# Degraded Text Recognition Using Word Collocation and Visual Inter-Word Constraints

**Tao Hong and 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
taohong@cs.buffalo.edu hull@cs.buffalo.edu

## Abstract

Given a noisy text page, a word recognizer can generate a set of candidates for each word image. A relaxation algorithm was proposed previously by the authors that uses word collocation statistics to select the candidate for each word that has the highest probability of being the correct decision. Because word collocation is a local constraint and collocation data trained from corpora are usually incomplete, the algorithm cannot select the correct candidates for some images. To overcome this limitation, contextual information at the image level is now exploited inside the relaxation algorithm. If two word images can match with each other, they should have same symbolic identity. Visual inter-word relations provide a way to link word images in the text and to interpret them systematically. By integrating visual inter-word constraints with word collocation data, the performance of the relaxation algorithm is improved.

## Introduction

Word collocation is one source of information that has been proposed as a useful tool to post-process word recognition results([1, 4]). It can be considered as a constraint on candidate selection so that the word candidate selection problem can be formalized as an instance of constraint satisfaction. Relaxation is a typical method for constraint satisfaction problems. One of the advantages of relaxation is that it can achieve a global effect by using local constraints.

Previously, a probabilistic relaxation algorithm was proposed for word candidate re-evaluation and selection([2]). The basic idea of the algorithm is to use word collocation constraints to select the word candidates that have a high probability of occurring simultaneously with word candidates at other nearby locations. The algorithm runs iteratively. In each iteration, the probability of each word candidate is upgraded based on its previous probability, the probabilities of its neighbors and word collocation data. The initial probability of each word candidate is provided by a word recognizer. The relaxation process terminates when the probability of each word candidate becomes stable. After relaxation finishes, for each word image, the word candidate with highest probabilistic score will be selected as the decision word.

Because the window size of word collocation is usually small, word collocation is a local constraint. Because word collocation data are derived from text corpora, it usually is incomplete and unbalanced. Those properties limit the usefulness of word collocation for candidate selection. By analyzing the performance of the algorithm, three sources of errors were identified: (1). the local context cannot provide enough information to distinguish the competitive candidates; (2). word collocation data trained from corpora are not complete so that it does not include the statistical data needed to select the correct candidate; and (3). word collocation data trained from unbalanced corpora are biased so that the wrong candidate is selected.

In a normal English text, there are many occurrences of the same words. Because the main body of a text is usually prepared in the same font type, different occurrences of the same word are visually similar even if the text image is highly degraded.

Visual similarity between word images can place useful constraints on the process of candidate selection([3]). If two word images can match with each other, their identities should be the same. For example, if there are two sentences, *"Please fill in the application X "* and *"This Y is almost the same as that one"*, where $X$ and $Y$ are visually similar, and both of them have the candidate set { *farm, form* } . The candidate *"form"* can be easily selected as the decision for $X$ and $Y$ if we consider both word collocation and visual inter-word constraints, although it is difficult to select a candidate for $Y$ by only using word collocation.

## Modified Relaxation Algorithm

Figure 1 is the description of the new relaxation algorithm that integrates word collocation and visual inter-word constraints for candidate selection. Given a sequence of word images from a text page, the first step of the algorithm is word image clustering. Then, a word recognizer is applied to the prototype for each image cluster to generate a set of word candidates. Each word inside a cluster inherits the candidate set for the cluster. In an iteration of relaxation, the probabilistic scores of the candidates for a word image are upgraded based on word collocation data. The probabilistic scores of the candidates for a cluster are upgraded by summing up the probabilistic scores of the word images inside the cluster. Each word image then inherits the candidate set from the cluster it belongs to. When there is no further significant change in the confidence scores, the relaxation stops. The top candidate for each word image is selected as the decision.

```
INPUT: A sequence of word images $W[i]$, $1 \leq i \leq n$;
OUTPUT: $W[i].decision$, $i = 1, 2, ...n$

/*Word Image Clustering*/
ClusterList ← {};
FOR i=1 to n DO
   FoundMatch ← FALSE;
   FOR each cluster C[j] in ClusterList DO
      IF( Distance(W[i].image, C[j].prototype) < threshold )
         C[j].ImageList ← C[j].ImageList ∪ W[i];
         W[i].ClusterIndex ← j;
         FoundMatch ← TRUE;
   IF ( FoundMatch == FALSE )
      Create a new cluster C[k];
      C[k].ImageList ← W[i];
      W[i].ClusterIndex ← k;
      ClusterList ← ClusterList ∪ C[k];

/*Isolated Word Recognition–Candidate Generation*/
FOR each cluster C[j] in ClusterList DO
   C[j].CandidateList ← WordRecognition(C[j].prototype);
   Sort candidates in C[j].CandidateList in decreasing order;

IterationCount ← 0;
REPEAT
   IterationCount ← IterationCount + 1;
   /* Generate Word Lattice */
   FOR each word image W[i] DO
      W[i].CandidateList ← C[W[i].ClusterIndex].CandidateList;

   /* Upgrade Confidence Scores For Candidates Of Word Images*/
   FOR each word image W[i] DO
      FOR each word candidate w[m] in W[i].CandidateList DO
         Upgrade w[m].prob by using word collocation;

   /* Upgrade Confidence Scores For Candidates Of Clusters*/
   FOR each cluster C[j] in ClusterList DO
      FOR each candidate c[n] in C[j].CandidateList DO
         c[n].prob ← 0.0;
         FOR each word image W[i] in C[j].ImageList DO
            FOR each word candidate w[m] in W[i].CandidateList DO
               IF( c[n].string == w[m].string )
                  c[n].prob ← c[n].prob + w[m].prob;
      Sort candidates in C[j].CandidateList in decreasing order;
UNTIL probabilistic scores of word candidates become stable;

/* Select Best Candidate For Word Image */
FOR each word image W[i] DO
   W[i].decision ←
      CandidateWithHighestScore(C[W[i].ClusterIndex].CandidateList);
END
```

Figure 1: Augmented Relaxation Algorithm

## Experiments and Analysis

Five articles from the Brown Corpus, *A06*, *G02*, *J42*, *N01* and *R07*, were randomly selected as testing samples. There are totally 11,402 words in those testing samples. For each word, a top10 candidate list was generated. The top1 correct rate is around 55% on highly degraded text. Word collocation data was trained from the Penn Treebank and the Brown Corpus after removing the testing samples. We used the frequency of a word pair to measure its collocation strength. There are totally 1,200,000 unique word pairs after training.

The result of applying the relaxation algorithm to the noisy text images is shown in Table 1. The top1 correct rate of word recognition is as low as 57%. Relaxation based on word collocation can improve top1 correct rate to 83%. After integrating word collocation and visual constraints, the correct rate of the first choice can be further improved to 88%. There is overall 5% improvement by introducing visual contextual constraints.

| | top1 | top2 | top3 | top5 |
|---|---|---|---|---|
| Word Recognition | 57.10% | 78.47% | 87.51% | 92.47% |
| Original Relaxation | 83.19% | 92.99% | 96.47% | 98.61% |
| Augmented Relaxation | 88.22% | 94.99% | 97.37% | 98.91% |

Table 1: Relaxation Results

## Conclusions

A word-collocation-based relaxation algorithm was proposed for candidate selection in degraded text recognition. Word collocation is a local statistical constraint, which sometimes is not sufficient to distinguish among the candidates. To make candidate selection more accurate, visual inter-word constraints are investigated. A new relaxation algorithm augmented with visual inter-word constraints was designed. Experimental results showed that the modified algorithm has better performance.

## References

[1] K. W. Church and P. Hanks, "Word Association Norms, Mutual Information, and Lexicography," *Computational Linguistics*, Vol. 16, No. 1, pp. 22-29, 1990.

[2] T. Hong and J. J. Hull, "Degraded Text Recognition Using Word Collocation", in *Proceedings of the Conference on Document Recognition of 1994 IS&T/SPIE Symposium*, San Jose, CA, February 6-10, 1994.

[3] T. Hong, "Integration Of Visual Inter-Word Constraints And Linguistic Knowledge In Degraded Text Recognition", in *Proceedings of 32nd Annual Meeting of Association for Computational Linguistics*, pp. 328-330, Las Cruces, New Mexico, 27-30 June, 1994(in Student Session).

[4] T. G. Rose and L. J. Evett, "Text Recognition Using Collocations and Domain Codes," in *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pp. 65-73, Columbus, Ohio, 1993.