INTER-WORD CONSTRAINTS IN VISUAL WORD RECOGNITION

Jonathan J. Hull

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

ABSTRACT

A method of using knowledge about constraints *between* words in a technique for reading a running text is presented. This knowledge is represented as a set of allowable transitions (called "inter-word constraints") and a method is given for incorporating this knowledge representation in a system for reading visual images of text. This system first analyses the shape of a word image to suggest a group or *neighborhood* of words in a dictionary (list of words) that contains an input word. The inter-word constraints are then used to reduce the size of the neighborhood and the smaller neighborhood is used to direct further detailed analysis of the input. This process results in a match of the input image to one of the words in the neighborhood. Results are reported in this paper on the performance and cost of two representations for inter-word constraints. The potential of these knowledge sources to reduce the neighborhood size is explored in a series of statistical experiments. It is shown that the average size of a neighborhood encountered when a text of 150,000 words is "read" on a word by word basis is reduced from 16 to about 2. It is also shown empirically that the memory needed for both knowledge representations grows linearly with dictionary size and the total additional memory requirement is about twice that needed for the original dictionary.

1. INTRODUCTION

The development of computer reading algorithms that possess the same versatility as a human reader is a long standing goal of artificial intelligence research. One approach that has been followed in the pursuit of this goal is to determine how humans read and to convert useful portions of this knowledge into algorithms. This method was followed by Shillman[10] in his work on isolated character recognition. He determined the parameters of the human character recognition process and developed algorithms that used these parameters.

The complete, fluent human reading process involves much more than just isolated character recognition. A human reader routinely employs knowledge about the syntactic, semantic, and featural dependencies between words [1,8]. Although the use of this type of knowledge source in an algorithm for reading text has been suggested [9], no such technique is known.

The use of featural dependencies between words (referred to as inter-word constraints) in an algorithm for reading text is explored in this paper. This knowledge is represented in two ways: either as a table of allowable transitions between words, or as a table of allowable transitions between the featural description of a group of words (called a *neighborhood*) and another word. Both these representations capture a different form of featural dependency. The word-to-word transitions represent knowledge of the words that can follow a given word under the assumption that the given word can be accurately recognized. The neighborhood-to-word transitions represent knowledge about legal successor words under the assumption that the given predecessor word can be approximately recognized. Word-to-word transitions represent the type of knowledge people might use when reading a text for detailed content (e.g. journal article). Neighborhood-to-word transitions are a less specific representation that corresponds to the way people might use featural dependencies in a relaxed reading situation, such as when reading a newspaper.

2. READING ALGORITHM

An outline of the reading algorithm used to evaluate the potential of inter-word constraints is presented in Figure 1. This algorithm assumes that its input is a sequence of word images W_i, $i = 1,2,\ldots$ that make up an input text. These images are input to a **word shape computation** routine that uses a few gross features of a word to determine a neighborhood of words (N_i) in a dictionary (a list of words that could occur in a text) that share that feature set. This neighborhood is then input to the **inter-word constraints analysis routine** that uses either the identity of the previous word (W_{i-1}) or the neighborhood of the previous word (N_{i-1}) along with an internal table of allowable transitions between pairs of words or between neighborhoods and words to produce a reduced neighborhood N_i'. The **hypothesis verification** routine uses the contents of N_i' to direct further selective analysis of the input image, resulting in a recognition decision. The hypothesis verification process is discussed in [5] and will not be extensively analyzed in this paper. Rather, the focus will be on the usefulness of inter-word constraints in reducing the number of words in a neighborhood, thereby simplifying the process of hypothesis verification.

The word shape computation procedure used here (more fully described in [4]) is designed as a generator of candidate words rather than as a complete method for word recognition. This technique is based on the detection of the following six features in a word image:

0. A significant filled space at the beginning or end of a word (e.g., the filled space to the right of the short vertical part in a 'c');
1. A short vertical part (e.g., the leg of an 'r');
2. A long vertical part extending above the main part of a word (e.g., the ascender portion of a 'b');
3. A long vertical part extending below the main body of a word (e.g., the descender in a 'p';
4. Dots over short vertical parts (occurs in an 'i');
5. Dots over long vertical parts (occurs in a 'j');

(Note that these features are specialized for lower case text, however, the technique can be extended to mixed case and upper case input). These features were chosen because they exist in a large

Proc. of Canadian Soc. for Computational Studies of Intelligence, Montreal, Canada, 134-138, May 21-23, 1986.

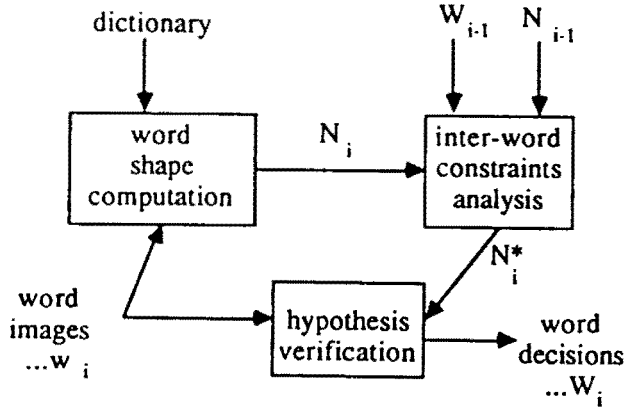

Figure 1. An outline of the word recognition algorithm.

(a) (b)

1. high-vert.
2. short-vert.
3. short-vert.
4. low-vert.
5. short-vert.
6. short-vert.

{"shape", "slope"}

(c) (d)

Figure 2. An example of word shape computation. (a). An input word; (b). The thresholded output of convolution with an 50x5 bar mask; (c). Description of the features detected in (b); (d). The neighborhood was determined from the dictionary of the Brown Corpus [7].

number of lower case fonts and it is easy to design algorithms to detect them. A neighborhood is computed by matching the left-to-right sequence of features that occur in an input word to a similar description of the words in the dictionary. Several classes of errors could be detected at this phase, including incorrect feature extraction and the absence of an input word in the dictionary. An example of the word shape and neighborhood computation is presented in Figure 2. The input word "shape" is shown in Figure 2(a), the result of a convolution operation used to detect vertical parts is shown in Figure 2(b), and a description of each component is shown in Figure 2(c). The neighborhood computed from the sequence of features ("02113110" as discussed above) determined from Figure 2(c) is shown in Figure 2(d). This neighborhood is { "shape", "slope" }.

Two procedures for using inter-word constraints to reduce the size of a neighborhood are explored next. Both of these methods use a table of allowable transitions to reduce the size of the neighborhood of an input word. This representation is somewhat analogous to the thresholded transition probabilities used in the binary n-gram technique for contextual postprocessing [2, 3]. In the first method for reducing the size of a neighborhood, transitions between two words are used. In the second method, transitions between neighborhoods and words are used. The discrete nature of this representation loses probabilistic information, however, it retains significant power, as will be seen.

These representations for inter-word constraints are used to remove words from the neighborhood of a word under the assumption that either the preceding word was perfectly recognized (for word-to-word transitions), or that the neighborhood of the preceding word was correctly computed (for neighborhood-to-word transitions). The use of word-to-word transitions has the advantage of increased accuracy while the use of neighborhood-to-word transitions is insensitive to errors in hypothesis verification that may have been committed on the preceding word. Figure 3. shows a portion of the data structure needed for both these techniques. If word-to-word transitions are used, a word dictionary is augmented by pointers from each word to all the words that can follow it. If neighborhood-to-word transitions are used, a neighborhood dictionary is provided that includes neighborhood-to-word pointers to entries in the word dictionary. For example, if word-to-word transitions were being used when the word "shape" was being recognized in the phrase "... a visual shape ...", and only "shape" can follow "visual" in the corpus, the neighborhood of "shape", (which is {"shape", "slope"}) could be reduced to just "shape" (a perfect recognition). If neighborhood-to-word transitions were employed and both "shape" and "slope" could follow "041112" (the shape number of "visual"), the neighborhood of "shape" would remain {"shape", "slope"}.

3. STATISTICAL EFFECTS OF CONSTRAINTS

The projected performance of both forms of inter-word constraint can be determined by their effect on the number of words that are expected to occur in the neighborhood of a word image. This is measured from a large text by two statistics, the percentage of text that is uniquely specified by shape (PER_UNIQ), i.e., the percentage of a given text that has a neighborhood containing a single word, and the average neighborhood size per text word (ANS_t). Given that $ns(tw_i)$ returns the number of words in the

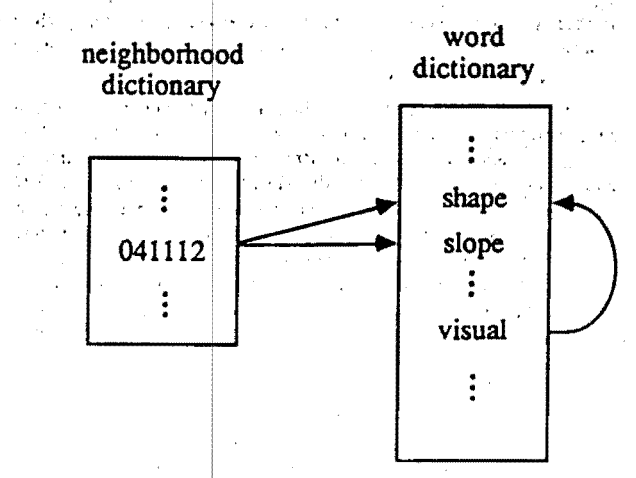

Figure 3. Example of data structures for two forms of inter-word constraint. If word-to-word constraints are used, only the word dictionary and the word-to-word pointers are needed. If neighborhood-to-word constraints are used, the neighborhhod dictionary, the neighborhood-to-word pointers, and the word dictionary are needed.

Proc. of Canadian Soc. for Computational Studies of Intelligence, Montreal, Canada, 134-138, May 21-23, 1986.

genre	N_d	N_t	% uniq. isolated	% uniq. nb-wrd	%uniq. wrd-wrd	ANS_t isolate	ANS_t nb-wrd	ANS_t wrd-wrd
A	11,743	88,051	19	71	78	14.1	1.8	1.6
B	8608	54,662	22	74	80	11.4	1.6	1.5
C	7724	35,466	24	82	86	9.7	1.4	1.3
D	5714	34,495	27	79	83	7.0	1.4	1.3
E	9803	72,529	21	73	80	12.8	1.7	1.5
F	12,622	97,658	19	70	78	14.4	1.9	1.7
G	16,126	152,662	15	68	76	16.0	2.1	1.8
H	6628	61,702	24	79	83	7.6	1.5	1.4
J	14,062	160,877	21	72	78	12.8	1.9	1.7
K	8490	58,650	18	70	77	11.9	1.8	1.5
L	6285	48,462	19	72	80	9.7	1.6	1.4
M	2996	12,127	32	85	89	5.7	1.2	1.2
N	8025	58,790	18	71	79	12.0	1.8	1.5
P	7664	59,014	19	71	79	11.9	1.8	1.5
R	4362	18,447	30	84	89	7.1	1.3	1.2
corpus	43,264	1,013,549	9	33	39	38.4	4.4	3.9

Table 1. Summary of the measures of potential usefulness of language context for lower case printed text. The results are broken down in terms of the genres as well as the entire text of the Brown Corpus. Each genre was processed using its dictionary. The allowable transitions were also determined from the genre itself. The percentages of the text uniquely specified by shape and the values of ANS_t are shown when words are considered in isolation, when neighborhood-to-word (nb-wrd) context is used, and when word-to-word (wrd-wrd) context is used. Note that N_d and N_t are the number of words in the dictionary and the text of the indicated genre, respectively.

neighborhood of the i^{th} word in a text,

$$ANS_t = \frac{1}{N_t} \sum_{i=1}^{N_t} ns(tw_i)$$

where N_t is the number of words in the given text. The average neighborhood size per text word thus measures the average number of words that would be presented to the hypothesis verification routine by the word shape computation procedure when it read the given text.

The potential influence of the inter-word constraints on the reading algorithm can be measured by their effect on PER_UNIQ and ANS_t. Since inter-word constraints are supposed to produce a smaller neighborhood, PER_UNIQ should be increased and ANS_t should be decreased as compared to when no inter-word constraints are used. Table 1 shows the results of a study conducted on a collection of over 1,000,000 words of text, known as the Brown Corpus[7], which is representative of contemporary American English. The corpus itself is divided into fifteen individual subject categories or genres. The PER_UNIQ and ANS_t figures were computed over each genre as well as the entire corpus under three constraints. The constraints were: no inter-word constraints, neighborhood-to-word transitions, and word-to-word transitions.

This study shows that there is a large expected improvement in performance when either form of inter-word constraint is employed. In some cases, up to 89% of text is uniquely specified by shape when word-to-word transitions are used and only 30% of the same text is uniquely specified when no inter-word constraints are used. Furthermore, the average neighborhood size per text word is reduced from 38.4 to 4.4 in the entire corpus and from 14.1 to 1.8 in an individual subcorpus when neighborhood-to-word transitions are used and from 38.4 to 3.9 in the entire corpus and from 14.1 to 1.6 in a subcorpus when word-to-word transitions are employed. The fact that these figures drop so dramatically indicates that a large portion of the texts could be recognized by only recognizing the six features discussed in section one and using either form of inter-word constraint. Furthermore, the lack of a significant difference between the two measures indicates that the more specific word-to-word context adds little to the statistical projections. This is satisfying from the reliability point of view since the gains provided by inter-word constraints are insensitive to the results of hypothesis verification. Therefore, it is theoretically possible to recognize a large portion of the texts used in this experiment with only six features and either of the inter-word constraints.

4. EXPERIMENTAL RESULTS

Two experiments were conducted to demonstrate the feasibility of this methodology. The first experiment shows that the shape of a word can be accurately computed under a variety of input conditions that include several different fonts and words with characters touching and overlapping. The second experiment explores the storage required for the transition tables for both representations of inter-word constraints. This is to answer the obvious questions about the practicality of these methods.

The first set of experiments tested the ability to reliably extract the features presented in section one of this paper. If an acceptable degree of reliability can be achieved, it is then reasonable to speculate on the applicability of this methodology to a more general purpose reading situation. The objective of these experiments was not to develop an algorithm for recognizing text in the limited number of fonts currently available, but rather to

show that with a few simple features that reflect font-independent characteristics, the neighborhoods of words printed in a variety of fonts can be reliably determined. The image database contained five complete 24 point font samples in five display faces (Americana, Baskerville Bold, Bembo, Bodoni Bold, and Bodoni Book) that were digitized on a laser scanner at a resolution of 500 binary pixels per inch. This data was manually segmented into characters and stored in individual files. Words were then generated by appending the appropriate character images.

Two methods were used for feature detection. Dots were determined by locating connected components in the top third of the image. Significant vertical parts were detected by convolution with a vertical bar mask. A response in the thresholded output was considered to correspond to the vertical part of a character only if it was large enough and it was significantly rectangular. Rectangularity was tested by determining if the size of the thresholded response area took up 60% or more of its bounding rectangle. If it did, it was allowed, otherwise it was considered to be a spurious response caused by a slanted part in the image. The dimensions of the mask and the threshold were optimized by manual inspection of the results of the masking operation on a small number of images in each font. These values remained constant for the rest of the experiments. An improved feature testing procedure that was run on twenty fonts with widely varying stroke widths is discussed in [6].

Several experimental runs were made to test the performance of this algorithm. The top 100 most frequent words from the Brown Corpus were used to generate word images. While this is a small sample from the entire corpus, these few words represent 483,355 words or 48% of the entire text. Test images were generated from each word in three ways. The first method merely appended the individual character images from a given font. This is designed to test the general purpose performance of the recognition algorithm on good quality input. The second method for generating word images appended the character images and moved them horizontally until the black portions of their images touched. The third method overlapped adjacent characters by two pixels. The second and third techniques are designed to test performance in a situation that is easy for humans to compensate for but is difficult for a recognition algorithm that requires topologically distinct characters.

The results of these tests are presented in Table 2. Under the first input condition, a 100% correct recognition rate was achieved in all cases except the Bodoni Bookface where a recognition rate of 99% was obtained. However, this did not result in an error since the corresponding shape number did not correspond to the shape number of any other word. When the second condition was tested, worst case performance was only 95% and best case performance was 97%. When overlapping characters were used (the third input condition), worst case performance dropped to 85%, but the best performance was still 97%.

Font	%correct not touch.	%correct touching	%correct overlapping
Amer.	100	96	88
Bas.Bold	100	97	90
Bembo	100	96	96
Bod.Bold	100	97	97
Bod.Book	99	95	85

Table 2. Results of recognition experiment for the top 100 most frequent words in the Brown Corpus generated in five different fonts. The performance under three conditions (all characters topologically distinct, all characters touching, and all characters overlapping by two pixels) is shown.

These results indicate that without much tuning, an algorithm could be developed that accurately recognized the shape number of a word. An inspection of the reasons for the incorrect classifications indicates that with further adjustments of the parameters, correct recognition performance could be increased.

The second set of experiments was designed to explore the storage overhead imposed by the inter-word constraints used here. The memory needed to store the dictionaries as well as the two types of transition tables was determined. The results of this experiment are displayed in Table 3. A full text representation of the dictionary was assumed where every entry is represented by a single byte for each character with an extra byte used to mark the end of a word. The word-to-word transition table was represented by the attachment of a linked list to every dictionary entry, where each element of the list contained a two-byte pointer to another dictionary entry as well as a one byte pointer to the next element in the list. A similar setup was used for the shape-to-word transition table, except that a separate dictionary was maintained for shape numbers. More efficient representations could obviously be designed for both data structures, however, these are suitable for comparison purposes.

The results shown in Table 3 illustrate several interesting points. First, it seems as if the number of links needed for the transition tables are a linear function of dictionary size. This is shown by the two columns headed "links per dictionary word" where the number of links divided by the number of dictionary words is displayed. These values do not fluctuate very much as dictionary size is increased, thus showing that the storage needed for the transition tables at least does not increase as a function of the square of the number of dictionary words, as is theoretically possible.

A further observation about storage requirements is that the number of bytes of memory for the linked list representations in all cases is less than twice that needed for the dictionary itself. This leads to the conclusion that the performance improvement provided by either representation of inter-word constraints can be achieved at reasonable cost.

5. CONCLUSIONS

Two methods of incorporating knowledge about transitions between words (referred to as inter-word constraints) in an algorithm for reading text were explored. The reading algorithm includes a step in which the shape of a word is computed and a group of visually similar words are retrieved from a dictionary. This group of words is then used to control further processing that determines which word matches the input image. Inter-word constraints are used to reduce the number of entries in the initial hypothesis set thus improving the performance of the matching process.

The use of two types of inter-word constraints were explored in this paper. Both of these techniques use tables of allowable transitions as their knowledge source. The first one uses knowledge of shape-to-word transitions and the second one uses word-to-word transitions. The effect of both methods on reducing the number of words that initially match an input word was explored in a series of statistical experiments on a large corpus of text. In all cases the average number of matches was reduced by approximately an order of magnitude to the range of 1.2 to 4.4.

A simulation of the word shape computation illustrated its reliability by showing that 85% to 100% correct performance could be achieved. The storage costs of both representations were also explored and it was found that the size of the transition tables only increases as an approximately linear function of the number of dictionary entries. The total number of bytes needed for the transition tables was shown to usually be about twice the number of bytes needed for the original list of dictionary words.

Proc. of Canadian Soc. for Computational Studies of Intelligence, Montreal, Canada, 134-138, May 21-23, 1986.

genre	N_d	$bytes_d$	n-w links	n-w links/ d.wrd.	bytes for n-w links	w-w links	w-w links/ d.wrd.	bytes for w-w links
M	2996	26,071	9280	3.1	27,840	9674	3.2	29,022
R	4362	38,417	12,953	3.0	38,859	13,577	3.1	40,731
D	5714	53,825	23,049	3.9	69,147	23,445	4.1	70,335
L	6285	54,518	28,335	4.5	85,005	30,836	4.9	92,508
H	6628	64,089	33,996	5.1	101,988	36,258	5.5	108,774
P	7664	67,258	34,169	4.5	102,507	37,454	4.9	112,362
C	7724	70,777	25,512	3.3	76,536	27,101	3.5	81,303
N	8025	69,675	34,762	4.3	104,286	38,155	4.8	114,465
K	8490	75,163	34,956	4.1	104,868	38,279	4.5	114,837
B	8608	79,779	35,062	4.1	105,186	37,781	4.4	113,343
E	9803	90,361	45,056	4.6	135,168	49,025	5.0	147,075
A	11,743	107,309	54,023	4.6	162,069	58,733	5.0	176,199
F	12,622	117,945	54,848	4.3	164,544	65,535	5.2	196,605
J	14,062	139,230	82,285	5.9	246,855	90,124	6.4	270,372
G	16,162	155,149	89,199	5.5	267,597	91,144	5.6	273,432

Table 3. The storage cost for the dictionaries and the transition tables. A full text representation for the dictionaries is assumed, and three bytes are needed for each link in the transition tables. $bytes_d$ refers to the number of bytes needed to represent the dictionary. "n-w links" and "w-w links" refer to the number of neighborhood-to-word and word-to-word transitions, respectively.

Therefore, this paper has shown that a relatively simple representation for inter-word constraints can significantly improve the performance of a text recognition algorithm at a relatively modest cost in additional storage. Future work in this area should include additional studies of the storage costs and the development of efficient representations for the transition tables.

Acknowledgements

The author acknowledges the advice of Sargur N. Srihari and the support of the United States Postal Service under the BOA program.

References

1. C. A. Becker, "What do we really know about semantic context effects in reading?", in *Reading Research: Advances in Theory and Practice*, vol. 5 , D. Besner, T. G. Waller and G. E. MacKinnon (editor), Academic Press, New York, 1985, 125-166.
2. A. R. Hanson, E. Riseman and E. G. Fisher, "Context in word recognition", *Pattern Recognition 8* (1976), 35-45.
3. J. J. Hull and S. N. Srihari, "Experiments in text recognition with binary n-gram and viterbi algorithms", *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-4*, 5 (September, 1982), 520-530.
4. J. J. Hull, "Word shape analysis in a knowledge-based system for reading text", *The Second IEEE Conference on Artificial Intelligence Applications*, Miami Beach, Florida, December 11-13, 1985, 114-119.
5. J. J. Hull and S. N. Srihari, "A computational approach to visual word recognition: hypothesis generation and testing", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami Beach, Florida, June, 1986.
6. J. J. Hull, *A computational theory of word recognition*, SUNY at Buffalo, Department of Computer Science, 1986. Ph.D. dissertation.
7. H. Kucera and W. N. Francis, *Computational analysis of present-day American English*, Brown University Press, Providence, Rhode Island, 1967.
8. K. Rayner, M. Carlson and L. Frazier, "The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences", *Journal of Verbal Learning and Verbal Behavior* 22 (1983), 358-374.
9. J. Schurmann, "Reading machines", *Proceedings of the 6th International Conference on Pattern Recognition*, Munich, West Germany, October, 1982, 1031-1044.
10. R. J. Shillman, *Character recognition based on phenomological attributes: theory and methods*, Massachusetts Institute of Technology, August, 1974. Ph.D. Dissertation.