Text Recognition Enhancement with a Probabilistic Lattice Chart Parser

Tao Hong and Jonathan J. Hull
Center of Excellence for Document Analysis and Recognition (CEDAR)
State University of New York at Buffalo
Buffalo, New York 14228-2567
email: hull@cs.buffalo.edu, taohong@cs.buffalo.edu

Abstract

A probabilistic lattice chart parser is proposed for improving the performance of a text recognition technique. Digital images of words are recognized and alternatives for the identity of each are generated. Local word collocation statistics and a probabilistic chart parsing algorithm are used to determine the top N best parses for each sentence using the alternatives provided for the identity of each word by the recognition system. In this paper, an approach in which text recognition and understanding are tightly integrated is discussed. An objective of this approach is to provide the capability to process images of unrestricted English text. A large-scale lexicon, which supports the system, was acquired by training on corpora of over three million words. This paper focuses on the implementation and performance of the probabilistic lattice chart parser.

Topic areas: visual text recognition and understanding, natural language parsing and word lattice parsing.

1 Introduction

The objective of visual text recognition is to transform an arbitrary image of text into its ASCII equivalent. This process is usually performed by recognizing the images of isolated characters. Such methods are reliable for well-printed images. However, they do not work well for text that has been degraded such as multiple-generation photocopies or facsimiles.

An alternative is to recognize images of words and to produce a set of alternatives for the identity of each word image. The performance of word recognition is reliable if measured over a reasonable number of alternatives. Accuracies of 85 percent correct in the top choice, but better than 99 percent correct in the top ten choices have been reported for highly degraded machine-printed text ([5]).

The integration of language-level contextual information with text recognition is one method that has been used to improve the accuracy of the top choice. The objective of such techniques is to reduce the number of possible hypotheses, or at best, to select a most preferred hypothesis using language-level knowledge, such as lexical, syntactic, semantic and pragmatic knowledge. Previous approaches have utilized local word-to-word transitions [8], statistical part-of-speech transitions [9], word collocation constraints [14] and semantic constraints from a limited domain [1].

Theoretically, other techniques developed in natural language understanding(NLU), such as parsing, can be integrated with text recognition. However, some problems in processing unrestricted English text are difficult to overcome. For example, a large-scale English lexicon which supports unrestricted English text understanding is difficult to build.

Statistical and structural approaches have been combined to overcome some of the difficulties in extending parsing techniques to unrestricted text ([4, 13]). Automatic acquisition of lexical knowledge also has become a promising approach for building the large-scale lexicons needed by such techniques ([3]).

Speech recognition is similar to visual word recognition. There have been several efforts to integrate NLU with speech recognition to improve its performance ([15]). However, these techniques face similar problems in processing text from outside a restricted domain.

In this paper, we propose a methodology for integrating structural language analysis with word recognition to improve its top choice performance. A *probabilistic lattice chart parser* is described that uses syntactic and semantic constraints to find the best candidate for each word image.

0-8186-4960-7/93 $3.00 © 1993 IEEE

Second IAPR Int. Conf. on Document Analysis and Recognition, Tsukuba, Japan, Oct. 20-22, 1993, 222-225.

2 Algorithm Description

Generally, there are two types of linguistic constraints used in the system. One is local word collocation in which the identity of a word is used to predict the identity of other nearby words ([2]). Global structural constraints are also used. These include syntactic, semantic and pragmatic constraints. The global structural constraints are exploited with a chart parsing model. The chart data structure provides a flexible framework for parsing([11, 16]). Statistical methods can be easily incorporated in a chart parser([13]). A chart parser can also be extended to a lattice parser which allows for several word candidates at the same position, and therefore can be directly used for speech recognition ([15]) and visual text recognition. 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.

In our approach (see **Figure. 1**), given a page of English text that has been segmented into sentences, a word recognition process generates a candidate list 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 a word lattice which is passed to the probabilistic lattice chart parser that builds all possible parse trees based on the reduced word lattice; finally, the word candidates that occur in the parse tree with the highest a posteriori probability are as well as the parse tree itself are output.

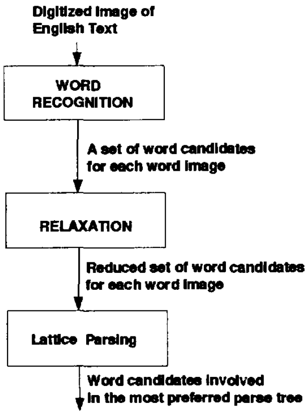
Digitized Image of English Text
WORD RECOGNITION
A set of word candidates for each word image
RELAXATION
Reduced set of word candidates for each word image
Lattice Parsing
Word candidates involved in the most preferred parse tree

Figure 1: Overall Design

It is very time-consuming to manually build a large-scale English lexicon that can support an unrestricted text understanding system. Automatic acquisition of lexical knowledge is a promising approach towards solving this problem([3]). Following this approach, we built an English lexicon by training on English text corpora that contain several million words. The corpora used are the entire Brown Corpus and part of the Penn Treebank. Currently, our lexicon has more than 95,000 entries, with about 80,000 different words. For each word, we have statistical data about 1) its frequency; 2) it's potential part-of-speech tags and their frequencies; and 3) the words strongly collocated with it and their word collocation scores. The parser uses all of these data.

The word recognition system generates the top-n candidates for each word image except for frequent function words, such as "*a*", "*the*" and "*of*". For the function words, we assume they can be recognized correctly by a word image matching and clustering procedure([10]). Thus, the function words are "islands" or words with identities that the parser can rely on.

The relaxation algorithm works as a filter to reduce the top-n word candidates at each location to the 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 a high probability of occurring simultaneously with word candidates at other nearby locations. The details of the relaxation algorithm can be found in [7].

3 Probabilistic Lattice Parser Based on Chart Data Structure

There are 706 rules in the context-free grammar (CFG) that were written manually. For example, the following rules are typical.

```
S      <- S-BODY
S-BODY <- NP VP
NP     <- NN
NP     <- NNS
NP     <- NP CC NP
VP     <- VB NP
VP     <- VB
```

This shows that a sentence(S) can be written as a sentence body(*S-body*) followed by a period. A sentence body is a noun phrase(NP) followed by a verb phrase(VP). A noun phrase is composed of a singular noun(NN), a plural noun(NNS), or two noun phrases with a coordinating conjunction (CC) between them. A verb phrase is composed of a verb(VB) followed by a noun phrase or just a verb.

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. Basically, our parser is a Cocke-Kasami-Younger(CKY) parser([16]). Every constituent or structure derived during parsing is recorded as an edge and every edge represents a completed constituent. This property restricts the number of edges in the chart during parsing. Edges are organized as trees or forests in which some parent edges may share the same child edge.

If parsing succeeds, it will print out the parse tree represented by the "*S*" edge with highest confidence score. All possible complete parse trees, can be generated in order of decreasing confidence score. If parsing fails, the parser will work as a partial parser and print out parse trees for fragments of the input word sequence.

4 Experimental Results

We chose 3 articles, **A06**, **G02** and **J42**, from the **Brown Corpus** as test samples. The **Brown Corpus** is a collection of 500 samples of English texts, each of which contains approximately 2000 words([12]). **A06** is a collection of six short articles from the *Newark Evening News*. **G02** is from an article "*Toward a Concept of National Responsibility* " from *The Yale Review*. **J42** is from a book "*The Political Foundation of International Law*". Some further information about the sample articles is listed in **Table. 1**.

Article	Num. of Words	Num. of Sentences	Average Length of Sentence	Maximal Length of Sentence
A06	2213	88	25.15	70
G02	2267	85	26.67	68
J42	2269	75	30.25	88
total	6749	248	27.21	88

Table 1: Information about the test samples

First, the parser was applied to the sentences from those articles. The results of parsing on the plain ASCII text are shown in **Table. 2**. For 204 out of 248 sentences(82.2%), the parser generated at least one complete parse tree. For those sentences without a complete parse tree, we found they contained some syntactic structures that were not covered by the grammar. The parser generated partial parse trees for most of the sentences that failed to be parsed completely. We manually checked the most preferred parse trees generated for the sentences from **G02** to see whether they were correct. For 45 out of 76 sentences, the most preferred parse trees are correct if we ignore prepositional phrase attachment and conjunct scoping problems. For the remaining sentences from **G02** with complete parse trees, the most preferred parse trees are partially correct. In **Table. 2**, the high standard deviation(σ) values show that the number of edges created, the parsing time, and the number of complete trees generated, vary significantly from sentence to sentence.

Article	Num. of Sent. with Complete Parse Tree	Avg Num. of Edges Generated	Avg Time of Parsing (sec.)	Avg Num. of Complete Parse
A06	76	2375 $\sigma = 4417$	41.0 $\sigma = 133.4$	14.7 $\sigma = 35.5$
G02	72	2291 $\sigma = 4876$	50.2 $\sigma = 230.5$	67.7 $\sigma = 185.1$
J42	56	2784 $\sigma = 6279$	55.3 $\sigma = 194.4$	174.8 $\sigma = 905.0$
total	204	2469 $\sigma = 5200$	48.5 $\sigma = 189.7$	77.3 $\sigma = 492.4$

Table 2: Result of Sentence Parsing

Next, we used all the sentences from the three articles to test the application of the parser for text recognition. After printing those sentences with *ditroff* in a 12 point font on paper using a laser printer, text images were created by digitizing the paper. The text images were segmented into sentences, and further into word images. Using a word recognition program which is based on word shape analysis([6]), the top ten word candidates were generated for each word image (for the frequent function words such as "*a*", "*the*" and "*of*", and punctuation marks, the recognizer generates just one word candidate). On average, there are 7.78 word candidates for each word image.

The relaxation algorithm generates the two best choices for each word by using word collocation constraints. Here, only collocation constraints for adjacent words are used. The data for word collocation constraints are collected by training on the **Brown Corpus** which has more than one million words.

If we use the word collocation data collected from the **Brown Corpus**, including our test articles: **A06**, **G02** and **J42**, after relaxation, about 97.9% of the correct word candidates still remained in top-2 lists(see **Table. 3**). The parser selected word candidates from the top-2 correctly for 89.4% of the words and also provided a parse tree for further processing. The average time for parsing a sentence is less than four minutes while the average number of edges created in the parser is only about 10,000.

If we use the word collocation data collected from

Second IAPR Int. Conf. on Document Analysis and Recognition, Tsukuba, Japan, Oct. 20-22, 1993, 222-225.

Article	Num. of Correct Candidates Remaining in top-2 After Relaxation	Num. of Correct Candidates Selected After Parsing	Avg Num. of Edges Generated per Sentence	Avg Time of Parsing per Sentence (sec.)	Avg Num. of Complete Parse per Sentence
A06	2187 (98.8%)	2024 (92.4%)	6964 $\sigma = 9295$	69.3	208.2 $\sigma = 326$
G02	2196 (96.8%)	2006 (88.4%)	7095 $\sigma = 11279$	250.5	140.1 $\sigma = 229$
J42	2230 (98.2%)	1982 (87.3%)	11955 $\sigma = 14556$	308.4	270.5 $\sigma = 381$
total	6613 (97.9%)	6034 (89.4%)	8518 $\sigma = 11978$	202.8	203.7 $\sigma = 320$

Table 3: Result of Word Lattice Parsing(using whole Brown Corpus as Training Data)

all the articles in the **Brown Corpus** except the test articles: **A06,G02** and **J42**, the performance of the parser dropped(see **Table. 4**). The overall correct rate of relaxation is 86.4% and the parser correctly selects 76.2% of the words. The reason for the drop is that the relaxation algorithm did not have as reliable collocation data as before. This result suggests that the training corpus is not large enough.

Article	Num. of Correct Candidates Remaining in top-2 After Relaxation	Num. of Correct Candidates Selected After Parsing	Avg Num. of Edges Generated per Sentence	Avg Time of Parsing per Sentence (sec.)	Avg Num. of Complete Parse per Sentence
A06	1861 (84.1%)	1689 (76.3%)	7103 $\sigma = 9787$	65.7	173.8 $\sigma = 282$
G02	2004 (88.4%)	1773 (78.2%)	7245 $\sigma = 11458$	275.6	151.5 $\sigma = 255$
J42	1972 (86.9%)	1681 (74.0%)	12458 $\sigma = 14814$	350.0	291.5 $\sigma = 408$
total	5837 (86.4%)	5143 (76.2%)	8771 $\sigma = 12298$	221.3	201.8 $\sigma = 323$

Table 4: Result of Word Lattice Parsing(using all texts in Brown Corpus except A06, G02 and J42 as Training Data)

5 Conclusions

In this paper, we described a visual text recognition and understanding system and focused on the ***probabilistic lattice chart parser*** it contains. The preliminary results are encouraging. They confirmed our belief that linguistic constraints are a powerful way to improve visual text recognition performance, and that visual text recognition and NLU should be integrated tightly. The data structure and algorithm of our lattice parser is efficient for both sentence parsing and OCR postprocessing. The simple constraints used to prune a parse tree and to reduce the search space are powerful. Experimental results on over 6,000 words of text showed it was possible to improve word recognition performance from 13 percent correct to nearly 80 percent correct.

References

[1] Henry S. Baird and Ken Thompson. Reading chess. *IEEE Transactions on pattern analysis and machine intelligence*, 12(6):552–559, 1990.

[2] Kenneth W. Church, William Gale, Patrick Hank, and Donald Hindle. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[3] Kenneth Ward Church, William Gale, Patrick Hank, and Donald Hindle. Using statistics in lexical analysis. In Uri Zernik, editor, *Lexical Acquisition: Exploitung On-Line Resources to Build a Lexicon*, pages 115–164. Lawrence Erlbaum Associates, Publishers, 1991.

[4] T. Fujisaki, F. Jelinek, J. Cooke an E.Black, and T. Nishino. A Probabilistic Parsing Method for Sentence Disambiguation. In *International Parsing Workshop '89*, 1989.

[5] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. Word recognition with multi-level contextual knowledge. In *Proceedings of the First International Conference on Document Analysis (ICDAR-91)*, pages 905–915, 1991.

[6] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. A word shape analysis approach to lexicon based word recognition. *Pattern Recognition letters*, 13:821–826, 1992.

[7] Tao Hong and Jonathan J. Hull. Degraded text recognition using word collocation. Paper submitted to the Conference on Document Recognition,1994 IS&T/SPIE Symposium, San Jose, Feb. 6-10, 1994.

[8] Jonathan Hull. Inter-word constraints in visual word recognition. In *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence*, May 21-23 1986.

[9] Jonathan Hull. Incorporation of a markov model of language syntax in a text recognition algorithm. In *Symposium on Document Analysis and Information Retrievals*, pages 174–185, March 16-18 1992.

[10] Jonathan Hull, Siamak Khoubyari, and Tin Kam Ho. Word image matching as a technique for degraded text recognition. In *Proceedings of 11th International Conference on Pattern Recognition*, 1992.

[11] Martin Kay. Algorithm Schemata and Data Structures in Syntactic Processing. Technical Report 80-12, CSL, October 1980.

[12] H. Kucera and W. N. Francis. *Computational Analysis of Present-day American English*. Brown University Press, 1967.

[13] David M. Magerman and Carl Weir. Efficiency, Robustness and Accuracy in Picky Chart Parser. In *Proceedings of 30th Conference of ACL*, 1991.

[14] T.G. Rose, L.J. Evett, and R.J. Whitrow. The Use of Semantic Information as an Aid to Handwriting Recognition. In *Proceedings of the First International Conference on Document Analysis (ICDAR-91)*, pages 629–637, 1991.

[15] M. Tomita. An Efficient Word Lattice Parsing Algorithm for Continuous Speech Recognition. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing*, 1986.

[16] Daniel H. Younger. Recognition and Parsing of Context-Free Language in Time n^3. *Information and Control*, 10:189–208, 1967.