Word Recognition Result Interpretation Using the Vector Space Model for Information Retrieval

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

Abstract

A method is presented to filter the output of a word recognition algorithm, which may contain errors, to locate decisions that should be correct with a high degree of certainty. The algorithm uses the output of a word recognition system and a vector space model for information retrieval to locate a set of documents that have topics which are similar to that of the input document. The vocabulary from these similar documents is then used to locate the correct word recognition decisions. Experimental results show that a subset of the word recognition decisions for an input document can be located that are between 90 and 99 percent correct. This performance was obtained on word recognition results for a sample of text into which a 13, 24, and 30 percent word recognition error rate had been introduced The subset located by this method can be used to drive other recognition processes applied to the rest of the text.

1. Introduction

The recognition of word images is a solution to text recognition by which images of text are transformed into their ASCII equivalent. Word recognition algorithms are an alternative to traditional character recognition techniques that rely on the segmentation of a word into characters. This is sometimes followed by a postprocessing step that uses a dictionary of legal words to select the correct choices.

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.

Solutions to improving the performance of a text recognition system have utilized the context of the language in which the document was written. Examples include using the syntax [7] as well as the semantics [1] of the underlying passage of text.

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.

This paper proposes a methodology to predict the vocabulary of a document from its word recognition decisions. The N best recognition choices for each word are used in a probabilistic model for information retrieval to locate a set of similar document in a database. 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. Those words could then be used as "islands" to drive other processing that would recognize the remainder of the text. 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.

The rest of this paper presents the algorithm in more detail. The technique for locating similar documents in a database is discussed. The use of the vocabulary from those documents to filter the word recognition output is presented. Experimental results demonstrate the ability of the algorithm to select a subset of the word recognition decisions that have a high probability of correctness.

2. Algorithm Description

The algorithmic framework discussed in this paper is presented in Figure 1. 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.

3. 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 [9]. 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 term represents 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:

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

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

Symp. on Document Analysis and Information Retrieval, Las Vegas, NV, April 26-28, 1993.

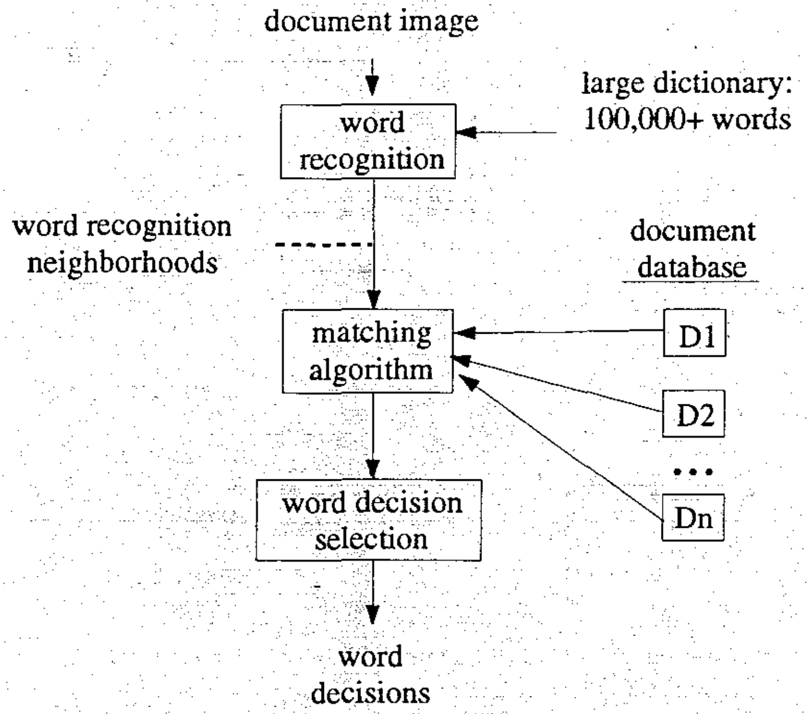

Figure 1. Algorithm description.

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.

3.1. Weight Calculation

The need to adapt the matching strategy to the output from a word recognition algorithm is illustrated in Figure 2. For each word in a document, a number of choices are possible. Those choices are ranked and assigned a confidence of being correct by the recognition algorithm. However, only one of those choices is correct.

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 confidences assigned to that word by the recognition algorithm. In the example word recognition input shown in Figure 2, the term frequency for each word would be its confidence. An exception would be the word `work` that would be assigned a term *input*

He was at work

output

choice	conf	choice	conf	choice	conf	choice	conf
He	0.9	was	0.95	as	0.67	work	0.85
His	0.08	work	0.04	at	0.32	word	0.12
Had	0.02	will	0.01	is	0.01	worm	0.03

Figure 2. Example word recognition output.

frequency of 0.89 since it occurs in two neighborhoods, once with a confidence of 0.85 and another time with a confidence of 0.04.

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

3.2. 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 our application, the index terms in an ASCII document are calculated by first removing all the stop words and proper nouns. Stop words occur frequently in a normal text passage and convey little meaning. Proper nouns are names of specific persons, places, or things. Every other word is assumed to be an "index term" for the purpose of matching.

The words in the neighborhoods of stop words and proper nouns are not used in the similarity calculation because reliable techniques exist to locate them in document images. They are also relatively easy to locate in ASCII text samples.

Stop words can be found in a document image with a clustering algorithm [6]. A simple lookup is sufficient to find stop words in an ASCII text sample [4]. Proper nouns can also be located in a document image by the presence of capitalization as well as their context within the text passage [3]. The location of proper nouns in an ASCII document can be reliably determined by part-of-speech tagging techniques [2].

4. Experimental Investigation

The word decision selection algorithm presented in this paper was demonstrated on the Brown corpus [8]. The Brown corpus is a collection of over one million words of running text that is divided into 500 samples

Symp. on Document Analysis and Information Retrieval, Las Vegas, NV, April 26-28, 1993.

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.

4.1. Testing Data

One of the samples in the Brown corpus was selected as a test document to demonstrate the algorithm presented in this paper. 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. Raster images were generated for those words with a postscript-to-bitmap generation technique. This was done to provide test data for a recognition algorithm that would compute neighborhoods of visually similar words for each of the 885 input words. The stop words and proper nouns were excluded from the test data set since it was assumed that algorithms existed to find those words in a document image.

Neighborhoods were generated for each word in G02 with a word shape calculation in which a feature vector that describes the global characteristics of a word is compared to similar feature vectors for each word in a dictionary [5]. A ranking of the dictionary results in which words that are visually similar to an input image are ranked close to the top. For the experimentation discussed here, the approximately 53,000 unique words that occur in the Brown Corpus were placed in the 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.

4.2. Training Data and Results

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.

4.3. Vocabulary Overlap

The intersection of the vocabularies of unique words in G02 and the ten most similar samples shows how well the lexicon of G02 is covered by the lexicon from the similar samples. Overall, there were 575 unique words among the 885 words that were left in G02 after the stop words and proper nouns were removed. Table 2 shows the number of unique words (excluding stop words and proper nouns) in each of the ten most similar samples, the cumulative lexicon size, and the number of unique words in G02 that are included in the cumulative lexicon.

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

Table 1. Ten most similar samples to G02

These results show that 56 percent of the lexicon of G02 is covered by the lexicon of 3103 words provided by all ten of the similar samples. This can be compared to the 85 percent coverage of the lexicon of G02 that is provided by the 37,781 unique words in the other 499 samples of the Brown corpus besides G02 that are neither stop words nor proper nouns.

4.4. 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 under three noise conditions using three selection criteria.

Noise was introduced in the word recognition output to test the tolerance of the decision selection procedure to imperfect input. A uniform random number generator was used to select a given number of neighborhoods from among those that had the correct decision as the first choice. The second choice was substituted for the first, thus providing a neighborhood that contained a visually similar, but incorrect, word in the first position. A 24 percent error rate was simulated by applying the above procedure to 93 of the 769 neighborhoods that were correct (i.e., the top choice of the recognition algorithm was correct). A 30 percent error

sample	G02	J42	J22	G25	H02	H20	G72	G35	H22	H19	G10	BC-G02
uniq. words	575	513	578	472	498	532	497	610	396	473	420	37,781
cumulative	-	513	980	1321	1602	1964	2211	2537	2723	2900	3103	37,781
intersection	0	114	181	216	240	257	273	289	296	307	323	488
lex. coverage	0%	20%	31%	38%	42%	45%	47%	50%	51%	53%	56%	85%

Table 2. Vocabulary coverage for G02.

rate was introduced with a similar method.

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 percent 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 correct and erroneous decisions are shown below.

correct word	decision	correct word	decision
certainty	century	aspects	expense
government	governmental (3)	statesman	common
movement	common	separate	square
natural	control		

Inspection of these nine errors is interesting. The word *government* was misrecognized as *governmental* three times. However, since government is a root form of governmental, in some sense those errors are not fatal and might be considered correct. This interpretation would correspond to the use of a stemming algorithm in a traditional information retrieval application.

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 percent input words were effectively recognized with a correct rate of 97 percent. The results for the *G02-nouns* matching condition show that up to 26 percent of the input can be recognized with a 99 percent correct rate. In the *nouns-matching* condition, 29 percent of the input words can be recognized with a 97 percent correct rate.

The results of word decision selection when applied to the corrupted versions of G02 mentioned before (with 24 percent and 30 percent error in the top choices) using the *overall* and *nouns-matching* selection criteria are given in Table 3. In the *overall* condition with 24 percent noise, an increase in correct rate from 76 percent to 93 percent is observed with up to 42 percent of the words being chosen. When the noise is increased to 30 percent, the correct rate is raised from 70 percent to 89 percent on 42 percent of the words.

samples	decision selection criteria								
	overall			*G02-nouns*			*nouns-matching*		
used	matches	errors	corr.pct	matches	errors	corr. pct	matches	errors	corr. pct
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

Table 2. Word selection performance on the original 885 neighborhoods (with 87 percent correct at the top choice.

	top choice error rate											
	24 percent						30 percent					
	decision selection criterion						decision selection criterion					
samples	*overall*			*nouns-matching*			*overall*			*nouns-matching*		
used	matches	errors	corr. pct	matches	errors	corr. pct	matches	errors	corr. pct	matches	errors	corr. pct
1	210	13	94	112	8	93	192	18	91	103	12	88
2	296	20	93	186	11	93	272	27	90	148	17	88
3	339	24	93	183	13	93	316	35	89	171	20	88
4	378	25	93	211	13	93	356	38	90	200	22	89
5	397	29	93	221	16	93	373	42	89	211	26	89
6	391	28	93	233	16	93	364	38	90	217	27	88
7	416	35	93	240	19	93	383	43	89	223	28	88
8	420	35	92	244	19	93	403	49	89	231	29	88
9	435	36	92	248	19	92	414	51	88	231	29	87
10	462	44	90	256	21	92	424	56	87	238	31	87

Table 3. Word selection performance on the corrupted versions of the 885 neighborhoods.

5. Discussion and Conclusions

This paper presented an adaptation of the vector space model for information retrieval to improving the performance of a word recognition algorithm. The neighborhoods of visually similar words determined by word recognition are matched to a database of documents and a subset of documents with topics that are similar to those of the input image are determined. The vocabulary from those similar documents are used to select the word recognition decisions that have a high probability of being correct.

This approach is based on the observation that any given document may contain only 500 or so distinct words but that a word recognition algorithm may need a dictionary of 100,000 or more words to be considered ''comprehensive.'' Also, two documents about similar topics may share a significant portion of their vocabularies. Thus, if the decisions of a word recognition algorithm on a given document are intersected with the vocabulary from a document of a similar topic, word recognition errors will occur in the output only if the recognition algorithm corrupted a correct word into another word that also occurred in the matching documents.

Because of the disparity in size between the word recognition dictionary and a document's vocabulary (200:1), it was expected that a low error rate would be attained in the intersection even though the input error rate might be high. Thus it would be possible to locate words in the input document that would have a high probability of being correct. This could be valuable information, especially in recognizing noisy documents, where a high recognition error rate might be expected, and where highly confident word decisions could be used to drive subsequent recognition processing of the remainder of the text.

Future work will consider the use of word co-occurrence probabilities to predict words that are likely to occur in a document based on the samples that are judged to be similar to it. Also, the use of the keyword weights from the matched document to correct recognition errors will be considered.

Acknowledgments

Dr. Tin Kam Ho provided the word recognition algorithm.

References

1. H. S. Baird and K. Thompson, "Reading Chess," *IEEE Transactions on Pattern Analysis and Machine Intelligence 12* (1990), 552-559.
2. K. W. Church, "A stochastic part of speech program and noun phrase parser for unrestricted text," *Second Conference on Applied Natural Language Processing*, Austin, Texas, 1988.
3. G. DeSilva and J. J. Hull, "Proper noun location in document images," *Center of Excellence for Document Analysis and Recognition*, 1993.
4. C. Fox, "A stop list for general text," *SIGIR Forum 24* (Fall/Winter, 1989), 19-35.
5. T. K. Ho, J. J. Hull and S. N. Srihari, "A computational model for recognition of multifont word images," *Machine Vision and Applications, special issue on Document Image Analysis*, Summer, 1992, 157-168.
6. J. J. Hull, S. Khoubyari and T. K. Ho, "Visual Global Context: Word Image Matching in a Methodology for Degraded Text Recognition," *Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, March, 1992, 26-39.
7. J. J. Hull, "Incorporation of a Markov model of language syntax in a text recognition algorithm," *Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, March, 1992, 174-185.
8. H. Kucera and W. N. Francis, *Computational analysis of present-day American English*, Brown University Press, Providence, Rhode Island, 1967.
9. G. Salton, *Automatic text processing*, Addison Wesley, 1988.