation procedure reduces the top-n candidates for each image to the two best candidates by applying word collocation information; the top-2 lists for a sentence form the word lattice processed by the probabilistic lattice chart parser. All possible parse trees are built from the reduced word lattice; finally, the word candidates involved in the most preferred parse tree are selected as the correct word candidates and the most preferred parse tree is output.

5.1. Word Collocation Data

A relaxation algorithm based on word collocation statistics is used as a filter to reduce the top-n word candidates at each location to the into top-m, where $m<n$. The basic idea of the relaxation algorithm is to use local word collocation constraints to select the word candidates that have stronger word collocation with their neighbor words.

Let W_i denote the word image at position i and w_{ij} denote the jth word candidate for the word image W_i, where $1 \le j \le n$. The word collocation score of w_{ij}, where $1 \le j \le n$, is

$$\max_{k=1}^{n} WordCollocationScoreOf\ (w_{i-1,k}\ ,\ w_{ij})$$

$$+\ \max_{k=1}^{n} WordCollocationScoreOf\ (w_{ij}\ ,\ w_{i+1,k})$$

where $WordCollocationScoreOf\ (word_1\ ,\ word_2)$ is the measurement of the strength of word collocation of the word pair $(word_1, word_2)$. After sorting the word collocation scores of w_{ij}, for $j=1, \cdots, n$, the first m word candidates with the highest word collocation scores are output.

5.2. Probabilistic Lattice Chart Parser

There are 668 rules in the CFG used in this system. For example, the following rules are typical.


```
S       <- S-BODY .
S       <- WH-S-BODY ?
S       <- YN-S-BODY ?
S-BODY  <- NP VP
S-BODY  <- S-BODY CC S-BODY
NP      <- NN
NP      <- NP PP
NP      <- NP CC NP
VP      <- VB NP
VP      <- VB
```

Each of the rules in the grammar is associated with a confidence score which indicates the priority of the rule in comparison to other rules. A rule with a high score means that it is more likely to be used.

The parser is a bottom-up chart parser. A chart is a graph, that is a set of nodes and a set of edges linking them. As a data structure, the chart provides a general, flexible, efficient and economic framework for parsing. Every constituent or structure derived during parsing is recorded as an edge.

There are five steps in the parser. They are:

1. loading a word sequence or word lattice;
2. loading word tags;
3. noun phrase parsing;
4. sentence parsing;
5. checking the chart to find the most preferred parse tree if it exists.

When parsing is finished, the parser decides whether it failed by checking whether there exists any edges with label "S". If parsing succeeds, it will print out the parse tree represented by the "S" edge with the highest confidence score. If parsing fails, parse trees for fragments of the input word sequence are output.

5.3. Experimental Results

The chart parsing model was applied to sentences from sample G02 of the Brown corpus. A model for word recognition was used to generate neighborhoods for the words in the sample. Each neighborhood contained ten words on average and no information about the confidence value for the recognition decisions was used. The overall correct rate for relaxation was 86.4% and the parser correctly selected 76.2% of the words. This is a significant improvement over the ten percent correct rate that would be obtained from the neighborhoods.

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

6. Structural Semantic Constraints

The underlying meaning of text has been utilized in an innovative way to improve performance in a restricted domain by allowing only alternatives that were sensible in context [1]. The use of definitional overlap in machine-readable dictionaries and word collocations as semantic representations to improve the performance of a handwriting recognition system has also been discussed [5, 22]

This section discusses a technique that models the semantic relationships between words by the network of connections in a thesaurus [13]. The thesaurus is represented by a directed graph in which root nodes point to related words. These words can also be root words that in turn point to other words, and so on. This structural representation is applied to text recognition by first using a word recognition algorithm to supply a number of alternatives for the identity of each word. The alternatives for the words in a passage of text are then recursively looked up in the thesaurus. Counters associated with each thesaurus word are incremented when the words are encountered in the lookup process. Semantically related alternatives in the neighborhoods from a passage of text are thus reinforced and unrelated words are suppressed. A threshold is applied to the scores to remove unrelated words from neighborhoods.

6.1. Semantic Model: The Thesaurus

The thesaurus used in this work was a directed graph of words that was commercially available. The thesaurus contained entries for 25,072 words. Each entry provides a root word and a list of related words. A minimum of two, a maximum of 257, and an average of 49.2 related words are in each entry. There are 50,357 different non-root words in the thesaurus.

The potential usefulness of the thesaurus in text recognition is illustrated by looking up all the words in a sample of text, accumulating the counts, and observing their distribution of values. The result of applying this process with two levels of lookup to 2000 words of a story about golfing[1] is illustrated in Table 3. The top 40 scores and the associated words are shown. It should be noted that no scores were accumulated for the true words. Thus any natural bias is eliminated and the pure effect of the thesaurus is given.

[1] Alfred Wright, "A duel golfers will never forget," Sports Illustrated, 64, pp. 18-21, April 17, 1961.

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

Table 3. Thesaurus activation values after two levels of lookup

COURSE	3423	ROUND	3187	CUT	2445	LINE	2427
RANGE	2360	HOLE	2330	POINT	2121	SET	1990
TURN	1985	CRACK	1885	PLACE	1862	CHECK	1794
SPACE	1790	MEASURE	1708	MARK	1687	PERIOD	1625
SCALE	1608	STEP	1607	ORDER	1571	FIELD	1561
STAGE	1548	RANK	1526	BEAT	1493	LEVEL	1393
OPENING	1383	NOTCH	1374	CYCLE	1350	PIT	1348
SHADOW	1334	RACE	1318	BANK	1304	GROUND	1304
DESCENT	1292	STRING	1290	PASSAGE	1274	CIRCUIT	1260
INTERVAL	1239	GALLERY	1237	COVER	1227	WAY	1224

It is very interesting to observe that key terms about golf are very frequent. For example, *golf* COURSE, ROUND *of golf*, *make the* CUT, *driving* RANGE, HOLE *in one*, are all near the top of the ranking. Numerous other similarities are also observed.

6.2. Algorithm

A statement of the algorithm for applying the thesaurus constraint to filtering neighborhoods output by the word recognition process is below:

1. Initialize counters for every thesaurus word (50,357)
2. For every word in every neighborhood:
   Recursively increment activation counters for lookup(word).
3. Rank the neighborhoods of content words (nouns) by:
   Confidence(word) = counter(word) / sum of counters in neighborhood
4. Reduce neighborhoods by thresholding the confidence value.

The function lookup(word) returns the list of related words from the thesaurus.

6.3. Experimental Results

The thesaurus algorithm was applied to correct simulated text recognition results from subset A38 of the Brown corpus. Two levels of thesaurus lookup were performed and the lexicon for the entire corpus was used. Using one level of lookup in the thesaurus, a neighborhood size reduction of about ten percent was achieved with an error rate of less than half a percent. Using a simulation in which the input neighborhoods contained approximately 15 words, a reduction of 14.82 percent was achieved with an error rate of 1.54 percent. When two levels of lookup were used, this was improved to a 19.59 percent reduction with a 0.77 percent error rate.

7. Discussion and Conclusions

This paper discussed several classes of technique that utilize language level syntactic and semantic constraints to improve the performance of a word recognition algorithm. The information used by such methods can generally be classified as graphical, vocabulary, statistical, structural-syntactic, and structural-semantic. Each area was discussed and algorithms that use each type of information were presented.

Acknowledgments. Siamak Khoubyari, Yanhong Li, and Tao Hong contributed to the preparation of this paper.

References

1. Baird, H. S. and Thompson, K., "Reading Chess," *IEEE Transactions on Pattern Analysis and Machine Intelligence 12* (1990), 552-559.
2. Cherkassky, V., Rao, M., Weschler, H., Bahl, L. R., Jelinek, F. and Mercer, R. L., "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5*, 2 (March, 1983), 179-190.
3. Chien, L. F., Chen, K. J. and Lee, L. S., "An augmented chart data structure with efficient word lattice parsing scheme in speech recognition applications," *13th International Conference on Computational Linguistics 2* (1990), 60-65.
4. Church, K., Gale, W., Hanks, P. and Hindle, D., "Parsing, word associations, and typical predicate-argument relations," *International Workshop on Parsing Technologies*, Pittsburgh, Pennsylvania, August 28-31, 1989, 389-398.

5. Evett, L. J., Wells, C. J., Keenan, F. G., Rose, T. and Whitrow, R. J., "Using linguistic information to aid handwriting recognition," in *From Pixels to Features III: Frontiers in Handwriting Recognition*, S. Impedovo and J. C. Simon (editor), Elsevier Science Publishers B.V., 1992.
6. Forney, G. D., "The Viterbi algorithm," *Proceedings of the IEEE 61*, 3 (March, 1973), 268-278.
7. Francis, W. N. and Kucera, H., *Frequency Analysis of English Usage: Lexicon and Grammar*, Houghton Mifflin, Co., Boston, Massachusetts, 1982.
8. Hong, T. and Hull, J. J., "A Probabilistic Lattice Chart Parser for Text Recognition," *ICDAR-93: Second IAPR Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, October 20-22, 1993. submitted.
9. Hull, J. J., Srihari, S. N. and Choudhari, R., "An integrated algorithm for text recognition: comparison with a cascaded algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5*, 4 (July, 1983), 384-395.
10. Hull, J. J., "Inter-word constraints in visual word recognition," *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence*, Montreal, Canada, May 21-23, 1986, 134-138.
11. Hull, J. J., "Feature selection and language syntax in text recognition," in *From Pixels to Features*, J. C. Simon (editor), North Holland, 1989, 249-260.
12. Hull, J. J., Ho, T. K., Favata, J., Govindaraju, V. and Srihari, S. N., "Combination of segmentation-based and wholistic handwritten word recognition algorithms," *From Pixels to Features III: International Workshop on Frontiers in Handwriting Recognition*, Bonas, France, September 23-27, 1991, 229-240.
13. Hull, J. J. and Chin, A. T., "Semantic information extraction with a thesaurus for visual word recognition," in *Advances in Structual and Syntactic Pattern Recognition*, H. Bunke (editor), World Scientific, 1992, 342-351.
14. Hull, J. J., "A hidden Markov model for language syntax in text reconition," *11th IAPR International Conference on Pattern Recognition*, The Hague, The Netherlands, August 30 - September 3, 1992, 124-127.
15. Hull, J. J. and Li, Y., "Word recognition result interpretation using the vector space model for information retrieval," *Second Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, April 26-28, 1993.
16. Jones, M. A., Story, G. A. and Ballard, B. W., "Integrating multiple knowledge sources in a Bayesian OCR post-processor," *First International Conference on Document Analysis and Recognition*, Saint-Malo, France, September 30 - October 2, 1991, 925-933.
17. Khoubyari, S. and Hull, J. J., "Keyword location in noisy document images," *Second Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, April 26-28, 1993.
18. Kucera, H. and Francis, W. N., *Computational analysis of present-day American English*, Brown University Press, Providence, Rhode Island, 1967.

19. Kuhn, R., "Speech recognition and the frequency of recently used words: A modified Markov model for natural language," *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, Hungary, August 22-27, 1988, 348-350.

20. Nagy, G., Seth, S. and Einspahr, K., "Decoding substitution ciphers by means of word matching with application to OCR," *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, 5 (September, 1987), 710-715.

21. Rabiner, L. R. and Huang, B. H., "An introduction to hidden Markov model," *ASSP Magazine 3*, 1 (1986), 4-16.

22. Rose, T. G., Evett, L. J. and Whitrow, R. J., "The use of semantic information as an aid to handwriting recognition," *First International Conference on Document Analysis and Recognition*, Saint-Malo, France, September 30 - October 2, 1991, 629-637.

23. Salton, G., *Automatic text processing*, Addison Wesley, 1988.

24. Salton, G., "Developments in automatic information retrieval," *Science 253* (1991), 974-980.

25. Seneff, S., "Probabilistic parsing for spoken language applications," *International Workshop on Parsing Technologies*, Pittsburgh, Pennsylvania, August 28-31, 1989, 209-218.

26. Tomita, M., "An efficient word lattice parsing algorithm for continuous speech recognition," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1986, 1569-1572.

27. White, G. M., "Natural language understanding and speech recognition," *Communications of the ACM 33*, 8 (August, 1990), 74-82.

Fundamentals in Handwriting Recognition (S. Impedovo, editor), Springer Verlag, 1994.

Verification of Handwritten British Postcodes Using Address Features

Hendrawan and A.C. Downton

Department of Electronic Systems Engineering, Univeristy of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

1 Introduction

The British postcode system consists of between five and seven alphanumeric characters which can specify any geographical postal location in the United Kingdom down to the resolution of a single office block or a group of several residential houses in a street. It is therefore possible to attempt to automatically sort handwritten letter mail by locating and recognising the handwritten characters of the postcode. However, OCR of handwritten characters is not perfect and misrecognitions will occur resulting in an unacceptable error rate in the sorting performance.

A verification process based upon correlating the postcode against the remainder of the handwritten or hand-printed address can reduce the error rate. In practical applications of automatic mail sorting, a low error rate is essential, although a higher rejection rate can be tolerated.

In this paper a verification method based on the concept of fuzzy sets is proposed. The verification is carried out by first defining a fuzzy membership function to measure the word similarities between an address image and the reference address corresponding to the recognised postcode. Degree of word matches are derived by measuring positional and structural feature similarities. The degree of similarity of the whole handwritten address is then obtained by finding the aggregation value of optimal word matches derived using the assignment problem in linear programming.

Postcode recognition and generation of the reference address corresponding to the recognised postcode are described in [1,2].

2 Preliminary Processing

In this work the handwritten addresses are partially constrained in that boxes are provided to write the postcode and horizontal guidelines are provided for guiding the position of the rest of the address. Therefore, whilst location and separation of the individual postcode characters can be achieved using a fairly simple algorithm, segmentation of the rest of the address is not trivial. The address images were scanned using the front end of an AEG postal sorter producing binary images of 512 x 2048 pixels at a resolution of a little better than 200 pixels per inch.