2014-04-03

Ensemble Methods for Historical Machine-Printed Document Recognition

William B. Lund
Brigham Young University - Provo

Ensemble Methods for Historical Machine-Printed Document Recognition

William B. Lund

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Eric K. Ringger, Chair
Kevin Seppi
Bryan Morse
Michael Jones
Mark Clement

Department of Computer Science

Brigham Young University

April 2014

ABSTRACT

Ensemble Methods for Historical Machine-Printed Document Recognition

William B. Lund
Department of Computer Science, BYU
Doctor of Philosophy

The usefulness of digitized documents is directly related to the quality of the extracted text. Optical Character Recognition (OCR) has reached a point where well-formatted and clean machine-printed documents are easily recognizable by current commercial OCR products; however, older or degraded machine-printed documents present problems to OCR engines resulting in word error rates (WER) that severely limit either automated or manual use of the extracted text. Major archives of historical machine-printed documents are being assembled around the globe, requiring an accurate transcription of the text for the automated creation of descriptive metadata, full-text searching, and information extraction. Given document images to be transcribed, ensemble recognition methods with multiple sources of evidence from the original document image and information sources external to the document have been shown in this and related work to improve output. This research introduces new methods of evidence extraction, feature engineering, and evidence combination to correct errors from state-of-the-art OCR engines. This work also investigates the success and failure of ensemble methods in the OCR error correction task, as well as the conditions under which these ensemble recognition methods reduce the Word Error Rate (WER), improving the quality of the OCR transcription, showing that the average document word error rate can be reduced below the WER of a state-of-the-art commercial OCR system by between 7.4% and 28.6% depending on the test corpus and methods.

This research on OCR error correction contributes within the larger field of ensemble methods as follows. Four unique corpora for OCR error correction are introduced: The Eisenhower Communiqués, a collection of typewritten documents from 1944 to 1945; The Nineteenth Century Mormon Articles Newspaper Index from 1831 to 1900; and two synthetic corpora based on the Enron (2001) and the Reuters (1997) datasets. The Reverse Dijkstra Heuristic is introduced as a novel admissible heuristic for the A* exact alignment algorithm. The impact of the heuristic is a dramatic reduction in the number of nodes processed during text alignment as compared to the baseline method. From the aligned text, the method developed here creates a lattice of competing hypotheses for word tokens. In contrast to much of the work in this field, the word token lattice is created from a character alignment, preserving split and merged tokens within the hypothesis columns of the lattice. This alignment method more explicitly identifies competing word hypotheses which may otherwise have been split apart by a word alignment. Lastly, this research explores, in order of increasing contribution to word error rate reduction: voting among hypotheses, decision lists based on an in-domain training set, ensemble recognition methods with novel feature sets, multiple binarizations of the same document image, and training on synthetic document images.

Keywords: historical document recognition, optical character recognition, OCR, OCR error correction, multiple sequence alignment, MSA, text alignment, progressive alignment, machine learning

ACKNOWLEDGMENTS

My deepest thanks go to my wife who has been fully supportive of this effort for many years. She has been steadfast and long suffering, believing that I could complete this work.

Thanks to Dr. Eric Ringger, who on more than one occasion assured me that completing this work was possible. He stated that he was “my biggest fan,” which provided much needed encouragement. I appreciate all of the efforts and contributions of my committee which have made significant improvements to my research and this document.

I express my appreciation to the Fulton Supercomputing Lab and the Harold B. Lee Library at Brigham Young University without whose support this research would not have been possible. In particular I thank Randy Olsen, the University Librarian at the start of my program, for encouraging me to begin the doctorate. The University Librarians since Randy’s departure, specifically Julene Butler, Scott Duvall, and Jennifer Paustenbaugh, have been very supportive of this work, seeing in it a way to build connections between the library and the Computer Science Department, which benefits both organizations.

I am appreciative to my colleagues in the Ph.D. program, Dr. Robbie Haertel and Dr. Daniel Walker, whose collaboration was important to me. I also express thanks to Dr. Walker, whose work in creating synthetic data sets was very useful. I thank the employees of the Digital Initiatives Lab of the Harold B. Lee Library for their extensive work in digitizing, OCRing, and correcting the OCR text of the Nineteenth Century Mormon Article Newspaper Index. Lastly, I express gratitude to Patricia A. Fraud of the Harold B. Lee Library for her extensive work in assembling, curating, and making available the Nineteenth Century Mormon Article Newspaper Index, and Dr. Richard Hacken for curating and making available the Eisenhower Communiqués digital collection.

Finally, I express my gratitude to my Heavenly Father, Who on more than one occasion provided assistance, guidance, and support.

Table of Contents

List of Figures **xi**

List of Tables **xv**

1 Introduction **1**

1.1 Thesis Statement 3

1.2 Contributions 4

1.3 Research Overview 5

1.3.1 Corpora 7

1.3.2 Document Scanning 8

1.3.3 Optical Character Recognition 8

1.3.4 Document Digitized Image Binarization 11

1.3.5 Ensemble Methods 13

1.3.6 Text Alignment, Lattice Creation, and Hypothesis Segmentation 13

1.3.7 Space Propagation 16

1.3.8 Lattice Features and Scoring 16

1.3.9 Hypothesis Selection 17

1.3.10 Validation with the Word Error Rate 17

1.4 Dissertation Organization 18

2 Background and Related Works **21**

2.1 General Formulation 22

2.2 Post-OCR Error Correction 24

2.2.1 Post-OCR Error Correction Formulation 25
2.2.2 Post-OCR Error Correction Discussion 26
2.3 Alignment 28
2.3.1 Tokenization and Alignment Order 28
2.3.2 Optimal Alignment 30
2.3.3 Heuristic Progressive Alignment 31
2.3.4 Character Alignment Costs 32
2.3.5 Alignment Formulation 33
2.3.6 Non-Alignment Approaches 35
2.4 Ensemble Methods 35
2.5 Diversity in Ensemble Methods 37
2.5.1 Diversity Formulation 41
2.5.2 Lattice Word Error Rate in Measuring Diversity 41
2.5.3 Diversity Discussion 43
2.6 Types of Ensemble Methods 43
2.6.1 Bagging and Boosting 43
2.6.2 Conditional Random Fields 44
2.6.3 Minimum Bayes Risk Decoding 45
2.6.4 N-best hypotheses 45
2.6.5 Reranking 46
2.6.6 System Combination 47
2.6.7 Voting 47
2.7 Applications of Ensemble Methods 49
2.7.1 Classifiers 49
2.7.2 Handwriting Recognition 50
2.7.3 Machine Translation 51
2.7.4 Named-entity Recognition 51

2.7.5 Post-correcting Using Multiple OCR Outputs From the Same Engine . . . 52

2.7.6 Post-Correcting Using Multiple OCR Outputs from Different Engines . . . 53

2.7.7 Automatic Speech Recognition 53

2.8 Corpora 54

2.8.1 OCR Word Error Rate 55

2.8.2 Synthetic Corpora 55

2.8.3 Corrected Historical Document Corpora 57

2.8.4 Research Corpora 57

2.9 Binarization 58

2.10 Conclusion 59

3 Improving Optical Character Recognition through Efficient Multiple System Alignment 60

3.1 Introduction 61

3.2 Approach 62

3.2.1 Background 62

3.2.2 Related Work 65

3.3 Methodology and Algorithm 67

3.3.1 Document Digitization and OCR 67

3.3.2 Alignment 68

3.3.3 Building the Word Hypothesis Lattice 78

3.4 Results and Analysis 79

3.4.1 A* Alignment Results 79

3.4.2 Word and Lattice Error Rate Results 81

3.5 Conclusions and Future Work 87

4 Error Correction with In-Domain Training Across Multiple OCR System Outputs 90

4.1 Introduction 90

4.2 Approach 90

4.2.1 Background 90

4.2.2 Related Work 91

4.3 Methodology and Results 92

4.3.1 Baseline OCR Results 92

4.3.2 Features of the Aligned OCR Text 93

4.3.3 Untrained Single Feature Success Rates 94

4.3.4 Combining Features 95

4.3.5 Decision List Training 97

4.3.6 Combined Features Results With Training 99

4.4 Conclusions and Future Work 99

4.5 Addendum 101

5 Progressive Alignment and Discriminative Error Correction for Multiple OCR Engines 102

5.1 Introduction 103

5.2 Related Work 104

5.3 Methods and Results on the Eisenhower Communiqués 104

5.3.1 Data 104

5.3.2 Baseline OCR 105

5.3.3 Progressive Alignment 105

5.3.4 Lattice Word Error Rates 106

5.4 Machine Learning Methods and Results with Out-Of-Domain Training 108

5.4.1 Enron Synthetically Generated Data Set 108

5.4.2 Training Maximum Entropy Model Using the Enron Data Set 109

5.4.3 Machine Learning System Results 111

5.5 Conclusions 113

5.6 Addendum 114

6 How Well Does Multiple OCR Error Correction Generalize? 116

6.1 Introduction . . . 117

6.2 Related Work . . . 118

6.3 Methodology . . . 120

6.3.1 Corpora . . . 122

6.3.2 Progressive Alignment . . . 124

6.3.3 Assigning Features in an Alignment Column . . . 124

6.4 Results . . . 126

6.4.1 Baseline Results . . . 127

6.4.2 New Features . . . 128

6.4.3 Results of the New Features . . . 130

6.4.4 Results on a Different Test Corpus . . . 133

6.4.5 Results on New Training Corpora . . . 135

6.4.6 Results of Sweeping the Size of the Training Sets . . . 135

6.5 Conclusions . . . 137

6.6 Addendum . . . 137

7 Combining Multiple Thresholding Binarization Values to Improve OCR Output 138

7.1 Introduction . . . 139

7.2 Related Work . . . 139

7.2.1 Optical Character Recognition . . . 139

7.2.2 Binarization . . . 141

7.2.3 Alignment . . . 142

7.2.4 Ensemble Methods for Error Correction . . . 143

7.3 Method . . . 144

7.3.1 Document Corpus . . . 144

7.3.2 Binary Thresholding . . . 145

7.3.3 Optical Character Recognition and Word Error Rate . . . 146

7.3.4 Alignment and Hypothesis Lattice . . . 146

7.4 Results . . . 149

7.4.1 Baseline Results . . . 150

7.4.2 Lattice Word Error Rate . . . 150

7.4.3 Committed Word Error Rate . . . 153

7.5 Conclusions . . . 154

7.6 Addendum . . . 154

8 Why Multiple Document Image Binarizations Improve OCR 156

8.1 Introduction . . . 157

8.2 Related Work . . . 158

8.2.1 OCR Error Correction . . . 158

8.2.2 Text Alignment . . . 159

8.2.3 Ensemble Methods . . . 160

8.2.4 Image Binarization . . . 161

8.3 Method . . . 162

8.3.1 Document Corpus . . . 162

8.3.2 Grayscale Documents and Thresholding . . . 163

8.3.3 Merging the OCR of Multiple Binarizations . . . 165

8.4 Results . . . 167

8.4.1 High WER Contributions . . . 169

8.4.2 Value of Multiple Binarizations . . . 172

8.5 Conclusions and Future Work . . . 174

8.6 Addendum . . . 175

9 Conclusion 176

9.1 Future Work . . . 178

A Effects of Space Propagation 180

A.1 Introduction 180

A.2 Method 180

A.3 Results 181

A.4 Conclusions 181

References 183

List of Figures

1.1 A library catalog record as a surrogate to the book itself 2

1.2 OCR on poor quality text 8

1.3 The general process from document to OCR text output. 9

1.4 Comparison of character and word error rates 10

1.5 The document to OCR output process including explicit binarization. 12

1.6 The correction process with multiple OCR engines 14

1.7 Aligned sequence hypotheses 15

2.1 Comparison of OCR word error rate range to the lattice word error rate 42

2.2 Document counts of uncorrected word error rates from the Eisenhower Communiqués 54

3.1 Poor quality text from Eisenhower Communiqué No. 237 63

3.2 OCR Results from Figure 3.1 63

3.3 Eisenhower Communiqué 23 63

3.4 Eisenhower Communiqué 368 64

3.5 Word error rate calculations 65

3.6 Text sequence alignment as a network problem 69

3.7 Alignment from Figure 3.6 70

3.8 Reducing a 3-dimensional alignment to three 2-dimensional alignments 71

3.9 Reverse Dijkstra Method for network exploration 73

3.10 Building the lattice from aligned sequences 78

3.11 Constructed word lattice from Figure 3.10 79

3.12 Comparison of the percent of nodes visited in A* alignments 80

3.13 Comparison of the number of nodes visited in A* alignments 82

3.14 Comparison of the OCR word error rate to the percent of nodes visited in an A* alignment 83

3.15 Comparison of the lattice error rate to the range of OCR word error rates 85

3.16 Comparison of the range of the OCR word error rates from multiple engines to the aligned lattice word error rate 86

3.17 Comparsion of the OCR word error rate, the lattice word error rate, and the word error rate of the dictionary-based word selection 88

4.1 Poor quality text from Eisenhower Communiqué No. 233a along with OCR output 91

4.2 Aligned output of the OCR engines 94

4.3 Word error rate on the training set 97

4.4 Alignment with features and scores 98

4.5 Comparison of the lattice word error rate, dictionary-based selection word error rate, and error rate using all features 100

4.6 Word error rates with Voting as a single feature 100

5.1 The word error rates of the five OCR engines for Eisenhower Communiqués and Enron corpora 103

5.2 Aligned text sequenes 107

5.3 Effect of the order of alignment on the lattice word error rate 108

5.4 Features extracted from a single hypothesis column 110

5.5 Word error rate decreases as OCR sequences are added 112

5.6 New low word error rates as OCR sequences are added to the alignment 114

6.1 Flowchart for creating a training set and correcting OCR errors 121

6.2 Eisenhower Communiqué No. 1 122

6.3 A fragment from document BM_24May1859_p2_c3 of the 19th Century Mormon Articles Newspaper Index 123

6.4 Aligned lattice and features from document CT_2Apr1872_p2_c1 of the 19thCMAN Index 125

6.5 The frequency of tokens with a given occurrence 129

6.6 Comparing the word error rate of the Eisenhower set as the proportion of the training set varies 136

6.7 Comparing the word error rate of the 19thCMAN set as the proportion of the training set varies 136

7.1 Comparison of character and word error rates 140

7.2 Original grayscale image and two adaptive binarizations 142

7.3 Original grayscale image with multiple threshold binarizations 147

7.4 OCR word error rates of threshold binarized images 148

7.5 Grayscale image with alignments 148

7.6 Lattice word error rates as the number of sequences increases 151

7.7 Distribution of documents error rates 152

7.8 Word error rate after committing to a single hypothesis 152

8.1 Grayscale image of of document BM_24May1859_p2_c3 158

8.2 Seven threshold binarizations 164

8.3 Word error rate of threshold binarizations 165

8.4 Distribution of grayscale values in an image 166

8.5 Distribution of black and white modes 166

8.6 Grayscale image with the aligned lattice of multiple binarizations 168

8.7 The distribution of the lowest word error rate binarization for all documents 169

8.8 The distribution of the binarization thresholds with the lowest word error rate . . . 170

8.9 The distribution of word error rates for each binarization 171

8.10 The percent of tokens seen in the OCR of binarizations 172

8.11 The occurrence of correct tokens in higher word error rate binarizations 173

A.1 Example of none of the aligned hypotheses being correct 181

A.2 An example of space propagation 181

List of Tables

1.1 Eisenhower Communiqués baseline document OCR word error rates 6

1.2 OCR word error rates showing variation in performance 6

1.3 Eisenhower Communiqués OCR word error rates compared to lattice word error rates 18

2.1 Comparison of character vs. token alignment 29

2.2 Word error rates compared to lattice word error rates on the Eisenhower Communiqués 49

3.1 Document word error rates 68

3.2 Comparison of the proportion of nodes visited by algorithm 81

3.3 Comparison of OCR word error rates to the lattice word error rate and dictionary-based word selection 84

3.4 Largest and smallest differences between baseline OCR WER and the lattice error rate 84

3.5 Differences between baseline OCR word error rates and the lattice error rates . . . 87

4.1 Baseline error rates 93

4.2 Success rates on the training set 95

4.3 Potential success rates of combinations of features 96

4.4 Comparing baseline error rates to Vote and Dictionary features 98

5.1 Baseline word error rates for the OCR engines and corpora 106

5.2 Improvements to the lattice word error are as OCR outputs are added 107

5.3 Word error rate reduction as OCR outputs are added 112

5.4 Percentage of documents reducing word error rate as OCR outputs are added . . . 113

6.1 Baseline word error rates for two test and two training corpora 120
6.2 Assignments of recurring tokens to the Recurring feature 129
6.3 The grouping of features used in various model configurations 131
6.4 Eisenhower test set word error rates with the Enron training set including the new features 132
6.5 Results on the Eisenhower and the 19th Century Mormon Article Newspaper Index test sets 134

7.1 Word error rates on the 19th Century Mormon News Articles image corpus 145
7.2 Word error rates on all binarizations of the corpus 150
7.3 Word error rates of multiple threshold binarizations 151
7.4 Summary of results 154

8.1 Word error rates from the 19th Century Mormon Article Newspaper Index 157
8.2 19thCMAN word error rates 163
8.3 Number of tokens correctly identified in higher word error rate binarizations 173

A.1 Lattice results of applying space and closure propagation 182

Chapter 1

Introduction

Researchers estimate that between 50 and 200 million books have been published [106], of which a small fraction are available digitally. Major digitizing efforts, such as the Google Book Project[1], the Hathi Trust[2], the Internet Archive[3], and the European IMPACT Project[4] are making pre-digital era materials available online at an unprecedented rate. However, simply placing page images in a digital repository does not fulfill the need for accessibility nor does it exploit the full value of the texts. In order to be accessible, all documents, digital or print, require either an abbreviated representation, such as a library catalog record (see Figure 1.1), or the full text of the document.

The full-text of digital documents is both flexible and accessible, adapting to the needs of the reader in various ways as indicated by Rose and Meyer (2002) [89]. Examples include:

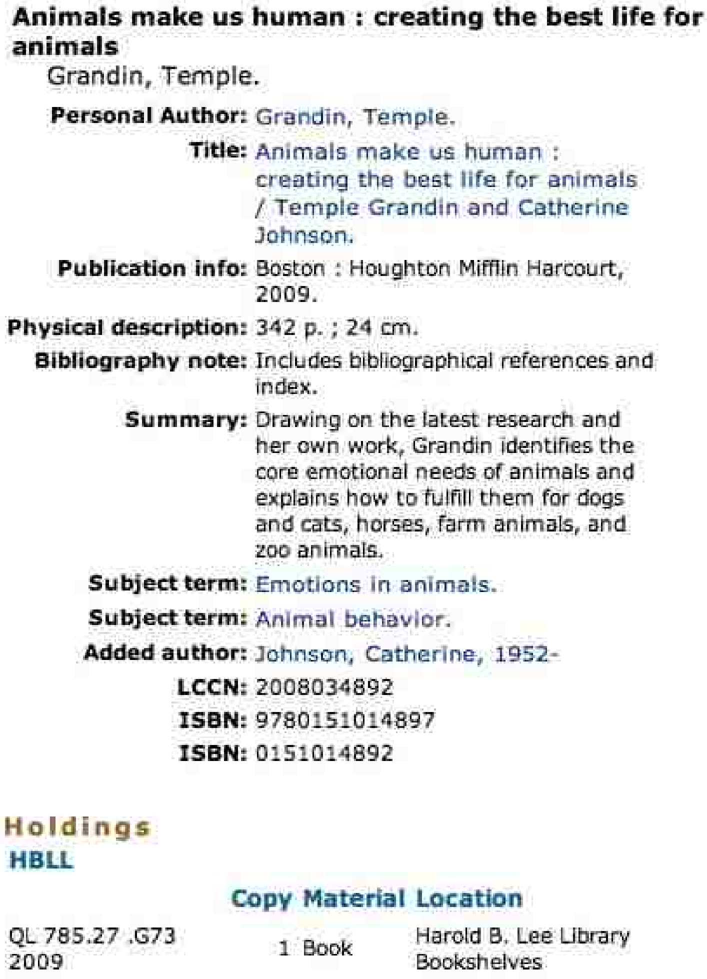
- Automated creation of descriptive metadata,
- Full-text searching of the entire document,
- Topic modeling,
- Automated cataloging,
- Information extraction,
- Conversion to Braille for visually-impaired readers,
- Synthesized speech for learning disabled or visually-impaired readers,

[1] http://books.google.com

[2] http://www.hathitrust.org

[3] http://www.archive.org/details/texts

[4] http://digitization.eu



Animals make us human : creating the best life for animals
Grandin, Temple.

Personal Author: Grandin, Temple.
Title: Animals make us human : creating the best life for animals / Temple Grandin and Catherine Johnson.
Publication info: Boston : Houghton Mifflin Harcourt, 2009.
Physical description: 342 p. ; 24 cm.
Bibliography note: Includes bibliographical references and index.
Summary: Drawing on the latest research and her own work, Grandin identifies the core emotional needs of animals and explains how to fulfill them for dogs and cats, horses, farm animals, and zoo animals.
Subject term: Emotions in animals.
Subject term: Animal behavior.
Added author: Johnson, Catherine, 1952-
LCCN: 2008034892
ISBN: 9780151014897
ISBN: 0151014892

Holdings
HBLL

	Copy	Material	Location
QL 785.27 .G73 2009	1	Book	Harold B. Lee Library Bookshelves

Figure 1.1: An example of a catalog record as a surrogate to the text itself. Although accurate in the information provided, much more information may be extracted from the full-text of the document. For instance, the Anglo-American Cataloguing Rules [1] stipulate that in order for a subject heading to be included in the catalog record for an item, the subject must comprise at least 20% of the material in the item. Automated analysis of the full-text, in contrast, may result in a much richer set of subjects.

- Dictionary support for looking up unfamiliar or foreign language terms,
- Print size adjustment for the visually impaired,
- Machine translation,
- Automated metadata creation or summarization to enable document discovery and access, and
- Managing copyright and digital rights of the document.

Given the number of undigitized documents in existence, automated methods must be found to make these documents accessible and usable [52]. Without either abbreviated representations or full transcriptions, there is little point in digitizing a corpus of printed documents in the first place. The usefulness of digitized documents is directly related to the quality of the extracted text. Optical Character Recognition (OCR) has reached a point where well formatted and clean machine-printed documents, such as this dissertation, are easily recognizable by current commercial OCR products; however, older or degraded documents present problems with word error rates (WER) that severely limit either automated or manual use of the text. Dordevic and Mihajlov [22] state: "Despite the fact that many satisfactory results have been reported for the OCR problem, there is still ... room for improvement especially in the cases of recognition of poorly printed and damaged documents."

1.1 Thesis Statement

The word error rate of the output from OCR engines can be reduced through the use of ensemble methods with diverse information sources, novel features of multiple sources, multiple OCR engines, and multiple binarizations of digital document images.

1.2 Contributions

The goal of the research presented in this dissertation is to develop methods to improve the quality of OCR output without human intervention, particularly for historical[5], degraded, machine printed documents and to understand the generalized conditions under which improvement is possible. The contributions of this work include:

- A novel approach to providing diverse input to ensemble methods using multiple threshold image binarizations, which are superior to single adaptive image binarizations (Chapters 7 and 8),
- New insight into the value of high WER inputs to the ensemble when they present diverse error corrections not found in other inputs; leading to the observation that diversity is more important than non-diverse higher quality inputs (Chapters 5 and 8),
- A new heuristic for use with the A* algorithm for optimal alignment, which reduced the number of nodes visited by two to three orders of magnitude over Dijkstra's algorithm (Chapter 3),
- The Lattice Word Error Rate (LWER), a novel measure of the lower bound on the possible improvement given the diverse inputs to the ensemble method (Chapter 3),
- A novel approach to aligning documents at the character level but evaluating the resulting alignment as a lattice of word hypotheses (Chapters 3 and 5),
- The assignment of multiple features from both individual word hypotheses and collective hypotheses from the aligned inputs, rather than limiting the evaluation to individual characters from the input sources (beginning with Chapter 5),
- The selection of a word from the source hypotheses using an ensemble method, including the ability to select none of the sources to reduce OCR noise (beginning with Chapter 5),

[5] For the purposes of this research, historical documents come from a pre-digital era, specifically the early twentieth century and the nineteenth century. The author notes that in European contexts historical documents are often manuscript documents prior to printing.

- Novel features for use in an ensemble method to select the best hypothesis from a lattice of word hypotheses (Chapters 4, 6, and 7),
- Small training sets, randomly selected from large training sets can be as effective as the complete training set (Chapter 6), and
- New test corpora of digitized printed documents with gold standard transcriptions for research into OCR error correction (Chapters 3 and 7).

These methods for creating diverse inputs for ensemble methods are shown to improve the OCRed text beyond what is available from state-of-the-art commercial and open source OCR engines.

1.3 Research Overview

The remainder of this chapter is written primarily for an audience unfamiliar with the specifics of post-OCR error correction, giving a high level overview of the technology and research. A more technical and detailed review of the research background and related work can be found in Chapter 2.

This research explores how variations in the output among multiple OCR engines and in image binarizations, along with other external knowledge sources can be leveraged to improve the quality of the text output. This research also contributes to an understanding of the underlying mechanisms which effect the improvement.

Only methods which do not require a human in the loop are considered. This decision is based on the intended long-term use of the results to process large corpora where human intervention document by document would be cost prohibitive. For small collections, page-by-page correction of the scanned image is possible, manipulating the scanning parameters to minimize background noise and improve the contrast between the background and foreground text or manually correcting OCR errors after the fact. However, this work aims at methods that are applicable for large collections, on the order of hundreds of thousands or millions of images, where individual correction is not feasible. In these circumstances no manual per-document scanning parameter optimization or OCR

Table 1.1: Baseline mean document word error rates by OCR engine on the Eisenhower Communiqués. WERs greater than 100% are possible when the OCR engine misinterprets noise in the digital document image as text to be recognized or when individual word tokens are split in error.

	Abbyy	OmniPage	Adobe	ReadIris	Tesseract
Mean	18.2%	30.0%	51.8%	54.6%	67.8%
	Average WER across all OCR engines: 44.5%				
Minimum	1.9%	1.5%	2.4%	2.4 %	2.0%
Maximum	84.7%	112.7%	151.2%	206.8%	1017.1%

Table 1.2: Individual WERs, showing the variation in performance between OCR engines. The best WER for each document is highlighted. The documents are from the Eisenhower Communiqués corpus.

	Word Error Rates				
Document No.	Abbyy	Omnipage	Tesseract	ReadIris	Adobe
78	**7.3%**	25.2%	17.2%	16.6%	23.8%
229	20.3%	36.8%	**15.7%**	57.9%	57.9%
260	21.4%	**16.4%**	44.3%	46.2%	50.0%

optimization is performed, other than what may be applicable to a collection as a whole. Further, the OCR engines will be treated as black boxes, a decision which is discussed in more detail in Section 2.2.2 of Chapter 2. Document segmentation, table recognition, and image recognition are beyond the scope of this work.

Individually, OCR engines can exhibit wide variations in their performance as observed by Klein and Kopel [49]. Table 1.1 shows the word error rate performance for five OCR engines on the Eisenhower Communiqués [43], a collection of press releases from the last year of World War II in Europe. The documents were typed and duplicated using the then available duplication methods and vary significantly in their quality as can be seen by the minimum and maximum WERs shown in the table. Further, even in a single document there can be significant differences in the OCR WERs of different engines as shown in Table 1.2. Although Abbyy FineReader is in general superior to Omnipage Pro and Tesseract, it is not always the case for every document. Each OCR engine is able to recognize some text that the others cannot.

The approach implemented in this research to the problem of extracting useful text from degraded or poor quality machine-printed documents uses ensemble methods for combining evidence collected from various sources. In this work, evidence is collected either from multiple OCR engines, which are treated as black box systems with external parameters that control their operation, or from multiple threshold binarizations of the same grayscale document image. (See Chapter 7.)

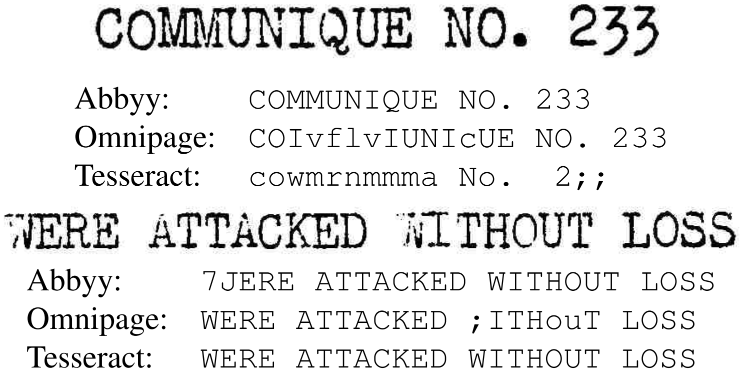
The rest of this section describes the end-to-end process from preparing the datasets to the final evaluation of the corrected OCR.

1.3.1 Corpora

Selecting the corpora for investigation required finding both test and training (or calibration) sets for which a gold standard text transcription is available. Four corpora are used in this research:

- **Eisenhower Communiqués** [43] consisting of 600 bi-tonal images of facsimiles of mid-twentieth century typewritten documents with transcriptions
- **Nineteenth Century Mormon News Articles** [29] consisting of 1074 grayscale images and transcriptions of newspaper articles from nineteenth century American publishers.
- **Enron Synthetic Dataset** consisting of digital images and original text from the Enron email dataset [8] from LDC constructed in collaboration with Dr. Daniel Walker [108]
- **Reuters 21578 Synthetic Dataset** consisting of digital images and original text from the Reuters 21578 dataset [55] from LDC constructed in collaboration with Dr. Daniel Walker [108].

The Eisenhower Communiqués and the 19th Century Mormon Articles Newspaper Index consist of bi-tonal or grayscale images and a human corrected gold standard text that are used either for training or a machine learner or to evaluate the results of the processes. Both the Eisenhower Communiqués and the Nineteenth Century Mormon Article Newspaper Index are available to be viewed.



COMMUNIQUE NO. 233

Abbyy:	`COMMUNIQUE NO. 233`
Omnipage:	`COIvflvIUNIcUE NO. 233`
Tesseract:	`cowmrnmmma No.  2;;`

WERE ATTACKED WITHOUT LOSS

Abbyy:	`7JERE ATTACKED WITHOUT LOSS`
Omnipage:	`WERE ATTACKED ;ITHouT LOSS`
Tesseract:	`WERE ATTACKED WITHOUT LOSS`

Figure 1.2: Poor quality text from Eisenhower Communiqué No. 233a [43] along with output from three OCR engines.

1.3.2 Document Scanning

OCR requires a digital document image from which the text of the image is extracted into a digital text form. The document may have degraded with time, may have been damaged, or may contain noise introduced during creation or reproduction. The OCR process is a deterministic software process and always results in the same OCR output given the same parameters and document image file, whereas it is possible for two scans of the same document with identical document placement and software parameters to result in different digital images due to scanner lamp temperature differences and other analog factors. Figure 1.2 shows a portion of a reproduction of a typewritten document with the OCR output from three different OCR engines.

1.3.3 Optical Character Recognition

Generally speaking, the OCR task consists of multiple sub-tasks including: 1) zoning images and non-textual elements of a page, 2) normalizing (potentially binarizing) the image that will be processed by the OCR engine, and 3) segmenting and recognizing characters, words, and paragraphs. Figure 1.3 gives a general outline of the process from original degraded historical document through scanning and OCR resulting in the output of digital text.

Common processes used in OCR are feature extraction from segmented characters, calculation of posterior probabilities of characters given the image features, and dynamic programming to

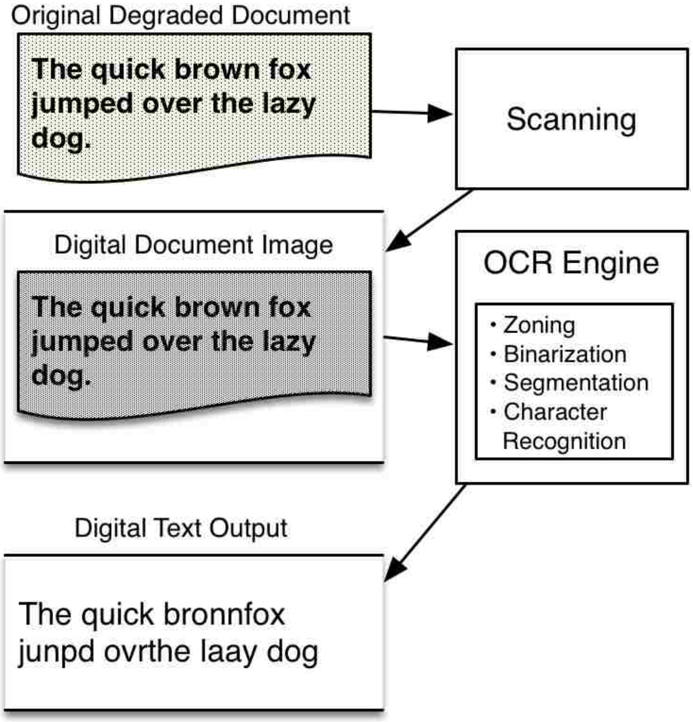

Original Degraded Document
The quick brown fox jumped over the lazy dog.
Scanning
Digital Document Image
The quick brown fox jumped over the lazy dog.
OCR Engine
• Zoning
• Binarization
• Segmentation
• Character Recognition
Digital Text Output
The quick bronnfox junpd ovrthe laay dog

Figure 1.3: The general process from document to OCR text output.

select from the best character or word hypotheses, often using a word or language model. The OCR output can include as little as a plain text transcription of the document up to capturing specific character fonts and document layout represented in a digital form.

Printing and duplication techniques up through the mid-twentieth century create significant problems for OCR engines, even for those trained on the documents' fonts. Examples of problematic documents include typewritten text, in which letters are partially formed, typed over, or overlapping, and documents duplicated by mimeographing, carbon paper, or multiple iterations of photographic copying common in the mid-twentieth century. Referring to Figure 1.2 note the variation in OCR output. Few of the words are correct across all documents; however at least one OCR engine is correct for each word. Current state-of-the-art OCR engines do well with modern machine-printed documents, particularly those printed with laser printing technology or other high quality printing techniques; however, there are significant problems with degraded historical printed documents.

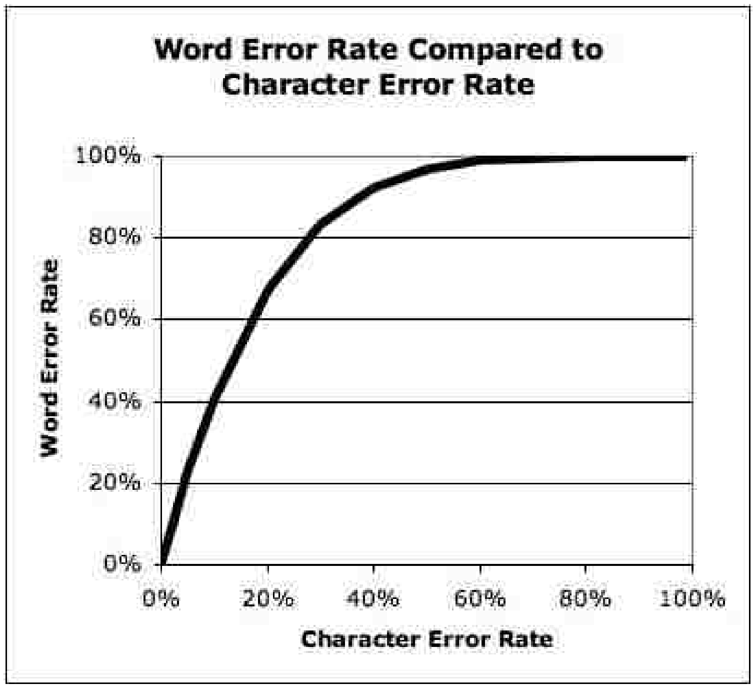


Figure 1.4: Comparing character to word error rates, assuming an average word length of 5 characters.

Extracting usable text using OCR from older, degraded machine-printed documents is often unreliable, frequently to the point of being unusable [4]. Even in situations where a fairly low character error rate is achieved, Hull [38] points out that a 1.4% character error rate results in a 7% word error rate on a typical page of 2,500 characters and 500 words (see Figure 1.4). Kae and Learned-Miller [44] remind us that OCR is not a solved problem and that "the goal of transcribing documents completely and accurately... is still far off." The word error rate of the OCR output can inhibit the ability of the user to accomplish useful tasks. In an automated speech recognition task, Munteanu et al. [74] determined that when creating text transcriptions of lectures, a transcript with a WER of 50% (which would be approximately a 12.9% character error rate) was no better than having no transcript at all.

This research uses five OCR engines:

- **Abbyy FineReader:** Version 10 for Windows, acknowledged as the state of the art for individual commercial OCR engines.
- **Tesseract** Version 3 on Linux, an open source OCR engine.

- **Adobe Acrobat X** for Windows.
- **OmniPage 18** for Windows.
- **ReadIris** Version 12 for Windows.

The output of the OCR engines is normalized as follows. Some engines produce ligatures using UTF-8 multi-byte characters instead of two ASCII 8-bit characters, requiring that UTF-8 multi-byte characters be converted to appropriate ASCII characters. On occasion, OCR engines output UTF-8 characters that have no correspondence in the ASCII 8-bit character set, requiring either conversion to an acceptable ASCII equivalent (e.g., a UTF-8 long space to a standard ASCII space) or, if no ASCII equivalent is available, to an ASCII character not found in the corpus signifying an unknown character. For the corpora in this research the unknown character is the tilde ($\sim$), which is not found in any of the documents.

1.3.4 Document Digitized Image Binarization

Binarization of an image consists of converting the individual grayscale or color pixels of the image into either black (value 0) or white (value 1). This is an implicit step often performed in an OCR engine. It is beyond the scope of this work to specifically manipulate hardware or software scanning parameters, other than to use generally accepted parameters that meet the needs of a collection as a whole. However in Chapters 7 and 8 this work evaluates the use of uniform binarization or thresholding. Uniform binarization takes a single threshold parameter and applies it to the entire image. By varying the binarization threshold parameter both under- and over-exposed regions are made clearer as the threshold parameter changes. Each binarization results in a separate digital image file as input for the OCR engines. The OCR engines can then recognize text in one binarization that is not visible in another. Figure 1.5 indicates the step in which explicit binarization occurs.

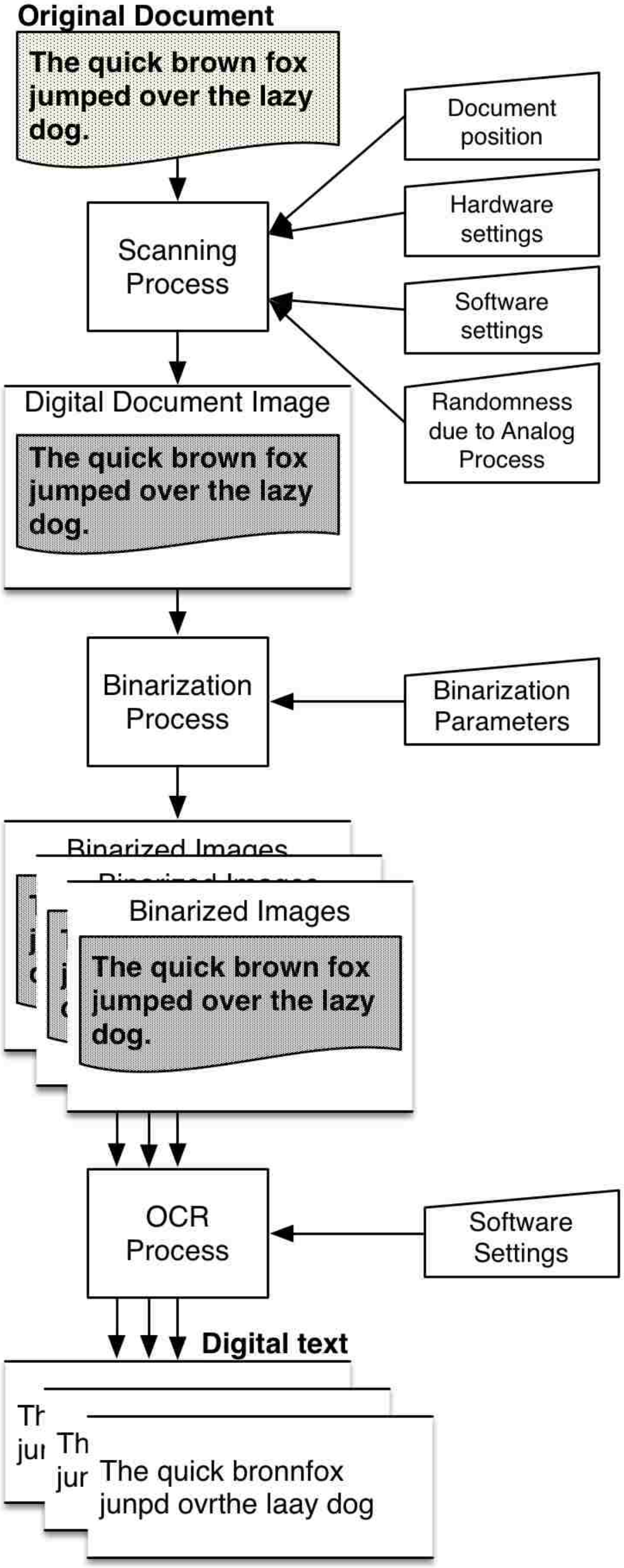


Figure 1.5: The document to OCR output process including explicit binarization.

1.3.5 Ensemble Methods

Ensemble recognition methods are used effectively in a variety of problems such as machine translation, speech recognition, handwriting recognition, and OCR error correction, to name a few. In a paper on a framework for ensemble classification methods, Kittler et al. state: "It had been observed ... that although one of the [classifiers] would yield the best performance, the sets of patterns misclassified by the different classifiers would not necessarily overlap. This suggested that different classifier designs potentially offered complementary information about the patterns to be classified which could be harnessed to improve the performance of the selected classifier." [48] This observation is behind the success of ensemble recognition methods, specifically, that multiple systems which are complementary can be leveraged for an improved combined output. The complementarity of correct responses of the methods is critical. One of the contributions of this work is a variation on the observation by Kittler et al. regarding multiple classifiers: using the same commercial OCR engine with multiple varied binarizations of the document image, a similar ensemble method can produce desirable results. (See Chapters 7 and 8 on multiple binarization methods.)

A more complete discussion of ensemble methods is presented in Chapter 2.

1.3.6 Text Alignment, Lattice Creation, and Hypothesis Segmentation

Required for any post-OCR error correction using multiple sequence sources is an alignment of the sequences. An alignment algorithm can be either exact (and optimal) or heuristic (and approximate). The purpose of alignment is to bring together competing hypotheses from various input sources, such as different OCR outputs of the same digital image. (See Figure 1.6 for an outline of the OCR error correction process including alignment.) The majority of multiple sequence alignment work is done in the field of bioinformatics, where the size of the alignment problems has forced the discovery and adoption of heuristic solutions such as progressive alignment.

Given the character aligned OCR sequences the next step is to segment parallel hypotheses. In order to align characters it is necessary to insert gaps, sometimes call an *INDEL* for "insert

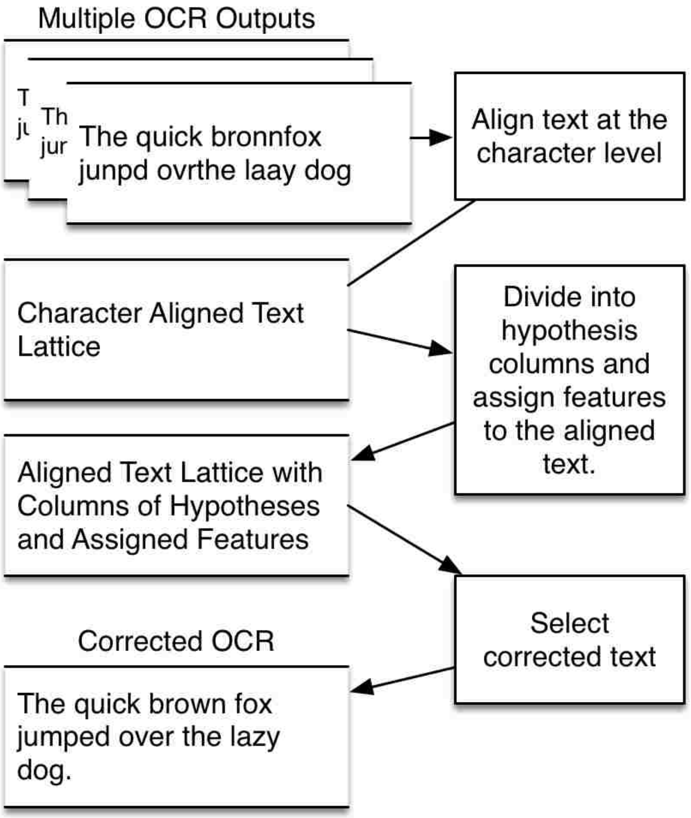
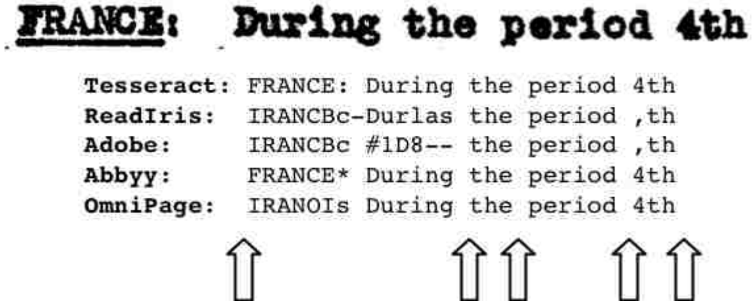

Multiple OCR Outputs
T
ju
Th
jur
The quick bronnfox
junpd ovrthe laay dog
Align text at the character level
Character Aligned Text Lattice
Divide into hypothesis columns and assign features to the aligned text.
Aligned Text Lattice with Columns of Hypotheses and Assigned Features
Select corrected text
Corrected OCR
The quick brown fox jumped over the lazy dog.

Figure 1.6: Given multiple OCR outputs of the same document, this figure shows the process for aligning the text, creating columns of hypotheses and features, and selecting post-OCR corrections.



```
Tesseract: FRANCE: During the period 4th
ReadIris:  IRANCBc-Durlas the period ,th
Adobe:     IRANCBc #1D8-- the period ,th
Abbyy:     FRANCE* During the period 4th
OmniPage:  IRANOIs During the period 4th
```

Figure 1.7: From Periodical Communiqué No. 1 of the Eisenhower Communiqués, an example of aligned sequence hypotheses from five OCR engines. The arrows indicate points of agreement on white space among the aligned sequences. The text between an adjacent pair of arrows constitutes an aligned column of hypotheses. The hyphen ("-") character in a sequence represents a gap aligned with characters in the other sequences.

or delete" into the alignment where characters found in one sequence are not found in the other sequences. Referring to Figure 1.7, an example of a gap, represented by two hyphen ("-") characters, is necessary to align the text "During" from Tesseract, Abbyy, and OmniPage with the text "#1D8" from Adobe. Since the text from Adobe is only four characters long, two gaps or *INDEL*s are required in order align the character strings.

One way to segment the aligned texts is to create "columns" of hypothesis segments across all aligned sequences where there is complete agreement on spaces. See Figure 1.7 where four columns of hypothesis segments are identified. Some columns are simple in that each sequence within a column only contains a single token. Other columns are more complex in that some sequences contain multiple tokens, such as the first column in the figure. These present opportunities for space propagation discussed in Section 1.3.7.

A fundamental difference between a lattice of OCR outputs and other natural language tasks is that the OCR is known to be inaccurate, at least for the corpora included in this research. A lattice in the speech recognition and machine translation tasks consists of hypothesized words as proposed in Stolcke, König, and Weintraub [98]. The resulting lattice contains words from a defined vocabulary whereas the OCR lattice may contain tokens based on document noise as well as misrecognized characters.

In Chapters 3 and 4 exact alignment methods are used. Subsequently, to handle larger documents non-optimal, progressive alignment techniques are used which are discussed in detail in Chapter 5.

1.3.7 Space Propagation

The current literature in OCR error correction assumes that recognized characters may be in error, but makes the assumption that the divisions between tokens, or white space, is accurate. Considering all possible locations of space insertions and deletions is a combinatorial problem and beyond the capacity to be computed in reasonable time for all possible locations of spaces within a document. A solution to this problem is to consider within the aligned lattice where a space may appear in one sequence but not in the others. Both propagating these spaces to other sequences within the lattice and removing spaces that are absent in other sequences, consistent with the character alignment of the lattice, can reveal instances where spaces have either been incorrectly inserted or deleted. This is discussed in more detail in Appendix A.

1.3.8 Lattice Features and Scoring

Within each hypothesis column features are assigned, which are subsequently used to select among the alternatives provided by the OCR engines. The method employs modern supervised discriminative machine learning methods trained on one of the synthetic datasets as a calibration set. The role of the learned model is to select the proper hypothesis from each aligned column in order to produce the best OCR correction. The model uses the following features:

- **Lexical Features** The token itself is the lexical feature.
- **Number** A binary value indicating whether the token looks like a number.
- **Dictionary** A binary value indicating whether the token appears in a general or domain specific dictionary, personal name list, or gazetteer.
- **Voting** A numeric value indicating how many tokens within the hypothesis column agree.

- **Spell Checker** A numeric value. If the token does not appear in the dictionaries, it is sent to a general purpose spell checker for a suggestion. The numeric value indicates how many tokens have the same spell checker result.
- **No Selection:** A "select none" outcome when none of the hypotheses is correct. This allows the machine learner during training to reject all of the hypotheses within the column. This is helpful in eliminating garbage OCR which may correspond to non-word elements or noise in the document.
- **Recurring Tokens**. No matter the size of the dictionary, it is possible that correct tokens in OCRed text may not be found in the dictionary. Recurring tokens in the corpus are a feature to identify correct tokens not found in a dictionary, yet found in the corpus multiple times.

Chapter 6 explores the extent to which the lexical feature training can be removed from the calibration set, which will make the generalization ability of the model trained from a synthetic dataset stronger.

1.3.9 Hypothesis Selection

Once all of the feature vectors have been extracted, the method here uses the maximum entropy learner in the Mallet [72] toolkit to train a maximum entropy (a.k.a., multinomial logistic regression) model to predict choices on unseen alignment columns. The model is used to select the most likely word hypothesis within the column of possibilities, or reject all hypotheses.

1.3.10 Validation with the Word Error Rate

Validating the results of this research takes two forms: determining whether the ultimate WER is reduced and understanding the reasons behind why the WER is reduced. The WER is calculated using Sclite from NIST [3]. In the literature both character error rates and word error rates are used to indicate the improvement in underlying OCR results. This research uses the word error rate since the goal of this work is to improve the accessibility of degraded machine-printed documents whose WER limits their usefulness. Expanding on Hull's [38] observation that a given character error rate

Table 1.3: Eisenhower Communiqué baseline corpus mean error rates compared to Lattice Error Rate and the results of Chapter 5.

Abbyy	OmniPage	Adobe	ReadIris	Tesseract	LWER	Ensemble WER
18.24%	30.02%	51.78%	54.64%	67.78%	9.27%	13.76%

results in a higher word error rate, Figure 1.4 in Section 1.3.3 shows the rate of change between the character and word error rates. Based on these observations, this research uses a word error rate as the basis for calculating the value of the solution, which agrees with Kolak et al. [52] who primarily use word error rate for output to NLP tasks.

This work takes two approaches to reporting WER results. The first, a document centric approach, calculates the WER of each document individually, with each document contributing equal weight to the result. This method values the quality of each document in the collection separately, regardless of whether it is long or short. This document centric method is used in Chapters 3, 4, and 5. In the remaining chapters, the entire corpus is considered a single document, with the focus on reducing the overall word error rate. Here each recognized token, whether correct or incorrect, contributes equal weight to the overall result.

There are two ways of demonstrating the value of the solution presented here. The first is to compare the resulting word error rate of the system with the original word error rates of the input OCR engines. Our current published results, shown in Table 1.3, show an improvement from 18.2% for the best OCR engine to 13.8%. The second method for demonstrating the system's effectiveness is to compare the resulting word error rate with a lattice word error rate. This is a lower bound on the WER given the evidence available from the OCR sequences, also shown in Table 1.3.

Ultimately, the only true measure of value in OCR error correction is if there is an improvement in how the results can be used by downstream tasks. Rose and Meyer [89] discuss the power of digital media as being versatile, transformable, sharable, and personalizable.

1.4 Dissertation Organization

The remaining chapters are arranged as follows.

Chapter 2 discusses the background technology and work that is related to this research and is intended for an audience familiar with this research field.

Chapter 3 introduces the concept of using multiple sources of text, specifically multiple OCR engines, to extract text from a digitized document which results in a lower or equal WER than in any of the individual OCR sources alone. Additionally, this chapter introduces a novel admissible heuristic used in conjunction with the A* minimum path algorithm for finding the optimal alignment of multiple text sequences. This heuristic, named the Reverse Dijkstra Heuristic, is able to significantly reduce the calculation time of an exact alignment. Lastly, this chapter introduces the Eisenhower Communiqués [43] as a test set for evaluating the effectiveness of these techniques. This chapter was published previously in the Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries in 2009 (JCDL 2009) [61].

Chapter 4 extends the results of Chapter 3 to explore the effectiveness of a decision list, trained on an in-domain training set in reducing OCR word error rates. The aligned text features used in this chapter are carried forward throughout the remainder of this work. The effectiveness for identifying correct hypotheses from the aligned text is demonstrated. This chapter was published previously in the Proceedings of the 11th International Conference on Document Analysis and Recognition in 2011 (ICDAR 2011) [63].

Chapter 5 introduces the use of machine learning in the form of maximum entropy and conditional random fields in the problem of identifying correct hypotheses from the aligned lattice of text sequences. Additionally, this research uses a progressive alignment heuristic rather than exact alignment introduced in Chapter 3. Progressive alignment allows more text sequences to be used in the alignment, improving the results. Importantly, this chapter also introduces the use of a synthetic training set, allowing the generalization of these results to cases where no in-domain training set is available. This chapter was also published previously in the Proceedings of the 11th International Conference on Document Analysis and Recognition in 2011 (ICDAR 2011) [64].

Chapter 6 generalizes the results of the previous chapters, additionally introducing a second synthetic training set incorporating grayscale images. One of the contributions of this chapter is an evaluation of the size of the training set required to achieve good results. This chapter appears in the Proceedings of the Document Recognition and Retrieval Conference in 2014 (DRR 2014) [67].

Chapter 7 introduces the use of multiple binarizations of a document image as input to a single OCR engine, in contrast to using a single document image recognized by multiple OCR engines. This chapter also introduces the use of a second test set, the Nineteenth Century Mormon Articles Newspaper Index [29]. This chapter was published previously in the Proceedings of the Document Recognition and Retrieval Conference in 2013 (DRR 2013) [65].

Chapter 8 explores the mechanisms behind the results found in Chapter 7. This chapter shows how diversity among multiple binarizations of the same document image make improvements to the OCR possible. The analysis reveals that sources of correct hypotheses are not limited to any one binarization and that the full range of binarizations holds information needed to achieve the best results, again emphasizing the contributions of low quality inputs if they provide diversity. This chapter was published previously in the Proceedings of the Workshop on Historical Document Imaging and Processing in 2013 (HIP 2013) [66].

Chapter 9 summarizes the conclusions of this research. And Appendix A explores a method involving the propagation of spaces across the aligned hypotheses.

Chapter 2

Background and Related Works

This chapter is intended for an audience familiar with post-OCR error reduction and related fields It covers the major technologies related to this research, how they apply, and how this research extends those previous results. The topics are covered in roughly the order they occur in the post-OCR error correction process displayed in Figures 1.5 and 1.6. A general formulation of the OCR error correction problem is presented in Section 2.1, followed by Section 2.2 discussing the general field of post-OCR error correction and the approaches that others have taken to this problem. Required for any process incorporating multiple string hypotheses as part of an ensemble is the process of alignment of those inputs, which is discussed in Section 2.3. Section 2.4 discusses the framework for ensemble methods. Diversity and its importance in ensemble methods are discussed in Section 2.5. Sections 2.6 and 2.7 discuss implementations and applications of ensemble methods respectively. Test and training corpora are required for any work in OCR improvement or post-OCR error correction and are discussed in Section 2.8 covering related work for building corpora. Scanning and OCR will both be considered black boxes per the research methods of [24, 41, 51] and will not be covered as related work since the manipulation of scanning and OCR parameters is beyond the scope of this work. However, document image binarization does play an important role; thus Section 2.9 discusses related work in binarization, including adaptive binarization. Lastly, Section 2.10 provides a conclusion to this chapter.

2.1 General Formulation

That post-OCR errors exist is a result of character misrecognition in the digital document image by the OCR engine . For the purposes of this work let d be the original pristine printed document in a physical document collection D, where

$$D = \{d_1, d_2, \ldots, d_n\} \tag{2.1}$$

is a collection of n physical documents. The original document d may include noise due to poorly formed characters (e.g. typewritten or newsprint) and typographical errors. Machine printed historical documents, specifically from the nineteenth century to the mid-twentieth century used in this research also have noise due to aging and damage. Refer to Figure 1.2 and Section 1.3.3 in Chapter 1 for an example of an image of a noisy historical document used in this work along with the resulting OCR problems. Let $d^N = N(d)$ be the printed document as it currently exists degraded by some physical process $N()$ which has introduced noise to the original, pristine document. Physical document d^N belongs to D^N, where

$$D^N = \left\{d_1^N, d_2^N, \ldots, d_n^N\right\}. \tag{2.2}$$

For a normal document, d^N is the only source we have to work with, although there are cases where multiple instances of the same printed document may exist in multiple corpora.

The evaluation of post-OCR error correction requires a gold standard, likely a manual transcription of the text in the documents D^N. Let $M(d_i^N) = t_i^M$ be the manually transcribed or corrected text of document d_i^N and $t_i^M \in T^M$ be the complete collection of manually transcribed documents. The text is as close as humanly possible to the original, pristine text t_i of the document d_i prior to the introduction of noise.

The initial step in extracting digital information from the physical document d_i^N is the digitization of the physical document to a digital image format,

$$d_i^I = Scan\left(d_i^N, \theta_S, \phi\right) \tag{2.3}$$

where θ_S are the parameters for the scanner and scanning software, ϕ is a random variable representing the noise inherent in the analog scanning process, and d_i^I is the resulting digital document image. Note that the scanning step is the only place where the outcome of the step is not deterministic given the input parameters due to the noise in the physical scanner itself. For small collections, the scanning parameters, θ_S, may be customized for each document, whereas for large collections it may be necessary to manually select a single set of parameters, θ_S^H, that apply to all physical documents in the collection or scanning run.

Given a digital document image, d_i^I, the information extraction process is

$$e_i = E\left(d_i^I, \theta_E\right) \tag{2.4}$$

where $E()$ is the process for extracting information from the printed document d_i^N and θ_E are the input parameters to the extraction process $E()$. The form of the extraction output, e_i, depends on the goal of the extraction. Examples include: segmentation information separating text from images; word matching where a glyph in the digitized image resembles a target word; tables for information extraction or data mining; mathematical equations; document structural information such as chapters, section headings, and the document title; and of particular interest to this work, text from an OCR engine.

Correction is denoted as $c_i = C(e_i, \theta_C)$ where $C()$ is the correction process taking the extracted data e_i and the configuration and input datasets θ_C. We can evaluate correction and extraction processes using an error function $Err()$ to measure the errors in their output. Selecting optimal parameters for the extractor can be represented as a discriminative error minimization process as follows:

$$\theta_E^* = \arg\min_{\theta_E} Err_E\left(E\left(D^I, \theta_E\right), T^M\right) \tag{2.5}$$

where $Err_E()$ is the error measure for extraction process $E()$, θ_E^* is an optimal configuration parameter minimizing $Err()$ and D^I and T^M represent the complete document and transcription corpora. Likewise a discriminative training process can be formulated as follows (see also Section 2.4):

$$\theta_C^* = \arg\min_{\theta_C} Err_C\left(C\left(E\left(D^N, \theta_E^*\right), \theta_C\right) T^M\right) \tag{2.6}$$

where $Err_C()$ is the error measure for correction process $C()$ and θ_C^* is the optimal configuration for the correction process. It is possible that both $Err_E()$ and $Err_C()$ are the same function such as a word error rate. It would be possible to jointly train both θ_E and θ_C together for a collection of documents D^I as

$$(\theta_C^*, \theta_E^*) = \operatorname*{argmin}_{\theta_C, \theta_E} Err_C\left(C\left(E\left(D^I, \theta_E\right), \theta_C\right) T^M\right). \tag{2.7}$$

Using the optimal parameters for extraction, θ_E^*, and correction, θ_C^*, the corrected extracted information is formulated as

$$e_i^C = C\left(E\left(d_i^I, \theta_E^*\right) \theta_C^*\right). \tag{2.8}$$

These formulations apply to individual documents. The multiple document or multiple source formulation will be covered in Section 2.4 on ensemble methods.

2.2 Post-OCR Error Correction

A post-OCR error correction process accepts input from one or more OCR engines (that are considered to be black boxes) and attempts to correct OCR errors due to misrecognition of characters and noise in the digitized document image. The correction process may include multiple OCR outputs from different OCR engines, multiple OCR outputs of variations on the same digitized document image using the same OCR engine, and various knowledge sources such as dictionaries, gazetteers, or language models.

2.2.1 Post-OCR Error Correction Formulation

Recalling the formulation for information extraction in Equation 2.4, $E()$ in the case of OCR as the extraction method may be written as

$$e_i^{OCR} = OCR\left(d_i^I, \theta_{OCR}\right) \tag{2.9}$$

where $e_i = e_i^{OCR}$, $E() = OCR()$, and $\theta_E = \theta_{OCR}$. If the OCR output e_i^{OCR} is sufficient this is the end of the information extraction process; however, the goal of this dissertation to improve upon the OCR output through post-OCR error correction processes.

Rewriting Equations 2.3 and 2.4 with the OCR and scanning processes and the inputs to the scanning, θ_S, and to the OCR, θ_O, we have

$$e_i = E\left(Scan\left(d_i^N, \theta_S, \phi\right), \theta_E\right) = OCR\left(Scan\left(d_i^N, \theta_S, \phi\right), \theta_{OCR}\right). \tag{2.10}$$

Rewriting joint training from Equation 2.7 to include scanning, OCR, and error correction yields

$$\left(\theta_S^*, \theta_{OCR}^*, \theta_C^*\right) = \underset{\theta_S, \theta_{OCR}, \theta_C}{\arg\min}\; Err_C\left(C\left(OCR\left(Scan\left(D^N, \theta_S, \phi\right), \theta_{OCR}\right), \theta_C\right), T^M\right). \tag{2.11}$$

Although it is possible to optimize for all three input parameters, $\theta_S^*, \theta_{OCR}^*, \theta_C^*$, in general a lack of programmatic instrumentation for the scanner or the OCR engine leads to a greedy choice of scanning and OCR parameters that are generally good for the entire corpus. This is particularly true for large corpora, such as newspaper collections, where individual adjustments are not feasible. Rangoni et al. [83] attempt to circumvent the problem of training the scanner and OCR engine by selecting a single best binarization of the digitized document image (to be discussed in Section 2.9).

Manually selecting generally good parameters for scanning, θ_S^M, and OCR, θ_{OCR}^M, the training is limited to the correction process as follows:

$$\theta_C^* = \arg\min_{\theta_C} Err_C\left(C\left(OCR\left(Scan\left(D^N, \theta_S^M, \phi\right), \theta_{OCR}^M\right), \theta_C\right) T^M\right) \tag{2.12}$$

where the physical documents, D^N are either existing document collections of the Harold B. Lee Library or synthetic document collections; the input scanner parameters, θ_S^M, are selected by the Digital Imaging Lab of the Lee Library; the OCR parameters, θ_{OCR}^M, are selected by the author depending on the OCR engine in use (Abbyy, Adobe, OmniPage, ReadIris and Tesseract); and the manual transcriptions, T^M, are provided by the curators of the physical document collections in the Lee Library. The error measure $Err()$ is the word error rate. Given these inputs, the corrected text, ${}^C e_i^{OCR}$, is formulated as

$${}^C e_i^{OCR} = C\left(OCR\left(Scan\left(d_i^N, \theta_S^M, \phi\right), \theta_{OCR}^M\right), \theta_C^*\right). \tag{2.13}$$

This is a simplified representation of the research in this dissertation, in which the focus is to propose error correction processes, $C()$ and learn the optimal parameters and inputs, θ_C^* to reduce the WER of the resulting text output of the correction processes.

2.2.2 Post-OCR Error Correction Discussion

Each OCR recognizer has its strengths and weaknesses [2], as shown in Figure 1.2 and Table 1.2 in Section 1.3. Klein and Kobel [49] as well as Cecotti and Belaïd [14] note that the differences between OCR outputs can be used to advantage. Post-OCR error correction systems accept OCR output as evidence about the original document. Most directly related to this work are post-OCR error correction methods in which no attempt is made to directly modify the algorithms that recognize the content of the given digital images themselves.

The approach taken in this research is to treat the OCR engines as black boxes. An early work using the black box approach is Handley and Hickey [35] in which the authors character align the output of three OCR engines using dynamic programming and select the string that minimizes the edit distance between the three outputs. Effectively this is a form of voting at the character level since if two or more strings have the same character, that character will be selected. (More on voting as an ensemble method will be covered in Section 2.6.7.) Treating OCR engines as black boxes in an error correction process is also seen in Esakov, Lopresti, and Sandberg [25] on evaluating recognition errors of OCR systems. Kolak and Resnik [51] specifically applied their algorithms to OCR black box systems for post-OCR error correction using Bayesian methods to estimate $P(truth|observations)$. And Kolak, Byrne, and Resnik [52] applied their algorithms to OCR black box systems for post-OCR error correction for using a noisy channel model. More recently Liobet et al. [57] post process OCR using weighted finite-state transducers.

Each of these treatments of post-OCR error correction aligns and processes at the character level, making the assumption that effectively voting at a character level will improve the overall error correction. A limitation of working at the character level for hypothesis selection is that if more than one OCR engine makes the same error, the error will be selected, whereas this research is able to select individual hypotheses as tokens which exhibit a higher likelihood of being correct in the face of multiple similar errors at a character level. (See Chapters 5, 6, and 7).

A natural question in response to using the OCR engine as a black box is why the OCR engine itself is not directly targeted for improvement? There are two responses to this from the perspective of the researcher. First, with the exception of the open source tools, Tesseract [96] and OCRopus [11], the character and word recognition algorithms are propriety and not available to researchers to be modified. Second, the value demonstrated in this work is in the variations of the recognizers. Each team of developers approaches the problem of character and word recognition differently, resulting in diversity, which is critical to the performance of ensemble methods. (See Section 2.5.) Chapter 5 also shows that even high error rate contributions to an ensemble, if they show diversity from the other ensemble inputs, contribute to reducing the overall WER of the ensemble output. This outcome contradicts Caruana et al. [13] and Cer et al. [15], who conclude that only the best contributions (the lowest WER sequences in our context) should be used in an ensemble.

2.3 Alignment

Although commonly required with ensemble methods, much of the literature on OCR error correction using multiple inputs does not discuss the specific method used to align and compare multiple input sources. Alignment in some form, although required in ensemble methods, is not itself an ensemble method since there is no single selection from the aligned sequences. Sequence alignment, of which text alignment is a special case, is typically performed using algorithms that are either exact, resulting in an optimal alignment, or heuristic, resulting in a good, although not guaranteed to be optimal alignment. The multiple sequence optimal alignment problem has been shown to be NP-Hard in a paper on multiple sequence alignment complexity by Wang and Jiang [109]. Elias [23] discusses how a simple edit distance metric, which may be appropriate to text operations, is not directly applicable to biological alignment problems, which means that much of the alignment work in bioinformatics requires some adaption for use in the text sequence case.

Sections 2.3.1 through 2.3.4 provide background into text alignment. The formulation of alignment in the context of Section 2.1 is found in Section 2.3.5.

2.3.1 Tokenization and Alignment Order

Implicit in multiple inputs and ensemble methods is the requirement to make comparisons between alternative hypotheses. Virtually all papers implementing ensemble methods do not include the details of the alignment methods used, which then must be inferred. The important issue is whether the tokenization occurs before or after the alignment, specifically, whether the alignment process considers possible alignments in which the ground truth token may have been split during the recognition processes, as can occur in OCR and handwriting recognition.

Consider the following example from the Eisenhower Communiqués Number 204a outlined in Table 2.1. The two approaches to the order of alignment and tokenization show significant

Table 2.1: A comparison of the alignment outcomes of tokenizing before and after alignment using Communiqué Number 204a from the Eisenhower Communiqués corpus. Aligned tokens are separated by the vertical bar. The dash character, "-" represents a gap or $INDEL$ needed for the alignment.

<table>
<tr><th colspan="2">OCR Output Prior to Alignment</th></tr>
<tr><td>OCR Engine</td><td>OCR Output</td></tr>
<tr><td>Abbyy</td><td>ARTILLERY. FIRE WAS</td></tr>
<tr><td>OmniPage</td><td>ARTILLERYFIRE NLS</td></tr>
<tr><td>Tesseract</td><td>ARTILLERY FI RE WAS</td></tr>
<tr><td>Transcript</td><td>ARTILLERY FIRE WAS</td></tr>
<tr><th colspan="2">Aligned by Character, Then Tokenized by White Space Across the Alignment</th></tr>
<tr><td>Abbyy</td><td>ARTILLERY. FI-RE | WAS</td></tr>
<tr><td>OmniPage</td><td>ARTILLERY--FI-RE | NLS</td></tr>
<tr><td>Tesseract</td><td>ARTILLERY- FI RE | WAS</td></tr>
<tr><td>Transcript</td><td>ARTILLERY- FI-RE | WAS</td></tr>
<tr><th colspan="2">Each Sequence Tokenized by White Space, Then Aligned by Token</th></tr>
<tr><td>Abbyy</td><td>ARTILLERY. | FIRE | - | WAS</td></tr>
<tr><td>OmniPage</td><td>ARTILLERYFIRE | - | - | NLS</td></tr>
<tr><td>Tesseract</td><td>ARTILLERY | FI | RE | WAS</td></tr>
<tr><td>Transcript</td><td>ARTILLERY | FIRE | - | WAS</td></tr>
</table>

differences. In the first case, "Aligned by Character, Then Tokenized by White Space Across the Alignment", the character alignment brings together character sequences from the three OCR inputs that are closely related. Tokenizing by agreement on white space results in all of the instances of "ARTILLERY FIRE" being brought together within the same column of hypotheses, which means that the down stream error correction process can see all of the possible work hypotheses gathered together. This will permit later steps to consider the possibility that the ground truth token might have been incorrectly segmented in OCR sequences. In the second case, "Each Sequence Tokenized by White Space, Then Aligned by Token", the obvious problem is that the word hypotheses for "ARTILLERY FIRE" have been split across multiple columns of hypotheses. This requires the down stream correction process to consider not only the current column of hypotheses but those on either side, which increases the complexity of the correction process. This can be summarized as: $Align(Tokenize()) \neq Tokenize(Align())$

A commonly used alignment and voting tool [6, 9, 33] is the Recognizer Output Voting Error Reduction (ROVER) [28] which assumes that the sequences it receives have been tokenized which is the $Align(Tokenize())$ method. Developed for speech recognition, ROVER aligns the sequence tokens, separated by white space, and can be configured to return a selected hypothesis based on voting. (See Section 2.6.7.) This is appropriate for tasks in which the recognizer selects the output from a set of classes representing words, such as speech recognition, machine translation, and handwriting recognition. In these tasks the output is from a known set, words that that are in the known vocabulary. In the OCR task, OCR engines recognize characters and may select from an n-best list of hypotheses based on dictionaries or language models. However, it is also common in OCR output to find unknown tokens which are due to artifacts not related to text on the digitized document image being mis-recognized as text.

The method used in this research is $Tokenize(Align())$ in which the sequences are first aligned by character and then tokenized based on common white space across the alignments. More details on this research's approach to alignment may be found in Chapters 3 and 5. The approach of aligning on characters first and then tokenizing on white space provides a broader perspective on the possible hypotheses for a word hypotheses. The implication of aligning first on tokens is that you trust the placement of white space, which if incorrect must be resolved in a later step in the processing. The argument is that aligning characters, then tokenizing resolves this issue by consolidating words that may have been split incorrectly by the OCR engine. For classifiers using a known set of classes consisting of words, such as voice recognition and machine translation, it is appropriate to assume that the white space between tokens is appropriately placed; however, in OCR it is common for words to be merged when a space is not recognized, or for words to be split when a white space is inappropriately placed in the middle of a word by the OCR engine.

2.3.2 Optimal Alignment

Ikeda and Imai [39] and Schroedl (2005) [93] represented alignment problems as minimum distance problems in a directed acyclic graph in which edge costs represent the cost of alignment decisions.

Notredame states “Computing exact MSAs [multiple sequence alignments] is computationally almost impossible, and in practice approximate algorithms (heuristics) are used to align sequences, by maximizing their similarity.” [77]

Chapter 3 presents a novel admissible heuristic to be used with the A* exact alignment algorithm. The size of the exact alignment problem is reduced by two to three orders of magnitude, which, although significant, is still insufficient to compute exact alignments in reasonable time for large documents or number of sequences. Based on unpublished work [60], the author of this dissertation determined that the difference between the exact alignment and a progressive alignment heuristic for digitized text was not sufficient to warrant the additional computational time to perform an exact alignment. Typically, the progressive alignment completes aligning three sequences of around 2000 characters in a matter of seconds. Even with this improvement, it was necessary for computational efficiency to adopt non-optimal alignment methods.

2.3.3 Heuristic Progressive Alignment

Progressive text alignment can be approached in two ways. One greedy approach, used by Boschetti et al. [10], identifies the two sequences (of the n sequences to be aligned) that have the lowest alignment cost to each other. These two sequences are aligned. From the $n - 2$ sequences remaining, the next to be aligned is the sequence with the lowest alignment cost with the first two sequences, where the alignment between the first two sequences can not be altered other than to insert a gap across the alignment to match the new (third) sequence. This process is repeated until all n sequences have been aligned in a progressive manner. The process described by Boschetti is the method used in Chapters 5 through 8.

Wemhoener et al. [110] use a progressive alignment in which first two sequences are aligned and a third is added to the first two. This is the method used in Chapter 5. From the alignment of the three OCR sequences, Wemhoener proceeds to vote character-wise. The weakness in Wemhoener’s method is that if no single character hypothesis is identified the algorithm skips over the character.

In Chapter 3 of this dissertation, when voting did not succeed, the method selected the hypothesis from the OCR engine with the lowest WER on the training set.

An alternative to Boschetti, et al., Spencer and Howe [97] construct a guide tree, similar to genetic alignment techniques, from the matrix of all pairwise distances of the sequences. The guide tree, which is calculated upfront, determines the order in which sequences are aligned. In this process it is possible that more than one group of sequences may be aligned before further alignment of groups occurs. Again, once an alignment of some number of sequences has been completed, the existing alignment can not be altered during the alignment of a new sequence other than to insert gaps. Further, Ikeda & Imai and Schroedl formulated the cost of DNA base (or character) level alignments in n-dimensions as the sum of the costs of 2-dimensional alignments.

2.3.4 Character Alignment Costs

Esakov et al. [25] addressed the notion that OCR recognition errors are not uniformly distributed and that some types of errors are more prevalent than others, for example misrecognizing an “r” for a “t” or a “c” for an “e”. Using digital text from which digitized images were synthetically degraded, Esakov et al. calculated the letter confusion matrix, demonstrating that the errors from OCR are not uniform. Additionally, they showed that the error rates and error types were dependent on the text’s font. Brill and Moore’s work [12] in noisy channel spelling correction suggests going beyond single character replacement ($1 \rightarrow 1$, $1 \rightarrow 0$, and $0 \rightarrow 1$) to include a confusion matrix trained on multiple character replacement. Their work focuses on the types of errors made by human writers, but could be adapted to the types of errors made by OCR engines. These papers make the assumption that the tokens presented to them either by the OCR engine or by the human writer are accurate, or in other words, they may expect errors in the characters within the token, but they trust the spaces between the tokens. Observation indicates that errors with spaces between tokens (insertions, deletions, and substitutions) need to be expected just as they are with the characters that make up the word tokens. The work in Chapter A attempts to utilize this observation.

2.3.5 Alignment Formulation

Extending the formulation for extraction and error correction in Section 2.1 to incorporate multiple inputs, let a set of extractions be defined as

$$\bar{e}_i = \left(e_i^1, e_i^2, \ldots, e_i^n\right) \tag{2.14}$$

where each e_i^j is the result of a different extractor, such as different OCR engines, or a variation on the input to the same extractor, such as multiple binarizations of the same digital document image, d_i^I. The individual components of $\bar{e}_i$ are expressed as

$$e_i^j = E_j\left(d_i^I, \theta_{E_j}^*\right). \tag{2.15}$$

In the case of OCR output, which is text strings, let e_i^j be rewritten as A_i for the purposes of describing the alignment process.

Consider n text strings, $A_j^1, \ldots, A_j^n$, outputs of n OCR engines for the same digital document image d_j^I, such that $A_j^i = OCR_i(d_j^I)$. Let ${}^k a_j^i$ be a token (either a character or a word token depending on the implementation) such that $A_j^i = \left({}^1 a_j^i, \ldots, {}^{l_j^i} a_j^i\right)$ where $||A_j^i|| = l_j^i$. Using the notation from Section 2.1, let T_j^M be the gold standard transcription for document d_j^I, and $T_j^M = \left({}^1 t_j, \ldots, {}^{l_j^M} t_j\right)$ be the transcription broken into tokens.

Referring to Chapters 3 and 5 for a more complete discussion of optimal and progressive heuristic alignment methods, let the token aligned sequences and the transcription for document d_j^I be

$$\left(\hat{A}_j^1, \ldots, \hat{A}_j^n, \hat{T}_j^M\right) = Align\left(A_j^1, \ldots, A_j^n, T_j^M\right) \tag{2.16}$$

where

$$\hat{A}_j^i = \left({}^1 \hat{a}_j^i, \ldots, {}^{\hat{l}_j} \hat{a}_j^i\right). \tag{2.17}$$

All aligned sequences are the same length, $||\hat{A}_j^i|| = l_j$ for all documents d_j^I, meaning that each sequence can include *INDEL*s. The same formulation applies to $\hat{T}_j^M$. Due to the insertion of *INDEL*s, for all ${}^k\hat{a}_j^i$ either ${}^k\hat{a}_j^i =^p a_j^i$ for some $p \leq k$ or ${}^k\hat{a}_j^i = \emptyset$ where we define the *INDEL* as $\emptyset$.

Using the notation of Equation 2.12 the training of the error correction process is formulated as

$$\theta_C^* = \arg\min_{\theta_C} Err_C \left(C \left(OCR_1 \left(D^N, {}^1\theta_{OCR}^M \right), \ldots, OCR_n \left(D^N, {}^n\theta_{OCR}^M \right), \theta_C \right) T^M \right) \tag{2.18}$$

where $OCR_i()$ represents one of n OCR inputs. The alignment formulation in Equations 2.16 and 2.17 are not explicitly found in Equation 2.18 but are a part of the error correction process $C()$.

As alignment is a part of the error correction process, the question is whether the order of alignment is significant to the outcome. For exact alignment algorithms, such as A*, the order of OCR engine inputs is irrelevant since all of the text sequences from the OCR engines are evaluated simultaneously. Exact alignment is used in Chapters 3 and 4. Heuristic alignment algorithms are not guaranteed to provide an optimal result, but allow the alignment of more text sequences than exact alignment. Progressive alignment, first discussed in Chapter 5, is sensitive to the order in which the text sequences are introduced to the alignment. The method described in Chapter 5 uses a training set to determine the order of alignment, choosing text sequences in the order of lowest to highest WER as determine on the training set. This provides a guide-tree-like order, permitting potentially high WER inputs to align with already seen tokens from lower WER sequences.

Since the lengths of all of the alignment sequences are equal we can define a lattice of hypotheses as $H_j = \left({}^1\mathbf{h}_j, \ldots, {}^N\mathbf{h}_j \right)$. Depending on the approach, each ${}^k\mathbf{h}_j$ may either consist of individual characters or tokens, where the tokens may either be individual white space divided tokens with a sequence, as shown in Table 2.1, heading “Each Sequence Tokenized by White Space, Then Aligned by Token”, or aligned characters separated into tokens by agreement on white space across the alignments as shown in the same table, heading “Aligned by Character, Then Tokenized by White Space Across the Alignment”.

2.3.6 Non-Alignment Approaches

In an approach that did not require the alignment of the multiple inputs, DeNero et al. [20] in a machine translation work search for common n-grams from multiple translations of the source text. Rather than attempting to align the machine translation inputs, which is problematic due to the possible differences in word order in translation, n-grams that occur more frequently are selected.

In post-OCR error correction, Volk et al. [107] assume that output from two OCR engines will be very similar and perform a character-wise step through of the two outputs. When there is disagreement between the two sequences they observe a window of forth characters, centered on the point of disagreement. Where there is agreement between the two sequences, they accept the character. Where there is disagreement they consider all possible combinations and using a unigram language model select the most likely sequence of characters.

2.4 Ensemble Methods

Ensemble methods, in which the combination of multiple weak classifiers create a stronger model than any of the individual components, are an important method used in this research. This research applies novel approaches to existing ensemble models to extract text with a lower WER than found in any of the multiple OCR inputs. Dietterich [21] observes that "ensembles are often much more accurate than the individual classifiers that make them up." The ensemble methods used are: voting in isolation (Chapter 3), a decision list (Chapter 4), and Conditional Random Fields (Chapters 5 through 8). Chapter 5 demonstrates that even combining strong and weak models leads to improvement in the OCR WER. Cecotti and Belaïd [14] note that combining multiple classifiers has been shown to outperform individual classifiers, but only when they complement each other. The complementarity of the multiple OCR engines (which are classifiers) is addressed by calculating, during the evaluation step, the lattice word error rate on the aligned lattice. This method provides a lower bound on the information available from the multiple OCR engines.

In a work on ensemble methods for combining classifiers Kittler et al. [48] formulate the problem in terms of pattern recognition, where an input pattern Z is to be assigned to one of m

classes from $\Omega = \{\omega_1, \ldots, \omega_m\}$. Given R classifiers, each has a unique measurement vector, $\mathbf{x}$, where $\mathbf{x}_i$ is the measurement vector of the i^{th} classifier. Deviating from Kittler et al.'s formulation for the sake of future usage, let $\phi_i(Z) = \mathbf{x}_i$ be a function which takes as input the pattern Z to be assigned a class, and outputs the measurement vector $\mathbf{x}_i$ for the i^{th} classifier. Each class from Ω is modeled by a probability function, $p_k(\mathbf{x}|\omega_k)$, which gives a probability that the given measure vector, $\mathbf{x}$, is associated with the class ω_k. In terms of the input pattern Z a clearer representation is

$$p_k\left(\phi_i(Z) \,|\, \omega_k\right) \tag{2.19}$$

showing the probability that the input pattern is associated with class ω_k.

Maximizing the *a priori* probability on the observed measurement vectors for Z, Kittler et al. formulate the assignment problem as

$$\begin{gathered} assign\ Z \to \omega_j \quad if \\ P(\omega_j|\mathbf{x}_1, \ldots, \mathbf{x}_R) = \max_k P(\omega_k|\mathbf{x}_1, \ldots, \mathbf{x}_R) \end{gathered} \tag{2.20}$$

maximizing the probability that ω_k is the class of the pattern given the combined observed feature measure vectors. Note that $P(\omega|\mathbf{x}_1, \ldots, \mathbf{x}_R)$ is a combination of all R recognizer probability functions from Equation 2.19. Proceeding, Kittler et al. use Bayes Rule to rewrite Equation 2.20 as

$$P(\omega_k|\mathbf{x}_1, \ldots, \mathbf{x}_R) = \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_R|\omega_k)\,P(\omega_k)}{p(\mathbf{x}_1, \ldots, \mathbf{x}_R)} \tag{2.21}$$

in which $p(\mathbf{x}_1, \ldots, \mathbf{x}_R|\omega_k)$ is the combined posterior probability that the measure vectors occur given the class. The denominator of Equation 2.21 is fixed given the feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_R$ permitting the consideration of the numerator alone

$$P(\omega_k|\mathbf{x}_1, \ldots, \mathbf{x}_R) \propto p(\mathbf{x}_1, \ldots, \mathbf{x}_R|\omega_k)\,P(\omega_k)\,. \tag{2.22}$$

Returning to Equations 2.20 and 2.22 the general formulation is

$$\begin{gathered} assign\ Z \rightarrow \omega_j \quad for \\ P\left(\omega_j|\mathbf{x}_1,\ldots,\mathbf{x}_R\right) \propto \max_k p\left(\mathbf{x}_1,\ldots,\mathbf{x}_R|\omega_k\right) P\left(\omega_k\right). \end{gathered} \tag{2.23}$$

Implicit in this formulation is that the ensemble consists of the recognizers, $p_k\left(\phi_i\left(Z\right)|\,\omega_k\right)$, a series of probability functions which which are not clearly delineated in Equation 2.23. An equivalent and more common formulation of Equation 2.23 is

$$\begin{gathered} assign\ Z \rightarrow \omega \quad for \\ \omega = \underset{\omega}{\operatorname{argmax}}\ p\left(\mathbf{x}_1,\ldots,\mathbf{x}_R|\omega\right) P\left(\omega\right). \end{gathered} \tag{2.24}$$

Kittler et al. proceed, with the assumption that the classifiers as indicated in Equation 2.19 are independent, to develop both product and summation based interpretations of Equation 2.23. Comparing the general formulation in Equation 2.18 to the above Equation 2.24, rather than finding the maximum probability of the class given the observations, the generalized equation for training the optimal parameters for the correction process $C()$ minimizes the error. In Kittler et al.'s formulation, Equation 2.24, training occurs in the form of the probability of the observations given the class. Kittler et al.'s work is a framework for ensemble methods in classification. More specifically, this research uses multiple features of the aligned hypotheses, described in Section 2.3 in Conditional Random Fields as a classifier which will be described in Section 2.6.2.

2.5 Diversity in Ensemble Methods

Diversity in both the errors and the accurate classifications in an ensemble of classifiers is recognized as a necessary and sufficient condition for the ensemble to be superior to any one of the individual classifiers, as long as the ensemble method is able to differentiate between correct and erroneous hypotheses from the ensemble. In a work on ensemble methods with neural networks, Hansen and Salamon [36] state that “randomness tends to differentiate the errors of the networks, so that the networks will be making errors on different subsets of the input space.” They continue, “as each network makes generalization errors on different subsets of the input space, . . . the collective decision produced by the ensemble is less likely to be in error than the decision made by any of the individual networks. The conclusion is that the ensemble can be far less fallible than any one network.” Dietterich [21] deduces that “a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse.” He continues:

> A learning algorithm can be viewed as searching a space $\mathcal{H}$ of hypotheses to identify the best hypothesis in the space. The statistical problem arises when the amount of training data available is too small compared to the size of the hypothesis space. Without sufficient data, the learning algorithm can find many different hypotheses in $\mathcal{H}$ that all give the same accuracy on the training data. By constructing an ensemble out of all of these accurate classifiers, the algorithm can “average” their votes and reduce the risk of choosing the wrong classifier.

Further, Cecotti and Belaïd [14] note that combining multiple classifiers has been shown to outperform individual classifiers, but only when they complement each other. Kittler et al. [48] recognized that “[a]n important issue in combining classifiers is that this is particularly useful if they are different.”

In a recent work on machine translation (MT), Cer, Manning, and Jurafsky [15] introduce the concept of “Positive Diversity Tuning,” a heuristic measure linearly combining measures of “Correctness” and “Similarity” between the existing inputs to the ensemble and candidates inputs as

$$PD_\alpha = \alpha \;\; Correctness\left(ref\,[\,]\,, sys_\theta\right) - \left(1 - \alpha\right) Similarity\left(other\_sys\,[\,]\,, sys_\theta\right) \tag{2.25}$$

where PD_α is the diversity measure, α is the parameter of linear combination and sys_θ are the machine translation system parameters which are varied to create the diverse MT systems. The

$Correctness()$ and $Similarity()$ functions are any which can score the correctness of translations against each other. In a formulation using BLEU[1] scores, Equation 2.25 becomes

$$PD_{\alpha} = \underset{\theta}{\arg\max} \quad \alpha \, BLEU \left(ref \, [\,] \, , sys_{\theta} \right) - (1 - \alpha) \, BLEU \left(other\_sys \, [\,] \, , sys_{\theta} \right) . \tag{2.26}$$

Cer, Manning, and Jurafsky start with an empty set of systems for the ensemble, a training set, and a translation system with some parameters θ. Looping through parameters of the translation system, a new system is trained to maximize Equation 2.26. This new trained system is added to the ensemble and the loop repeats. The authors note that in principle it is possible to have a large number of machine translation systems in the ensemble, but practically this presents computational challenges.[2] The authors publish up to five iterations of this process, indicating that there were five machine translators in the ensemble. For more details the reader is referred to [15].

In Gimpel et al. [32] the authors also explore creating a diverse set of translations in the machine translation task. They propose a measure of diversity of an *n*-gram as it appears in two translations

$$\Delta_n \left(\mathbf{y}, \mathbf{y}' \right) = \, - \sum_{i=1}^{|\mathbf{y}|-q} \; \sum_{j=1}^{|\mathbf{y}'|-q} \mathbf{1} \left(\mathbf{y}_{i:i+q} = \mathbf{y}'_{j:j+q} \right) \tag{2.27}$$

where $\mathbf{1}(\cdot) = 1$ if the enclosed statement is true, otherwise 0, $\mathbf{y}$ and $\mathbf{y}'$ are the two translations to be compared, and q is the length of the *n*-grams to be explored. The strength of this dissimilarity function is that it can be applied without knowledge of the ground truth of the translation. The limitation however is that although *n*-grams may occur in each of the translations, this does not mean that they are necessarily accurate. This would be particularly true in the post-OCR error correction task where more than one OCR engine may make the same errors. In particular, this behavior was found in two OCR engines which had common recognition components, consequently making similar errors.

[1] BLEU (Bilingual Evaluation Understudy) [81] is a method for evaluating the quality of machine translation.

[2] It is interesting to note that the same statement was made in papers twenty years ago on algorithms which were impractical then, but are well within the reach of even small capacity computers today. It is likely only a matter of time and increasing computational capacity until what appears in 2014 to be computationally challenging to be commonplace.

This dissertation addresses the contribution of diversity in three ways. First, Chapter 3 introduces the concept of a Lattice Word Error Rate (LWER), which is a lower bound on the possible document WER given the evidence from the ensemble inputs. This approach to finding a lower bound on an ensemble method appears to be unique to this work. The LWER is introduced and discussed in more detail in Chapter 3, but in brief, the character output from multiple classifiers, generally OCR engines, are aligned. From the alignment multiple hypotheses for a single token are extracted. The collection of each set, or column, of hypotheses is called the lattice. Given the aligned hypotheses of a single column (see Figures 3.10 and 6.4 for examples of the aligned lattice and hypothesis column) if any of the hypotheses are correct, the column of hypotheses is counted as having correctly identified the token. This is clearly an oracle calculation in that knowledge of the correct string is required.

The second evaluation of diversity is the size of the training set and how this effects the eventual results of the ensemble method as found in Chapter 6. One of the results found in this chapter (see Section 6.4.6) is that for the large Reuters-21578 training set used to train an Conditional Random Field (CRF), only a randomly selected 2% of the training set, selected five times with replacement to create five training sets, the results of which are averaged, created a model that had effectively the same performance as 100% of the training set. Given the feature vectors found in the training set, many are duplicates, contributing little to the overall CRF model. This observation can be seen as the inverse to the the previous paragraphs, in that once a level of diversity, available in the training data is achieved, additional training examples are not necessary.

The third evaluation of diversity in this work is found in the chapters on multiple binarizations of the same digital document image as inputs to an ensemble method. (See Chapters 7 and 8.) The first observation is that within the various binarizations of a digital document image, no single binarization is across the board best, measured by the resulting WER of the binarized document image. (See Figure 8.8.) Looking at each binarization level, Table 8.3 shows that even the binarized digitized document images with the highest WERs contribute to the overall improvement of the ensemble. A similar result is found in Chapter 5, Table 5.4 and Figure 5.6. This result is in contraction to Caruana et al. [13] and Cer et al. [15] that only “better” performing classifiers should be used in an ensemble.

2.5.1 Diversity Formulation

Although the concept that diversity within an ensemble is intuitively correct, this dissertation presents a formulation of how diversity can be viewed and measured within the context of post-OCR error correction. This can easily be adapted for other types of problems.

2.5.2 Lattice Word Error Rate in Measuring Diversity

Given the definitions for hypothesis alignment from Section 2.3.5, the Lattice Word Error Rate is calculated as

$$LWER(H_j) = \frac{\left|\left|\left\{\mathbf{h}_i^j \in H_j \mid \exists {}^k\hat{h}_i^j \, = \, {}^k\hat{t}_i\right\}\right|\right|}{||T_j^M||} \tag{2.28}$$

Equation 2.28 counts the number of aligned hypotheses, ${}^kh_j^i$, where none match the ground truth. The denominator is $||T_j^M||$ rather than $||\hat{T}_j^M||$ since we measure the error rate by the length of the transcript. Note that if $\hat{t}_i = \emptyset$, this identifies a point in the alignment where at least one OCR engine has inserted a token that was not aligned with the ground truth. This can occur when the OCR engine encounters noise in the digital document image, which can result in a WER and a LWER greater than 100%.

Writing the word error rate of a document in the format of Equation 2.28 we first create the alignment of a single text input, $\hat{A}_1$ which is aligned with the ground truth, $\hat{T}$. The resulting definition for WER is

$$WER\left(\hat{A}_j^i\right) = \frac{\left|\left|\left\{{}^k\hat{a}_j^i \in \hat{A}_j^i \mid {}^k\hat{a}_j^i \, \neq \, {}^k\hat{t}_j\right\}\right|\right|}{||T_j||} \tag{2.29}$$

Conceptually, where the LWER of a document is close to the WER of one of the inputs, diversity is low. In contrast where there is a significant difference between the LWER and the best WER of the ensemble inputs, there is significant diversity in the inputs as shown in Figure 2.1.

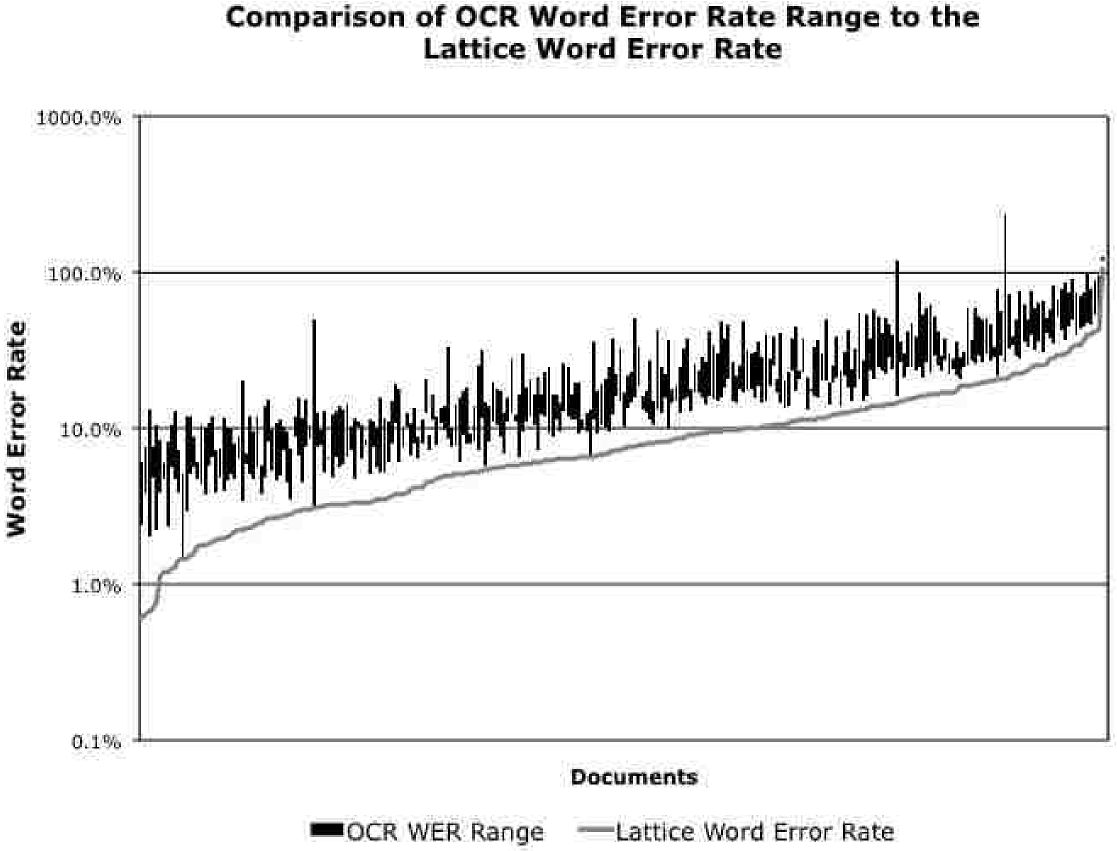


Figure 2.1: A comparison of the LWER to the range of WERs of the ensemble of inputs from multiple OCR engines on the Eisenhower Communiqués. The gap between the LWER line and the lowest point of the OCR WER Range bar indicates the potential contribution of diversity in the ensemble. Note that the OCR engine with the lowest WER is not necessarily always the same one. Refer to Figure 1.2.

Using the data from Chapter 3, Figures 3.15 and 3.17 shows the relative improvement that would be possible given the diversity of inputs from the ensemble of OCR inputs from the Eisenhower Communiqués. Chapter 3 and later chapters use the LWER as an indication of how close the ensemble method, when selecting a single hypothesis from the column of available hypotheses, is able to get to this theoretical minimum given the available evidence. This view may also be interpreted as the breath of diversity available in the ensemble inputs.

Clearly the case where the inputs to the ensemble show little diversity, the WERs of individual documents would be close together and noting Equation 2.28 there would be little difference between the WER and the LWER.

2.5.3 Diversity Discussion

Returning to Figure 2.1 there are two observations. First, that the $LWER \leq WER$ for all documents in the figure. Given the definition of the LWER this is clear in that if a sequence had a WER less than the LWER, the LWER would have used those correct hypotheses to reduce its own value. However, and this is the second observation, if $LWER = WER$ for a given text sequence in the ensemble this implies that there is no diversity in the remaining text sequences in the ensemble. Again considering the definition of the LWER, if for some sequence the WER equals the LWER, then the other sequences could be discarded without increasing the LWER, hence those sequences were not required and provided no diversity.

One last observation is that if $LWER = 0$ the ensemble of text sequences represents "perfect" diversity in that it is possible using the information in the inputs to extract the true text of the digital document image.

2.6 Types of Ensemble Methods

Ensemble methods incorporate a large number of algorithms with the common goal of merging multiple weak systems into a stronger system. The following sections briefly describe a number of ensemble methods.

2.6.1 Bagging and Boosting

Bagging and boosting are processes in which multiple, potentially "weak" classifiers are combined into a "strong" classifier, is effective for classification problems [13]. The method requires training to identify weights for each classifier, based on their relative strengths. An example of this method can be found in the information retrieval task in biomedical named entity recognition. Si, Kanungo, and

Huang [95] successfully applied boosting, and noted: “This approach provides us the opportunity to combine results from multiple systems that collectively use rich and diverse feature representations and also take advantage of utilizing multiple algorithms for achieving higher recognition accuracy.” Xiao et al. [112] apply bagging and boosting in machine translation by randomly selecting a subset of the training data to create multiple machine learners from the same statistical machine translation engine. Using the combination of the machine translation systems, a stronger ensemble is learned from the translation data, which is the boosting step. An alternative to selecting a subset of the training data to create multiple machine learners, is to use the full training data set but vary the parameters. Such an approach was used by Cer, Manning, and Jurafsky [15].

2.6.2 Conditional Random Fields

Using the derivation of Sutton and McCallum [100] for Conditional Random Fields (CRF), also known as logistical regression, the labeling problem is represented as

$$p\left(\mathbf{y}|\mathbf{x}\right) = \frac{1}{\mathbf{Z}\left(\mathbf{x}\right)} \prod_{t=1}^{T} exp\left\{\sum_{k=1}^{K} \theta_k f_k\left(y_t, y_{t-1}, \mathbf{x}_t\right)\right\} \tag{2.30}$$

where $\mathbf{Z}\left(\mathbf{x}\right)$ is the normalization function, $\theta_k \in \mathbb{R}$ is the parameter for the feature function $f_k()$. Note that this formulation is a chain function in that both the current label y_t and the previous label y_{t-1} are considered in the feature function, as well as the feature vector for $\mathbf{x}_t$ for the current location. Further, this formulation includes the probability for all of the labels $\mathbf{y}$ given all of the feature vectors for all locations in the sequence.

This research’s formulation of the CRF starting in Chapter 5 considers only the current location in the sequence and does not consider the value of the previous label. Chapter 6 shows that the value of the previous label, meaning the source of the selected hypothesis, such as the OCR engine or the value “NONE”, does not contribute to reducing the WER and is not used in this work. Simplifying Equation 2.30 to consider only the value of a single label, and eliminating the information concerning the previous label, the result is

$$p\left(y_t|\mathbf{x}_t\right) = \frac{1}{\mathbf{Z}\left(\mathbf{x}\right)} exp\left\{\sum_{k=1}^{K} \theta_k f_k\left(y_t, \mathbf{x}_t\right)\right\}. \tag{2.31}$$

The features and their values vary by chapter, to which the reader is referred. The CRF for this research was implemented using the Mallet Toolkit [72] which provides means for training and using the trained model to select a single hypothesis for each set of hypotheses.

2.6.3 Minimum Bayes Risk Decoding

Although Minimum Bayes Risk (MBR) is not exclusively an ensemble method, Zhou and Lopresti [117] propose a use of MBR using repeated sampling with the goal of improving classifier accuracy in OCR. Using Bayes decision rules (see Section 2.4) Zhou and Lopresti calculate the Bayes risk as the mis-classification rate for all classes. In brief, the MBR is the risk associated with mis-classifications that may occur due to a feature vector lying close the the decision boundary and, due to noise in the data, falling on the wrong side of that boundary. Zhou and Lopresti propose to improve the performance of the Bayes classifier by creating multiple inputs, specifically for their paper, of rescanned documents images for OCRing. Given the multiple inputs, the authors use voting to select the best hypothesis. Goel et al. [33] use MBR in speech recognition and propose a method through the use of n-best lists over which the speech recognition occurs.

2.6.4 N-best hypotheses

N-best lists can be created using a single recognizer and a single input where the list consists of alternative hypotheses with confidence measures. Venugopal et al. [105] in the machine translation task use n-best alignments rather than selecting a single best alignment of word translations. They note that selecting the n-best alignments is superior to selecting a single best alignment in that any error made in the a single alignment step is propagated to the remaining steps in the translation. By propagating multiple alignments it is possible to avoid propagating the error of a single alignment.

As mentioned in Section 2.6.3, Goel et al. use an n-best list of of possible speech recognition lattices, in which each lattice is an alternative segmentation of the phones of the speech input.

2.6.5 Reranking

Reranking takes multiple outputs of a model, such as an n-best list from a single input or the multiple outputs of multiple OCR engines, and reranks the n outputs based on features that may not have been found in the original, first step model. For instance, in a model such as in Equation 2.26, the training set may be sparse such that no instances of a combination of features occurs in the training set. In this case the probability of assigning a class to this unseen combination of features would be zero, or slightly more than zero in the case of smoothing. The reranking system is trained on the new features and makes decisions of which output from the model in the first step. In example of reranking for parsing, Collins and Koo [17] describe their use of reranking in the parser problem:

> The base parser produces a set of candidate parses for each input sentence, with associated probabilities that define an initial ranking of these parses. A second model then attempts to improve upon this initial ranking, using additional features of the tree as evidence. The strength of our approach is that it allows a tree to be represented as an arbitrary set of features, without concerns about how these features interact or overlap and without the need to define a derivation or a generative model which takes these features into account.

In another example of reranking of n-best lists, Charniak and Johnson [16] state:

> The reranker attempts to select the best parse for a sentence from the 50-best list of possible parses for the sentence. Because the reranker only has to consider a relatively small number of parses per sentences, it is not necessary to use dynamic programming, which permits the features to be essentially arbitrary functions of the parse trees. While our reranker does not achieve anything like the oracle f-score, the parses it selects do have an f-score of 91.0, which is considerably better than the maximum probability parses of the n-best parser.

In this research the output of multiple recognizers, either multiple OCR engines on the same digitized document image or multiple threshold binarizations of the same digitized document image recognized by the same OCR engine, are reranked using a Conditional Random Field model as implemented in the Mallet toolkit [72].

2.6.6 System Combination

System combination takes independently developed systems as input to the ensemble. The value in this approach is that the independence of the systems provides complementary error patterns. Rosti et al. [90] note: "[T]hree approaches [to machine translation, phrasal, heirarchical and syntax-based,] yield similar translation accuracy despite using fairly different levels of linguistic knowledge. The availability of such a variety of systems has led to a growing interest toward finding better translations by combining outputs from multiple systems."

The methods in this dissertation can be considered system combination when using multiple OCR engines, since each OCR engine was developed individually[3], as with Rosti, and Macherey and Och [69] they may be considered to be separate, independent systems.

2.6.7 Voting

Voting among multiple hypotheses is a common ensemble method used effectively to reduce the error rate. Using the notation from Equation 2.28, voting selects the class $\hat{h}^i$ from hypothesis vector $\mathbf{h}_i$ such that

$$\hat{h}^i = \operatorname*{argmax}_{h^i_j \in \mathbf{h}^i} \sum_{k=1, k \neq j}^{n} \mathbf{1}\left(h^i_j == h^i_k\right). \tag{2.32}$$

Lopresti and Zhou [59] use voting among multiple sequences generated by the same OCR engine from different scans of the same document. Their experimental results were between 20%

[3] It was discovered that the Adobe Acrobat Pro and the ReadIris commercial products shared a common base, which explained the correlation in the types of errors made by those two engines. Overall the Adobe Acrobat Pro embedded OCR engine had a lower WER, therefore it was retained and the ReadIris product removed from the OCR engines used in this research in later papers.

and 50% improvement. The results in Chapter 4 show a mean improvement of 50% using voting and feature selection with multiple OCR engines.

Although voting has strengths, the authors of ROVER [28] note the that "[i]n some cases, only one of the systems has the correct hypothesis, and its hypothesis is out-voted by more errorful systems." Another case in which voting can fail is when there is no consensus. Some papers [110] do not address this; however, in Chapter 3, in which voting is the primary decision tool, this dissertation addresses the case of no consensus by selecting the OCR engine with the lowest WER on the training set.

Klein and Kobel [49] match the output of multiple OCR engines of the same digital image. Their process first matches words from multiple OCR outputs, not by aligning the output but considering tokens from the sequences in order. They do not directly address the problem of tokens that are merged words (the result of not recognizing a space) or of split words (a space incorrectly inserted into the sequence). This research addresses this problem through aligning characters of the sequences directly, identifing insertions and deletions (INDEL) of spaces prior to the creation of aligned word token hypotheses. It appears that Klein and Kopel are using fairly clean documents in which INDELs are not common, which is in contrast to the collections used in this research. (See Section 2.8.) Treating word tokens as given, ignoring possible word segmentation problems, does not recognize a source of error in the OCR sequences. However, considering all possible combinations of spaces and word combinations, is combinatorially complex, as noted by Kukich [53].

Kittler et al. [48] indicate that "if only labels are available a majority vote ... is used." In contrast to this statement, this dissertation in fact selects between the labels of the multiple inputs, but attaches features to each label, making voting only a single facet of the space available for selecting the best token hypothesis. Recall from Section 2.2.2 the paper by Handley and Hickey (1990) [35] in which the authors minimize the edit distance between three OCR sequences. Effectively this is a form of voting in that the edit distance between matching characters is of course zero.

Table 2.2: Eisenhower Communiqué baseline corpus mean error rates compared to Lattice Error Rate and the results of Chapter 5 [64]

Abbyy	OmniPage	Adobe	ReadIris	Tesseract	LWER	Ensemble WER
18.24%	30.02%	51.78%	54.64%	67.78%	9.27%	13.76%

Voting between multiple OCR outputs as described above has been shown to be a very strong tool in correcting errors; however, it is limited to only discovering words that have been correctly recognized by at least one of the OCR engines. Table 2.2 compares voting by itself to machine learning and feature engineering techniques from Chapter 5. This research shows that in some cases, the expectation of achieving a reduced WER is not fulfilled despite using methods that have been successful in other cases. Exploring the causes of success or failure of various ensemble methods in the OCR error correction task is discussed in Chapter 8.

2.7 Applications of Ensemble Methods

Ensembles methods, such as those mentioned in Section 2.6, are used in a variety of applications. One of the conclusions of this work is that if reasonable multiple inputs to an ensemble method are available, the resulting outcome of the ensemble method will be superior to individual methods. For example, observe that no OCR engine, even the state-of-the-art engine, is superior to an ensemble of OCR outputs, which may include the state-of-the-art engine combined with OCR engines with much higher WERs. Another example is that multiple, threshold binarizations are superior to single adaptive binarizations when using the resulting WER of the text sequence of the digital document image.

The following sections discuss applications of ensemble methods.

2.7.1 Classifiers

Bertolami and Bunke [9] note that "classifier ensemble methods are used in may pattern recognition problems to improve the classification accuracy." They use a classifier ensemble for off-line handwriting recognition, in which the classes are words to be recognized. Bertolami and Bunke recognize that this creates a large number of classes. From the multiple word sequences generated by the ensemble of classifiers, the author use ROVER [28] to align and vote between the alternatives. Note that since the handwriting recognizers will only output word classes known to the classifier, the alignment will align on the given tokens and subsequently vote.

Beginning with Chapter 5 the process of post-OCR error correction is treated as a classification problem, where the classes are the sources of the OCR text (either an OCR engine or an image binarization threshold level) and the class “NONE”. This allows us to create a rich set of features, related to the label, even across labels, and to include a “NONE” label indicating that none of the inputs should be selected and all the hypotheses for this set of hypotheses should be rejected. This is particularly important for eliminating OCR noisy output, in which the OCR engine attempts to recognize image noise as characters.

OCR engines are also an example of a classifier. Cecotti and Belaïd [14] create an ensemble of OCR output focusing on the character recognition problem, considering each character individually without the benefit of the context. They also do not address the possibility of space mis-recognition, either recognizing a space where this is none, or not recognizing a space that exists. Further their work does not include external knowledge sources and feature engineering. Dordevic and Mihajlov [22] use classifiers in an OCR system for Macedonian Cyrillic. The authors’ system aligns character output and uses weighted voting to order an n-best output with confidence ratings for each output. The voting weights are discovered on a training set, which informs the confidence ratings.

2.7.2 Handwriting Recognition

Handwriting recognition continues to be an active research problem. In an early work in handwriting recognition of digits using an ensemble of neural networks Hanson et al. [37] state: “Neural network ensembles were introduced . . . as a means for improving network training and performance. The consensus of a neural network ensemble may outperform individual networks[.] . . . Furthermore the consensus may be used for realization of fault tolerant neural network architectures.” Recently

Govindarajan [34] notes that “Hybrid models have been suggested to overcome the defects of using a single supervised learning method, such as radial basis function and support vector machine techniques. Hybrid models combine different methods to improve classification accuracy. . . . Ensemble [methods improve] classification performance by the combined use of two effects: reduction of errors due to bias and variance” Govindarajan uses bagging to create multiple classifiers from the same training dataset and classifier, noting that an alternative to creating multiple classifiers would be to use multiple recognizers with an independent implementation.

2.7.3 Machine Translation

In the field of machine translation Gimpel et al. [32] note that translations in n-best lists can be very similar, limiting the value in their combination. Zhao and He [116] use eight models, which are combined using voting. Each of the models in their research uses a different paradigm for machine translation, noting that “a combination of [features and models] produces ... better translation results.” In another example, DeNero et al. [20] also apply model combination to the machine translation task. The strength of their work is to combine machine translation (MT) models that may provide very different outputs. N-grams that appear in multiple MT outputs give weight to each model in which they occur, resulting in selecting a derivation with the greatest weight due to common n-grams. Regardless of how the n-grams are placed within the sequence of the translation, their existence is considered a vote in the model. Finally, Cer, Manning, and Jurafsky [15] systematically build an ensemble of machine translation systems, emphasizing the diversity as was discussed in Section 2.5.

2.7.4 Named-entity Recognition

The Named-entity Recognition (NER) problem is to segment and classify proper names such as people, places, organizations, and dates within a text. The difficulty in the NER problem is that there are likely named-entities within any given text that are not found in a training set so there are instances where a named-entity must be identified by context. Extracting named-entities from noisy historical documents, Packer et al. [80] note: “We also conclude that combining basic methods can produce higher quality NER. Each of the three ensembles maximizes a different metric. The Majority Ensemble achieves the highest F-measure over the entire corpus, compared to any of the base extractors and to the other ensembles. The Intersection Ensemble achieves the highest precision and the Union Ensemble achieves the highest recall.”

2.7.5 Post-correcting Using Multiple OCR Outputs From the Same Engine

The only step in the OCR process that is truly noisy (in the sense that there is randomness in the output) is the image digitization step. Given the digital image, the OCR outputs from multiple runs with the same engine and the same input parameters, are identical; however small variations in the digital image can result in different OCR outputs. This observation has motivated approaches to extracting more than one OCR output from the same image. Lopresti and Zhou [59] scan a single physical document three times, slightly adjusting the document each time. These three digital images of the same document are recognized by the same OCR engine. Their system then aligns the character output and creates a consensus sequence through voting, which corrects between 20% and 50% of the OCR errors. They argue that using multiple OCR engines dilutes the effect of the best OCR engine; however, our research [64] in Chapter 5, shows that even OCR outputs with a comparatively high WER can contribute to an overall reduction in both lattice word error (LWER, an oracle calculation of the lowest possible WER given the information) and the observed WER. Another limitation of Lopresti and Zhou’s work is that voting is only done at the character level. Our methods, previously published [61, 62] and found in Chapters 3 and 4, demonstrate the usefulness of word level voting.

Due to the potential for OCR engines to interpret noise in the digital document image as characters, voting alone can not always identify hypotheses that should be rejected. This dissertation deals with this problem by allowing a “NONE” classification incorporated into machine learner, used first in Chapter 5.

2.7.6 Post-Correcting Using Multiple OCR Outputs from Different Engines

Abdulkader and Casey [2] implemented a system in which the output from a primary, trusted OCR engine is compared to potentially lower quality aligned outputs from multiple OCR engines. Disagreements at the character level are clustered where the primary OCR output and features of the token image are similar. This implies that the primary OCR engine is making consistent mistakes. A median example from each cluster is reviewed by a human to decide the correct output. This system merges the use of multiple OCR engines, voting, error clustering, and human correction.

Huanfeng and Doermann [68] provide a system in which segmented characters are clustered based on features and a human identifies the character associated with the cluster. The resulting character identification is then used to train a recognizer.

Again, although one OCR engine may be better overall, our previous work [64] found in Chapter 5 has demonstrated the advantages to leveraging the contributions of multiple OCR engines of varying quality. Further, one of the goals of this research is to address the very large corpus of pre-digital documents, there will be no human involvement in selection of output.

2.7.7 Automatic Speech Recognition

Automatic speech recognition (ASR) is a difficult task due to many factors outside of the control of the speech recognizer, such as ambient noise, variations in speech such as accent or dialect, and environment acoustics. Audhkhasi et al. [6] observe that "the fusion of hypotheses from multiple ASR systems is essential to achieve state-of-the-art word error rates." Working primarily with ROVER (see Section 2.3.1) the authors explore the effects of diversity in the multiple ASR systems on the ROVER output. In their conclusions they note: "Sentence hypotheses with high WER can lower the ROVER WER provided they are diverse and the fusion rule is able to take advantage of them." This is a similar result to that found in this dissertation in Chapter 5.

Goel et al. [33] approach the ensemble evaluation of ASR outputs by building the n-best lattices from a single ASR system. Each lattice has potentially different interpretations of phones from the speech input. Their approach is to cut each lattice at a point of agreement on the word

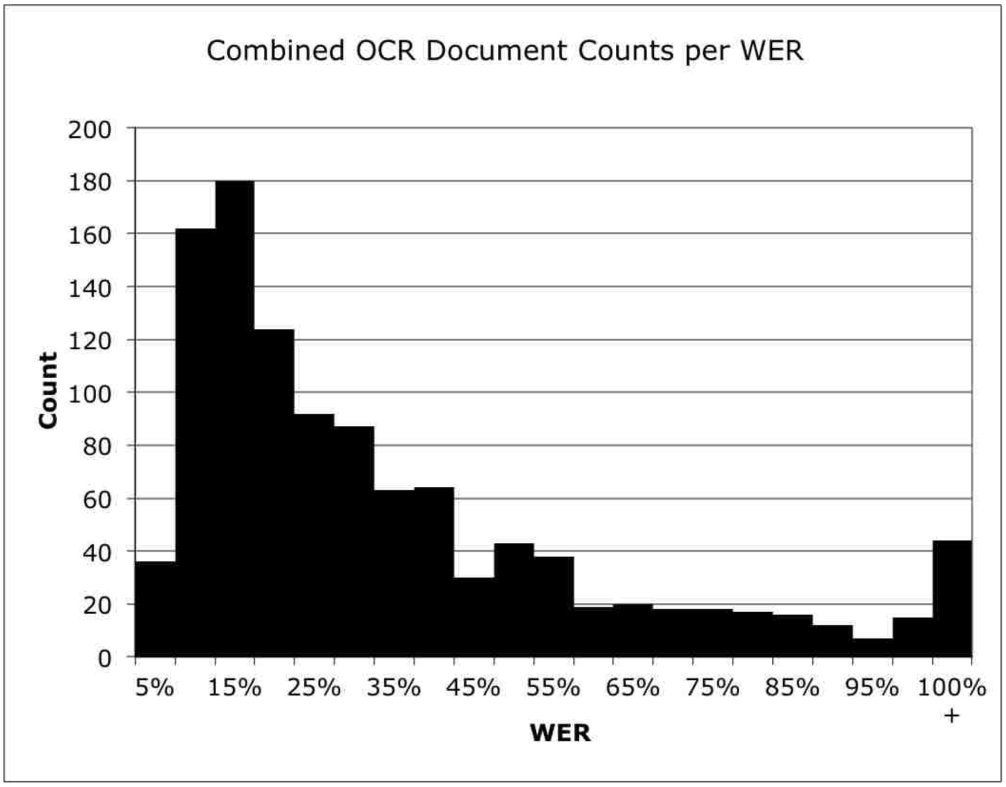


Figure 2.2: Document counts of uncorrected word error rates from the Eisenhower Communiqués.

output and evaluate the combination transitioning between between lattices at these points of agreement. This process results an improvement over a single best evaluation from the ASR system. the sequences from the OCR engines alone, without returning to the digital image.

2.8 Corpora

For the most part papers on post-OCR tend to use corpora created for the research situation. There are two common methods for creating a corpus for post-OCR error correction: a corpus created from known text and synthetically degraded either photographically or computationally, and corrected historical corpora from existing documents. A third corpus creation method is in finding multiple copies of the same work. Each of these will be covered below.

2.8.1 OCR Word Error Rate

One significant difference between the problems addressed in the related work cited here and the problem in this research is the WER of the uncorrected OCR. An early work on post-OCR error correcting by Jones, Story, and Ballard [42] dealt with a corpus having between 7% and 16% WER. In 2003, Kolak, Byrne, and Resnik [52] used a corpus with an 18.31% WER. And in 2009 Kluzner et al. [50], in an interesting work involving historical documents, used a corpus with a 17.5% WER. The corpora used in this work have a much higher WER as shown in Figure 2.2 with uncorrected OCR results of the Eisenhower Communiqués averaging 44.5% WER.

The word error rate of a document is calculated based on the number of token insertions, deletions, and substitutions divided by the number of tokens in the gold standard text. Let $T = \{t_1, \ldots, t_l\}$ be the tokens or words found in the gold standard text, which may be a manual transcription or manually corrected OCR.

$$WER = \frac{\#insertions + \#deletions + \#substitutions}{||T||}$$

Note that the WER may be greater than 100% since the number of insertions is not limited and may be caused by the OCR engine attempting to recognize noise as text and the possibility of inserting in error spaces in the middle of tokens. (See Section 2.5.2 for a more formal definition of the WER when aligned and compared to the gold standard transcription.)

2.8.2 Synthetic Corpora

Tong and Evans (1996) [103] create a corpus beginning with documents from the Ziff-Davis news wire. These documents were printed and photographically degraded by copying them on a photocopier on the “light” setting. Subsequently the degraded documents were scanned and OCRed. The resulting corpus WER was 22.9%. The original documents as used by Tong and Evans are available through TREC [104], however, the specific digital images and OCR used by the authors were not released. Kanugo and Haralick (1999) [46] likewise used photographic techniques to take a given digital text and create degraded images for input into an OCR engine.

In a related problem, the TREC-5 Confusion Track of 1999, created a corpus for the information retrieval task. The findings of this task are presented by Kantor and Voorhees [45], who describe the process of creating the corpus. The participants of the task were given three versions of the corpus: typeset files consisting of the digital text, which were considered to be the ground truth; an OCRed version of the clean typeset files with a corpus WER of 5%; and a downsampled version of the original image pages with a corpus WER of approximately 20%. These documents with some changes were later released by NIST as the Standard Reference Database 25 [76]. The TREC-5 images are stored in a proprietary format, not generally usable with current image manipulation tools.

Lopresti [58] created an OCR error correction database from the Reuters-21578 news corpus [55] selecting 661 articles. The digital text of the articles were printed and scanned at 300 dots per inch (DPI). Two more image sets were created by photocopying the printed documents using the "darkest" setting in two iterations. Similarly, two image sets were created by photocopying the printed documents using the "lightest" setting in two iterations. The OCR corpus work errors rates tended to be quite low, less than 2%. This dataset is not currently available.

For this research, in collaboration with Dr. Daniel Walker, we have created two synthetic corpora. These were introduced in Section 1.3.1. The 2001 Topic Annotated Enron Email Data Set [8] introduced in Walker, Lund, Ringger [108] is a synthetic dataset in which the text from the Enron Email Data Set is converted to a digital image and digitally degraded by the method proposed by Baird [7]. The resulting Enron dataset is bitonal with the original text files providing the gold standard for evaluation and training.

The Reuters-21578 [55] synthetic corpus was also created in collaboration with Dr. Daniel Walker and introduced for the first time in Lund, Kennard, Ringger [65]. Similar to the Enron dataset, the text is converted to images and degraded by the method proposed by Baird. In contrast to the Enron dataset, the images in the Reuters-21578 dataset are grayscale.

2.8.3 Corrected Historical Document Corpora

Although a historical document corpus with gold standard transcription is the best resource to capture the types of errors made by scanners, digital image binarization, and OCR engines, creating a historical document corpus requires significant manual effort and cost. Manually created gold standard transcriptions may either be created by direct manual transcription of the corpus, or post-OCR manual correction.

Reynaert (2008) [85] used three corpora, one of which is modern born-digital text of parliamentary proceedings and two historical newspaper datasets from the early twentieth century. Their process OCRs the historical datasets without correction. The focus of the author's work is to correct high-frequency words derived from the corpus. The processed corpora are not available; however, the original image and text files, either OCR or digital original, are available on the websites of the Acts of Parliament of The Netherlands[4] and the Database Digital Daily Newspapers[5].

Rangoni et al. (2009) [83] processed a subset of the Google 1000 Books dataset for post-OCR error correction. The focus of the Rangoni et al. paper is on selecting a threshold binarization value. They state that there are no good commonly used corpora for this task.

Volk et al. [107] created a corpus consisting of Alpine yearbooks from 1864 to 1995. Their strategies for reducing OCR errors included enlarging the OCR systems lexicons and two post-correction methods: merging the output of two OCR engines and auto-correction based on additional lexical resources. This corpus was never manually corrected.

2.8.4 Research Corpora

Of the four corpora (Eisenhower Communiqués [43], Nineteenth Century Mormon Articles Newspaper Index [29], 2001 Topic Annotated Enron Email Data Set [8, 108], and Reuters-21578 [55]), the Eisenhower Communiqués were manually transcribed and the Nineteenth Century Mormon Articles Index was OCRed using Abbyy FineReader, version 10, and manually corrected in two

[4] URL: http://www.statengeneraaldigitaal.nl/

[5] URL: http://kranten.kb.nl/

passes. Given the lack of standardized corpora for research into post-OCR error correction, the author intends to release all four corpora, either as a local download or as a part of a larger collection of corpora.

Note that across the remaining chapters, the number of documents in a given corpus may vary. This is due to either dividing up the corpus into test and training sets, or to more documents becoming available as the individual documents were transcribed and added to the corpus for this work. This occurred because the papers comprising Chapters 3 through 8 were written over the period of 2008 to 2013.

2.9 Binarization

Binarization, the process of converting a color or grayscale digital image file to a bitonal image, is an implied step in most OCR engines without necessarily being visible to or controllable by the user. Details of the process and related work in binarization, including adaptive binarization, can be found in Chapters 7 and 8. Most approaches to binarization attempt to find the single best binarization that is then processed to extract information from the digital image file. Often the quality of binarization is based on correctly identifying foreground (text) versus background (paper) pixels. For historical documents the question can become whether correctly identifying a pixel as foreground or background from the noisy digital document image, or from the presumed pristine original. A pixel may be correctly identified as foreground or background, given the state of the document, but still result in an OCR error as shown in Figure 1.2. Given that historical documents can be noisy even in the original, pristine document, correctly identifying foreground versus background pixels does not necessarily meet the goals of this research. This research's approach is more focused on the outcome in that since the goal of this research is to correct OCR, the quality of a binarization will be measured as the WER of the resulting OCR output from the digital image file (covered in Chapters 7 and 8).

A focus of binarization is in finding a single best binarization of a digital document image. Rangoni et al. (2009) [83] make use of multiple binarizations, but still attempt to find the single best binarization to be OCRed. Their approach is to create multiple binarizations for each digital document image. For the binarizations of a single document image they segment a portion of the document, for instance a single line of text, and OCR the digital image of that portion. Using a dictionary, they evaluate which of the binarizations of the digitized document image has the lowest error rate according to the dictionary. Selecting the binarization that has the lowest error rate, they OCR that binarization. The author of this dissertation believes that the methods covered in this research are superior in that rather than attempting to find a single best binarization, all of the available information in multiple binarizations is used.

2.10 Conclusion

The following chapters, as described in Section 1.4, are the published papers which constitute the research of this dissertation. Each chapter includes the reference to the publication venue of the work and has not been modified with the exception of addenda which include related works which the author has either become aware of since the publication or which were published subsequent to the paper's publication.

Chapter 3

Improving Optical Character Recognition through Efficient Multiple System Alignment

Abstract

Individual optical character recognition (OCR) engines vary in the types of errors they commit in recognizing text, particularly poor quality text. By aligning the output of multiple OCR engines and taking advantage of the differences between them, the error rate based on the aligned lattice of recognized words is significantly lower than the individual OCR word error rates. This lattice error rate constitutes a lower bound among aligned alternatives from the OCR output. Results from a collection of poor quality mid-twentieth century typewritten documents demonstrate an average reduction of 55.0% in the error rate of the lattice of alternatives and a realized word error rate (WER) reduction of 35.8% in a dictionary-based selection process. As an important precursor, an innovative admissible heuristic for the A* algorithm is developed, which results in a significant reduction in state space exploration to identify all optimal alignments of the OCR text output, a necessary step toward the construction of the word hypothesis lattice. On average 0.0079% of the state space is explored to identify all optimal alignments of the documents.

Published in *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2009)* [61].

3.1 Introduction

The digital era has set expectations that all documents are available electronically for searching and retrieval. Many legacy documents, available in print only, are difficult to impossible for optical character recognition (OCR) software to recognize. If these documents are going to be available for indexing, searching and other automated uses, some way must be found to create digital transcriptions.

Problems with OCR of poor quality printed text make the documents less accessible as the OCR output is less accurate to the point of being useless. Examples of this include typewritten documents where letters are incompletely formed or misaligned, copies of documents using poor duplicating technologies of carbon paper and mimeographing, and newsprint on deteriorating paper. An example of the first type is shown in Figure 3.1. This example is rendered by three OCR engines, as shown in Figure 3.2.

This paper presents results demonstrating the degree to which the output of multiple OCR engines may be used to improve the overall quality of the recognized text. A necessary component of the process is the alignment of the results from multiple OCR engines in order to identify alternative hypotheses for words in the form of a word hypothesis lattice. The alignment problem itself is the subject of much research, and this paper will present an admissible heuristic for use in the A* algorithm which substantially reduces the fraction of the state space that needs to be explored to find all optimal alignments of the texts.

For this paper we will use the Eisenhower Communiqués [43], a collection of 610 facsimiles found in the Harold B. Lee Library at Brigham Young University of original documents held in private hands. The original typewritten documents were created during World War II between the invasion of Normandy in June 1944 and the end of hostilities in Europe in May 1945. Although to a human reader they are legible (sometimes only just), experience has shown that OCR software cannot handle many of the problems in the documents, such as incomplete or malformed letters, typeovers, and artifacts of the duplication process.

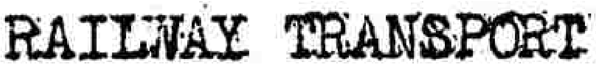
The remainder of the paper is organized as follows. Section 3.2 will provide background to the problem and related work. Section 3.3 discusses the algorithms used, in particular, an admissible heuristic used with the A* algorithm to align the output of the OCR engines. Section 3.4 discusses the reduction in search using this admissible heuristic and the reduction in the error rate found in the aligned word lattice over the individual word error rates of the OCR text output, as well as the word error rate (WER) of a dictionary-based selection among word alternatives within the lattice. Section 3.5 summarizes our conclusions and proposes future work.

3.2 Approach

3.2.1 Background

Optical character recognition is a classic noisy channel problem in which the true message is the original perfect text of the document and the recognized text of the digital image is the result of a transmission through a noisy channel in which errors are introduced. The quality of the output, as measured by the word error rate, can vary by the OCR software used. The objective of this research is to exploit these variations among OCR software systems in order to improve the quality of the final transcription. Figure 3.2 shows the OCR results of the three OCR engines used in this research. Clearly, an improvement can be made by considering the results of all three engines. One of the goals of this research is to quantify for this collection the extent to which the word error rate can be improved.

The Eisenhower Communiqués consist of 610 documents of varying quality. Initially the communiqués were typed on plain paper with upper and lower case letters and place names typed in all capital letters. An example from the communiqués is shown in Figure 3.3. This also shows examples of typeovers, strikeouts, and uneven alignment. Later documents were typed on a form in all capitals. Many of these documents have artifacts of the duplication process as can be seen in Figure 3.4. The entire collection was divided into two groups, a development set to be used in this phase and a test set for future work. The results reported here only use the documents found in the development set.



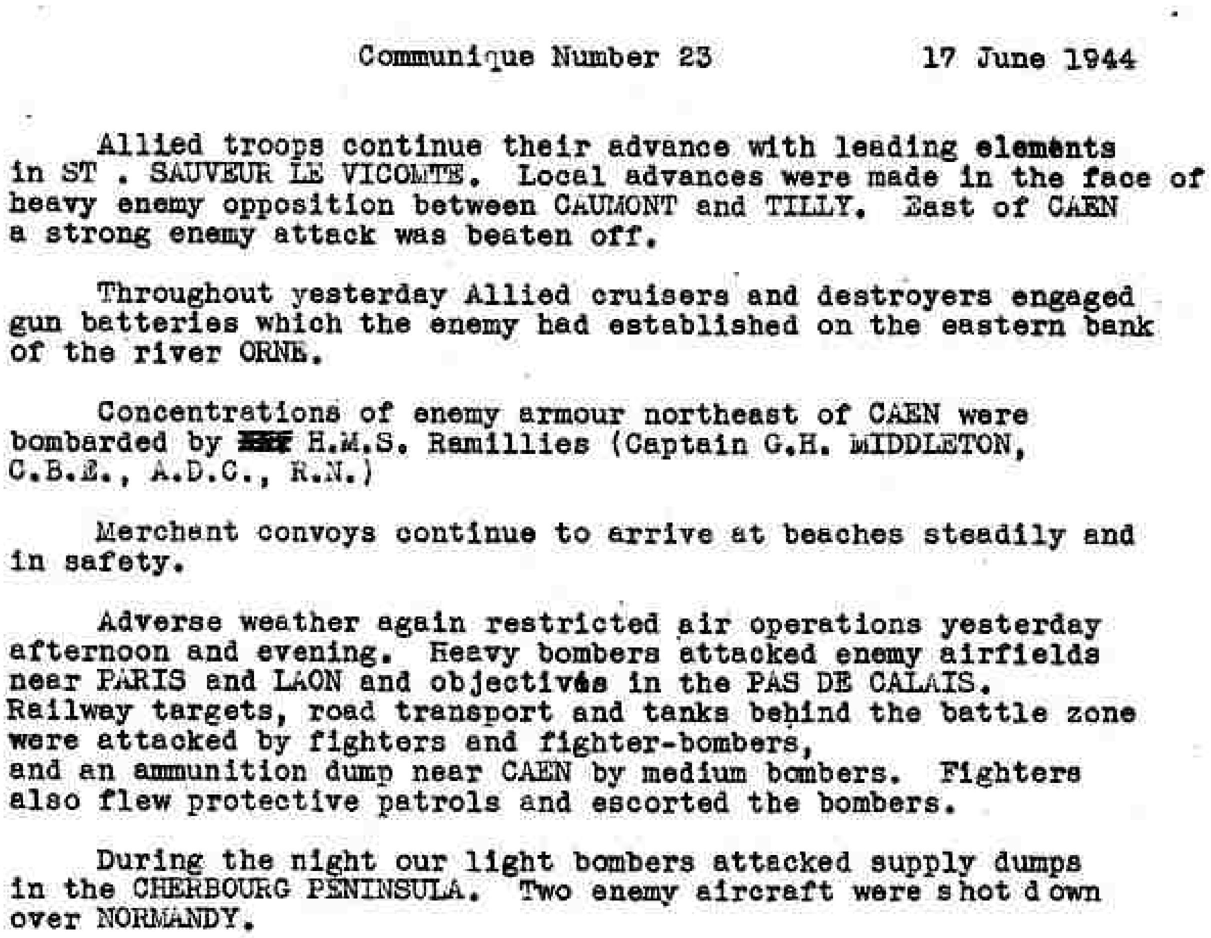
RAILWAY TRANSPORT

Figure 3.1: Poor quality text from Eisenhower Communiqué No. 237.

OCR Output

OCR A: RAILWAY mmmSBZ
OCR B: RAILWAY ANSP
OCR C: RAILWAI TRANSPORT

Figure 3.2: OCR Results from Figure 3.1. Underlining added for emphasis.



Communique Number 23 17 June 1944

Allied troops continue their advance with leading elements in ST . SAUVEUR LE VICOMTE. Local advances were made in the face of heavy enemy opposition between CAUMONT and TILLY. East of CAEN a strong enemy attack was beaten off.

Throughout yesterday Allied cruisers and destroyers engaged gun batteries which the enemy had established on the eastern bank of the river ORNE.

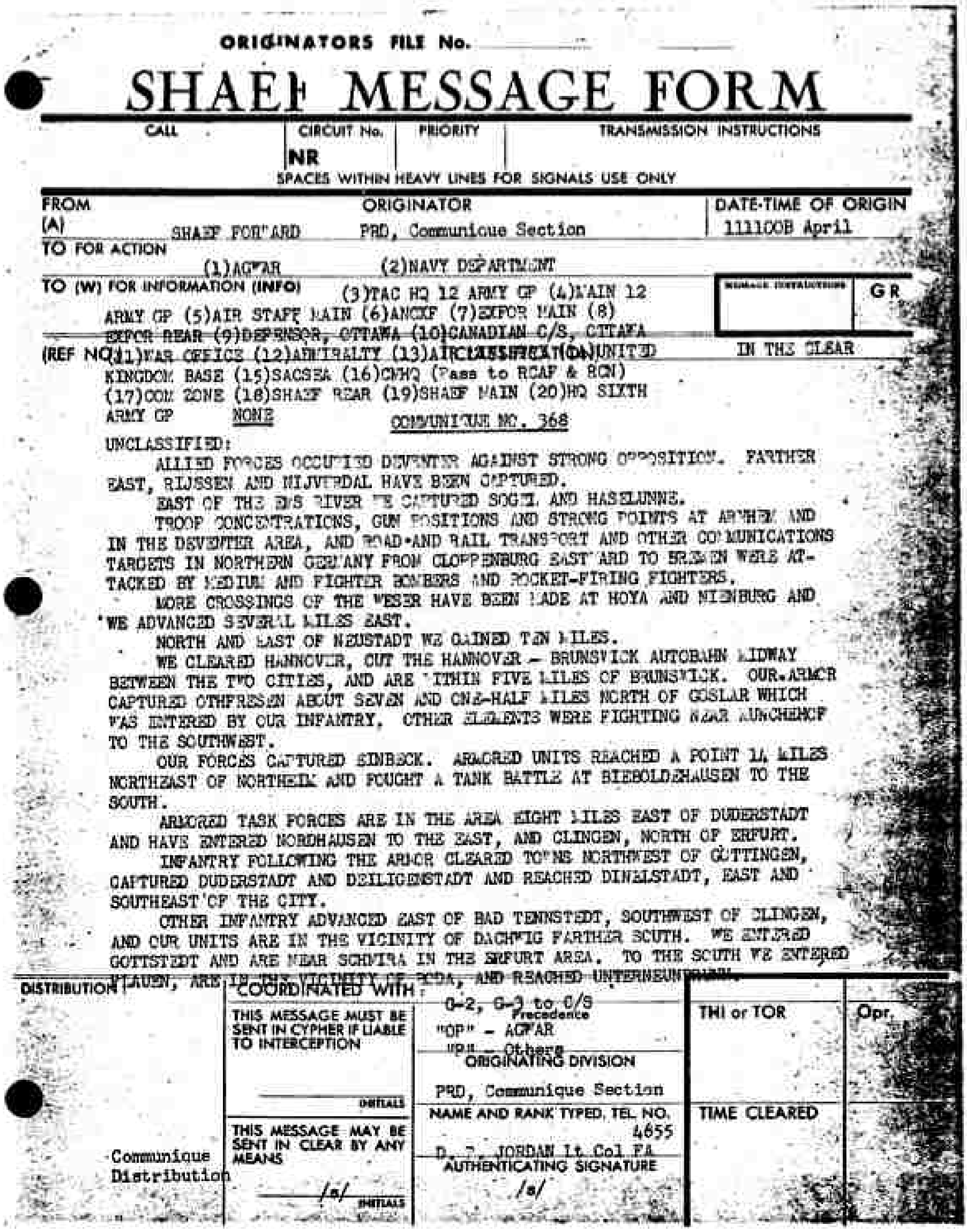
Concentrations of enemy armour northeast of CAEN were bombarded by [redacted] H.M.S. Ramillies (Captain G.H. MIDDLETON, C.B.E., A.D.C., R.N.)

Merchant convoys continue to arrive at beaches steadily and in safety.

Adverse weather again restricted air operations yesterday afternoon and evening. Heavy bombers attacked enemy airfields near PARIS and LAON and objectives in the PAS DE CALAIS. Railway targets, road transport and tanks behind the battle zone were attacked by fighters and fighter-bombers, and an ammunition dump near CAEN by medium bombers. Fighters also flew protective patrols and escorted the bombers.

During the night our light bombers attacked supply dumps in the CHERBOURG PENINSULA. Two enemy aircraft were s hot d own over NORMANDY.

Figure 3.3: Eisenhower Communiqué 23



ORIGINATORS FILE No. ______

SHAEF MESSAGE FORM

CALL	CIRCUIT No.	PRIORITY	TRANSMISSION INSTRUCTIONS
	NR		

SPACES WITHIN HEAVY LINES FOR SIGNALS USE ONLY

FROM (A) SHAEF FORWARD ORIGINATOR PRD, Communique Section DATE-TIME OF ORIGIN 111100B April

TO FOR ACTION (1)AGWAR (2)NAVY DEPARTMENT

TO (W) FOR INFORMATION (INFO) (3)TAC HQ 12 ARMY GP (4)MAIN 12 ARMY GP (5)AIR STAFF MAIN (6)ANCXF (7)EXFOR MAIN (8) EXFOR REAR (9)DEFENSOR, OTTAWA (10)CANADIAN C/S, OTTAWA (11)WAR OFFICE (12)ADMIRALTY (13)AIR MINISTRY (14)UNITED KINGDOM BASE (15)SACSEA (16)CMHQ (Pass to RCAF & RCN) (17)COM ZONE (18)SHAEF REAR (19)SHAEF MAIN (20)HQ SIXTH ARMY GP

MESSAGE INSTRUCTIONS GR

(REF NO) NONE CLASSIFICATION IN THE CLEAR

COMMUNIQUE NO. 368

UNCLASSIFIED:

ALLIED FORCES OCCUPIED DEVENTER AGAINST STRONG OPPOSITION. FARTHER EAST, RIJSSEN AND NIJVERDAL HAVE BEEN CAPTURED.

EAST OF THE EMS RIVER WE CAPTURED SOGEL AND HASELUNNE.

TROOP CONCENTRATIONS, GUN POSITIONS AND STRONG POINTS AT ARNHEM AND IN THE DEVENTER AREA, AND ROAD-AND RAIL TRANSPORT AND OTHER COMMUNICATIONS TARGETS IN NORTHERN GERMANY FROM CLOPPENBURG EASTWARD TO BREMEN WERE ATTACKED BY MEDIUM AND FIGHTER BOMBERS AND ROCKET-FIRING FIGHTERS.

MORE CROSSINGS OF THE WESER HAVE BEEN MADE AT HOYA AND NIENBURG AND WE ADVANCED SEVERAL MILES EAST.

NORTH AND EAST OF NEUSTADT WE GAINED TEN MILES.

WE CLEARED HANNOVER, CUT THE HANNOVER – BRUNSWICK AUTOBAHN MIDWAY BETWEEN THE TWO CITIES, AND ARE WITHIN FIVE MILES OF BRUNSWICK. OUR ARMOR CAPTURED OTHFRESEN ABOUT SEVEN AND ONE-HALF MILES NORTH OF GOSLAR WHICH WAS ENTERED BY OUR INFANTRY. OTHER ELEMENTS WERE FIGHTING NEAR MUNCHEHOF TO THE SOUTHWEST.

OUR FORCES CAPTURED EINBECK. ARMORED UNITS REACHED A POINT 14 MILES NORTHEAST OF NORTHEIM AND FOUGHT A TANK BATTLE AT BIEBOLDEHAUSEN TO THE SOUTH.

ARMORED TASK FORCES ARE IN THE AREA EIGHT MILES EAST OF DUDERSTADT AND HAVE ENTERED NORDHAUSEN TO THE EAST, AND CLINGEN, NORTH OF ERFURT.

INFANTRY FOLLOWING THE ARMOR CLEARED TOWNS NORTHWEST OF GOTTINGEN, CAPTURED DUDERSTADT AND DEILIGENSTADT AND REACHED DINELSTADT, EAST AND SOUTHEAST OF THE CITY.

OTHER INFANTRY ADVANCED EAST OF BAD TENNSTEDT, SOUTHWEST OF CLINGEN, AND OUR UNITS ARE IN THE VICINITY OF DACHWIG FARTHER SOUTH. WE ENTERED GOTTSTEDT AND ARE NEAR SCHWIRA IN THE ERFURT AREA. TO THE SOUTH WE ENTERED PLAUEN, ARE IN THE VICINITY OF RODA, AND REACHED UNTERNEUBRUNN.

DISTRIBUTION:		COORDINATED WITH:	THI or TOR	Opr.
Communique Distribution	THIS MESSAGE MUST BE SENT IN CYPHER IF LIABLE TO INTERCEPTION ______ INITIALS	G-2, G-3 to C/S Precedence "OP" - AGWAR "P" - Others ORIGINATING DIVISION PRD, Communique Section		
	THIS MESSAGE MAY BE SENT IN CLEAR BY ANY MEANS /s/ INITIALS	NAME AND RANK TYPED, TEL NO. 4655 D. F. JORDAN Lt Col FA AUTHENTICATING SIGNATURE /s/	TIME CLEARED	

Figure 3.4: Eisenhower Communiqué 368

Word Error Rate Calculation

Hypothesis text:	A	X	Y	Z	–	B
Reference text:	–	X	Y	T	J	B
Word Error Types:	I			S	D	

Figure 3.5: One insertion, one substitution, and one deletion relative to the reference text.

The word error rate used here to quantify the quality of the OCR output is the sum of the number of insertions (tokens not appearing in the transcription), deletions (tokens found in the transcription but missing from the OCR) and substitutions (a token substituted in the OCR for a word found in the transcription). See Figure 3.5. Note that a WER can be greater than 100%. The National Institute of Standards and Technology (NIST) [3] provides *sclite*, a toolkit to evaluate word error rates. This tool compares a reference transcription with a hypothesis text, calculating, among other things, the WER. In addition to flat hypothesis text, *sclite* accepts a hypothesis word lattice of alternatives, so that if any of the alternatives within the lattice matches the transcription word it is counted as a match even though alternatives may have been incorrect.

The reported lattice error rate provides a lower bound on the potential error rate achievable by choosing among the alternatives. It is the goal of the alignment portion of this paper to construct just such an aligned word lattice from the character-level OCR output.

3.2.2 Related Work

There are several research areas which are related to correcting OCR; for example, error correction in speech recognition is in many respects similar in that both problems are correcting errors from a decoded signal. Along these lines Ringger [86, 87] explored statistical methods in speech recognition post-processing to correct errors, while Fiscus [28] reduced the word error rate by voting among the output of multiple automated speech systems using a heuristic alignment of input, which is not guaranteed to be optimal. Mangu, Brill, and Stolcke [71] used a word lattice provided by a speech recognizer and converted it to one which is probabilistically consistent and on which they perform word error minimization, achieving a 1.4% WER reduction over previous work. Lastly, Brill and Moore (2000) [12] propose a string to string edit to correct spelling errors. Mardy and

Darwish [70] provide results for the OCR of Arabic text, using confusion matrices based on training data from the Arabic documents. The alignments use dynamic programming and the Levenshtein edit distance as the cost.

Xiu and Baird (2008) [113] examined improving text recognition at the character and word level with iconic and linguistic models that adapt the output of the recognizer using mutual-entropy-based adaptation. Their paper extends earlier work from single lines to full pages, improving the overall accuracy. The use of multiple recognition systems in named-entity recognition was explored by Si, Kanungo, and Huang in 2005 [95] where the results of three different recognition models were combined. Taking an approach different than that presented here, Ma and Doermann [68] propose a methodology for adaptive OCR with limited supervision; their goal being to improve the capabilities of OCR classifiers in noisy documents and where characters are overlapping or touching.

The multiple alignment problem has been shown to be NP-Hard first in the paper on multiple sequence alignment complexity by Wang and Jiang in 1994 [109], and later Elias in 2006 [23] presented results extending the NP-Hard classification of the multiple sequence alignment problem to several algorithms. Elias discusses further how a simple edit distance metric, which may be appropriate to text operations, is not directly applicable to biological alignment problems. Due to the number of sequences in genetic alignment problems the state-of-the-art involves finding heuristics that provide good results. Notredame states "Computing exact MSAs [multiple sequence alignments] is computationally almost impossible, and in practice approximate algorithms (heuristics) are used to align sequences, by maximizing their similarity." [77] Frequently, heuristic progressive alignments are performed, in which the optimal alignment of two sequences is found, to which additional sequences are added incrementally. Many of the current genetic alignment packages follow this approach with variations.

Ikeda and Imai (1994) [39] and Schroedl (2005) [93] represented alignment problems as minimum distance problems in a directed acyclic graph in which edge costs represent the cost of alignment decisions. Further, Ikeda & Imai and Schroedl formulated the cost of DNA base (or character) level alignments in n-dimensions as the sum of the costs of 2-dimensional alignments. This sum of pairwise costs is not the only way one might determine cost values on the edges in the n-dimensional graph. It is an open question how best to set the weights in such a graph, but we follow the prior work, while refining it and significantly reducing the number of nodes visited to discover all optimal alignments.

3.3 Methodology and Algorithm

The method presented in this paper consists of six steps: 1) Digitize the documents, 2) Submit each document to OCR engines, 3) Evaluate the word error rate of each individual output, 4) Collectively align the OCR outputs at the character level, and retrieve all optimal alignments, 5) Build the lattice of word hypotheses based on the optimal alignments, 6) Select words from the lattice of alternatives using a simple dictionary-based method, and 7) Evaluate the lattice error rate of the resulting word lattice and the word error rate of the selected word sequence.

3.3.1 Document Digitization and OCR

The documents were scanned at 400 dots per inch (DPI) with one bit black/white. The quality of the initial OCR output was very poor with many documents having a word error rate above 50%. Table 3.1 shows the resulting WER of each OCR engine. The "Maximum Average" entry takes the largest WER from the three OCR results for each document, and across all documents calculates an average of these maxima. Since the OCR engine that has the maximum WER can change for each document, this number is not the average of the numbers to the left of it. Likewise, the "Minimum Average" takes the minimum WER from the three OCR results for each document and calculates the average across all documents. Due to these high WERs, the collection curator chose to have the documents transcribed by hand, which provided a ground truth against which this work could measure a word error rate.

The OCR engines used in this work are Abbyy FineReader, OmniPage Pro, and Tesseract; however it is beyond the scope of this paper to make a formal comparison of these engines, hence

Table 3.1: Document Word Error Rates

Engine	A	B	C	Averages
Mean	16.4%	25.9%	26.3%	22.9%
Maximum	120.0%	125.0%	238.1%	122.1%
Minimum	2.3%	1.5%	2.0%	3.4%
Variance	0.0167	0.0381	0.0607	

they will not be explicitly identified with their output. Each of the scanned documents in the development set was processed by each of the OCR engines. Since the goal of this research was not to separate text from forms, the text of the communiqué was in some cases manually isolated from the form. (See Figure 3.4.)

3.3.2 Alignment

Alignment as a Minimum Distance Problem

A common way to formulate alignment problems is as a minimum distance problem using dynamic programming or the A* algorithm [39, 40, 93]. Figure 3.6 shows an alignment of the strings "ABCD" and "ACBD". Each edge in the network represents a decision in the alignment problem, and a path from the start to the goal represents a series of alignment decisions. The numbers on the edges represent a simple cost function in which a match between the two sequences has zero cost and all other edges have cost of one. The numeric labels along the top and left edges of the diagram are for reference only.

Starting at the upper left corner and proceeding to the goal node in the lower right corner, an alignment algorithm selects a path that minimizes the sum of the edges along the path. One possible optimal solution is shown in Figure 3.6 as bold arrows. All edges represent an alignment choice, whether to skip a letter in one of the sequences or to align two letters. All vertical and horizontal edges represent a choice to advance one sequence without advancing the other, creating a skip in the string not advanced. In Figure 3.6 the first vertical edge from $(1, 1)$ to $(1, 2)$ on the path would result from advancing sequence two without advancing sequence one. Diagonal edges represent

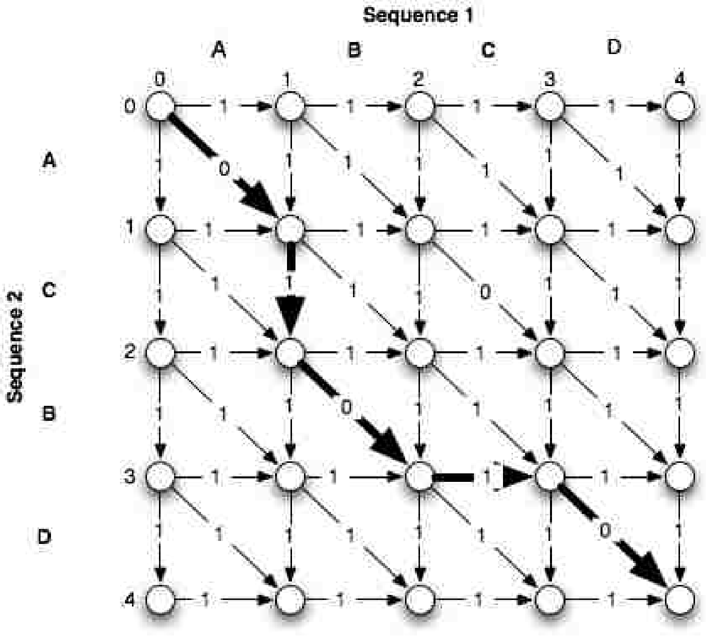


Figure 3.6: Text sequence alignment as a network problem. One optimal alignment is shown in bold.

an alignment (either match or substitution) between the two strings. Note that the algorithm may choose to align two dissimilar characters. The path shown in Figure 3.6 represents the alignment and edge interpretation found in Figure 3.7. The total cost of this path is 2. The method extends itself well into more than two dimensions. The cost of an edge is proposed by Imai and Ikeda [40] to be the pairwise sum of the alignment costs of all sequences. For example, suppose three sequences are being aligned and the edge under consideration is $(1, 1, 2) \rightarrow (1, 2, 3)$. The proposed cost for this edge would be the pairwise costs in two dimensions: $(1, 1, -) \rightarrow (1, 2, -)$, $(1, -, 2) \rightarrow (1, -, 3)$, and $(-, 1, 2) \rightarrow (-, 2, 3)$. Another way to think about this is to consider the edge being projected onto planes. (See Figure 3.8.) Every interior node will have $2^n - 1$ edges entering and the same number leaving, where n is the number of sequences being aligned.

To formalize this, consider n sequences, $S_1 \ldots S_n$ to be aligned. Let $S_i = (c^i_1, c^i_2, \ldots, c^i_{m_i})$, where c^i_j is the jth character of sequence i. S_i is an ordered sequence and $m_i = length(S_i)$. Let the cost function be such that it only depends on the single character from each sequence being aligned, ignoring all characters before and after. Suppose that this is implemented as a pairwise sum of the

Alignment from Figure 3.6

```
A-BCD
ACB-D
```

Edges	**Sequence 1**	**Sequence 2**
$(0,0) \rightarrow (1,1)$	align “A”	and “A”
$(1,1) \rightarrow (1,2)$	align no character	and “C”
$(1,2) \rightarrow (2,3)$	align “B”	and “B”
$(2,3) \rightarrow (3,3)$	align “C”	and no character
$(3,3) \rightarrow (3,4)$	align “D”	and “D”

Figure 3.7: Alignment Results from Figure 3.6

costs of aligning sequences. This can be represented as:

$$C_n(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_2(x_i, x_j) \tag{3.1}$$

where $C_n(\cdot)$ and $C_2(\cdot)$ are the n- and two-dimensional cost functions respectively, and $x_i \in \{INDEL\} \cup S_i$ in which $INDEL$ is a special character indicating a gap in the alignment of one sequence. In Figure 3.5 the $INDEL$ character is represented by the “-”. One of the requirements of the A* algorithm is that all edge costs are non-negative. Consequently, it is required that $C_2(x_i, x_j) \geq 0$ for all sequences.

Using the A* Algorithm

The A* algorithm [26] is a generalization of Dijkstra’s minimum cost algorithm in which an admissible [19] heuristic is used to estimate the cost from the current node under consideration to the goal node. The role of the heuristic is to reduce the number of nodes that need to be visited in order to find the minimum cost paths through the network, which Pearl [82] and Schroedl [93] note is related to how closely the heuristic estimates the true cost. When employed with an admissible heuristic, the A* algorithm reduces the number of nodes without sacrificing completeness or optimality.

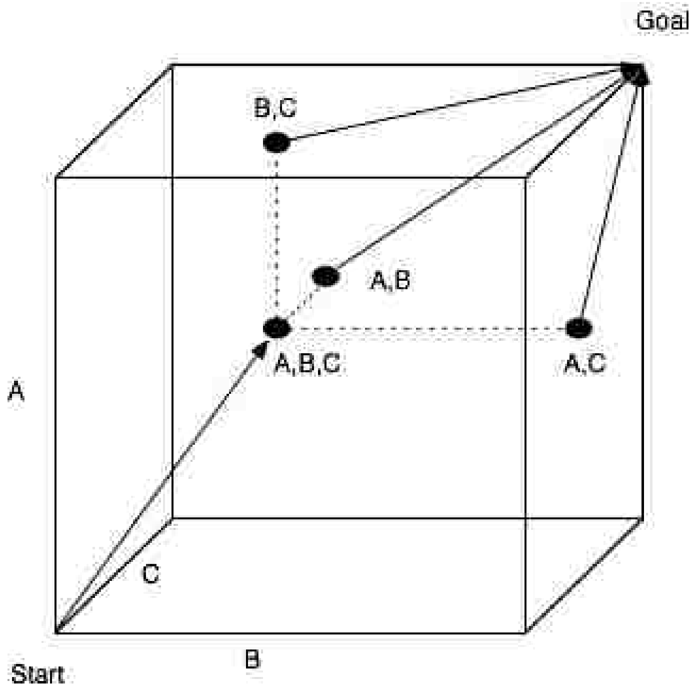


Figure 3.8: The 3-dimensional problem is reduced to three 2-dimensional problems in the heuristic for the A* algorithm.

Since one objective of this research (beyond the scope of the present paper) is to explore *all* optimal alignments of the text sequences for additional word hypotheses, the typical A* algorithm is modified slightly to return all optimal paths rather than stopping after a single minimum cost path is found.

The Reverse Dijkstra Heuristic

An alignment of three sequences of 2500 characters would consist of a network of 1.5×10^{10} nodes and 1.1×10^{11} edges to be evaluated. Any reduction in this is helpful to make the alignment feasible in shorter time with fewer resources. As would be expected, the heuristic and the edge cost function are tightly linked. Imai and Ikeda [40] suggest that a reasonable approach to the cost function, where the dimensionality of the problem is greater than two, is a sum of the pairwise costs of each sequence pair. This approach is used extensively in genetic alignment.

Considering the example above, two sequences of 2500 characters have 6,255,001 nodes and 18,755,000 edges in the network. This is considerably smaller than the 3-dimensional problem. To calculate the 3-dimensional heuristic cost from any one point in the 3-dimensional network, there will be three pairwise alignment problems, each of a size less than the full 2-dimensional problem since if we are in the interior of the network, there are fewer than the full set of characters remaining to be aligned. (See Figure 3.8.) Adapting our notation from the A* algorithm above, let $\underline{u} = (u_1, u_2, \ldots, u_n)$ and $\underline{v} = (v_1, v_2, \ldots, v_n)$ where $\underline{u}, \underline{v} \in U$, the set of nodes and $(\underline{u}, \underline{v}) \in E$ the set of edges in the network. Calculate the heuristic cost of an n-dimensional node, $h_n(\underline{v})$, as the pairwise sum of 2-dimensional costs, written as

$$h_n(\underline{v}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} h_2^{i,j}(v_i, v_j) \tag{3.2}$$

where $h_2^{i,j}(v_i, v_j)$ is the heuristic on sequences i and j starting at v_i and v_j respectively. The proposed admissible heuristic uses the actual costs of aligning the 2-dimensional sequences from v_i to t_i (the goal of sequence i) and from v_j to t_j (the goal of sequence j). We can write $h_2^{i,j}$ as,

$$h_2^{i,j}(v_i, v_j) = g^{i,j}[(v_i, v_j), (t_i, t_j)] \tag{3.3}$$

where $g^{i,j}$ is the actual calculated optimal aligned cost from (v_i, v_j) in sequences S_i and S_j to their respective goals (t_i, t_j).

In the case above of three sequences, although the three 2-dimensional problems are significantly smaller than the 3-dimensional problem, the cost of calculating the sum of the 2-dimensional alignments from each point in the 3-dimensional space is significant. If we can, however, retain information from the 2-dimensional problems and reuse calculated costs, significant savings in time would be possible.

Consider the effect of reversing the starting and goal nodes in the 2-dimensional problem, beginning at the goal node and proceeding to the starting node of the 2-dimensional problem using Dijkstra's algorithm. Figure 3.9 shows this reversal in which the new starting node was actually

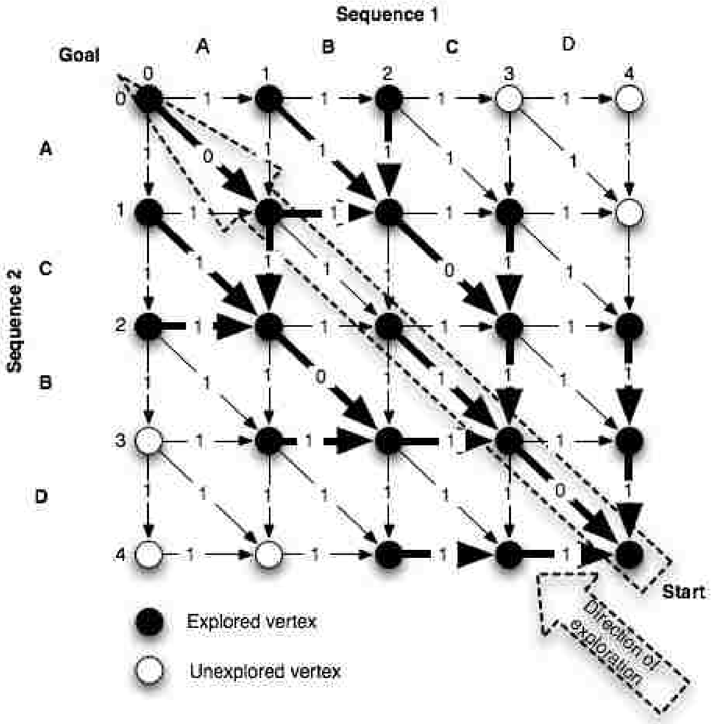


Figure 3.9: Reversing the direction of exploring the network from the goal node toward the starting node, the cost from an explored node to the goal is known; the Reverse Dijkstra Method.

the goal node of the original problem. The solid arrows show the edge directions of the original problem. Beginning with the new starting node Dijkstra's algorithm will calculate the actual cost from each node to the new starting node. The bold arrows in Figure 3.9 show the edges that are explored and the solid circles indicate the nodes where the cost is known to the goal node. Retaining this cost information from the nodes to the goal node of the original problem dramatically speeds up the calculations of the heuristic costs in the pairwise sum of equation (3.2) as the n-dimensional problem progresses. Should the heuristic pairwise sum require cost information from a node not calculated during the initial run, it is possible to set the unknown node as the goal node of the 2-dimensional problem, continue the minimum cost calculations to the new goal, adding the nodes to those known by the 2-dimensional problem.

In this research, Dijkstra's algorithm was used in all of the 2-dimensional cases. There is value in using Dijkstra's algorithm, although it explores far more space than needed, in that the resulting information about the cost from the goal node to many nodes in the space is used as the n-dimensional problem explores space around the optimal path. Future work may compare using another heuristic in the 2-dimensional problem to see if there are additional savings.

Proof of Admissibility

The next question is whether this heuristic is admissible. For $h_n(\underline{\mathrm{v}})$ to be admissible we need to show that the actual cost from $\underline{\mathrm{v}}$ to the goal node $\underline{\mathrm{t}}$ is greater than the heuristic value. Let $h^*_n(\underline{\mathrm{v}}) = g_n(\underline{\mathrm{v}}, \underline{\mathrm{t}})$, the actual cost from $\underline{\mathrm{v}}$ to $\underline{\mathrm{t}}$. We then need to show that

$$h^*_n(\underline{\mathrm{v}}) \geq h_n(\underline{\mathrm{v}}) \geq 0 \tag{3.4}$$

for all $\underline{\mathrm{v}} \in E$.

The gist of the proof is in noting that the optimal alignment from any node $\underline{\mathrm{v}}$ to the goal node $\underline{\mathrm{t}}$ can be broken down into pairwise "padded" alignments. The pairwise alignments are padded since in the n-sequence alignment it is possible for two $INDEL$ characters to be aligned, an event which would not occur in an optimal alignment of just two sequences. Further, each of these padded pairwise alignments from the n-dimensional case has a cost that is greater than or equal to the optimal 2-dimensional alignment of the sequences involved; a result of Dijkstra's minimum cost algorithm, which guarantees that there is no alignment of two sequences whose cost is smaller. Therefore the cost of a pairwise alignment extracted from any n-sequence alignment will be greater than or equal to an optimal alignment of just the two sequences alone. The proof of this follows.

Consider n sequences, $S_1 \ldots S_n$ to be aligned. An alignment consists of an ordered sequence of n-tuples, where

$$\begin{aligned} &Align^a(S_1, S_2, \ldots, S_n) \\ &= [({}^a x_1^1, {}^a x_1^2, \ldots, {}^a x_1^n), \ldots, ({}^a x_p^1, {}^a x_p^2, \ldots, {}^a x_p^n)] \end{aligned} \qquad (3.5)$$

is the alignment labelled "a". The n-tuple $({}^a x_1^1, {}^a x_1^2, \ldots, {}^a x_1^n)$ represents a specific alignment of characters, one from each sequence, where ${}^a x_i^j \in S_j \cup \{INDEL\}$ and represents the ith character from alignment a of sequence j. Let $p^a = length(Align^a(S_1, S_2, \ldots, S_n))$, the length of alignment a. Because of the addition of the $INDEL$ characters, it is possible for the alignment to be longer than any one sequence, hence

$$p^a \geq \max_i(m_i).$$

It should be noted that the order of the characters from any one sequence is preserved, with the possible insertion of $INDEL$ characters at any point.

The cost assigned to an alignment is

$$\begin{aligned} &Cost(Align^a(S_1, \ldots, S_n)) \qquad (3.6) \\ &= \sum_{i=1}^{p} C_n({}^a x_i^1, {}^a x_i^2, \ldots, {}^a x_i^n) \end{aligned}$$

and the cost of an optimal alignment $Align^*(S_1, S_2, \ldots, S_n)$ is such that

$$\begin{aligned} g_n(\underline{\mathrm{s}}, \underline{\mathrm{t}}) &= Cost(Align^*(S_1, \ldots, S_n)) \\ &= \min_a(Cost(Align^a(S_1, \ldots, S_n))). \end{aligned}$$

Since the heuristic is dealing with partial sequences as the algorithm explores various nodes and edges in the network, we will designate $S_i(c_j)$ to be

$$S_i(c_j) = (c_j, \ldots, c_{m_i})$$

a partial sequence from S_i, starting at c_j and proceeding to the end of the sequence where $1 \leq j \leq m_i$.

From equations (3.2) and (3.3), $h_n(\underline{\mathrm{v}})$ can be written as

$$\begin{aligned} h_n(\underline{\mathrm{v}}) &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} h_2^{i,j}(v_i, v_j) \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} g^{i,j}[(v_i, v_j), (t_i, t_j)] \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Cost(Align^*(S_i(v_i), S_j(v_j))) \qquad (3.7) \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{p^*} C_2^{i,j}({}^*x_k^i, {}^*x_k^j) \end{aligned}$$

where $({}^*x_k^i, {}^*x_k^j) \in Align^*(S_i(v_i), S_j(v_j))$. Recall, however, that $C_2^{i,j}(x_i, x_j) \geq 0$ for all x_i and x_j. Therefore $h_n(\underline{\mathrm{v}}) \geq 0$.

We now need to show that $h_n^*(\underline{\mathrm{v}}) \geq h_n(\underline{\mathrm{v}})$ for all $\underline{\mathrm{v}} \in U$.

Let $n*$ designate an optimal n-dimensional alignment. Starting with the actual cost from any node $\underline{\mathrm{v}}$ to the goal node $\underline{\mathrm{t}}$, we can write

$$h_n{}^*(\underline{\mathrm{v}}) = Cost(Align^{n*}(S_1(v_1), S_2(v_2), \ldots, S_n, (v_n)))$$

$$= \sum_{i=1}^{p^{n*}} C_n({}^{n*}x_i^1, {}^{n*}x_i^2, \ldots, {}^{n*}x_i^n).$$

From equation (3.1) we can expand C_n as

$$= \sum_{i=1}^{p^{n*}} \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} C_2({}^{n*}x_i^j, {}^{n*}x_i^k)$$

and from equation (3.6) we can simplify this as

$$= \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} Cost(Align^{n*}(S_i(v_i), S_j(v_j))). \tag{3.8}$$

Note that $Align^{n*}(S_i(v_i), S_j(v_j))$ is the 2-dimensional pad-ded alignment of $S_i(v_i)$ and $S_j(v_j)$ extracted from the n-dimensional optimal alignment “n*”. This is not necessarily the same as $Align^*(S_i(v_i), S_j(v_j))$ which is the optimal 2-dimensional alignment of $S_i(v_i)$ and $S_j(v_j)$. In fact based on Dijkstra’s minimum cost algorithm we may say,

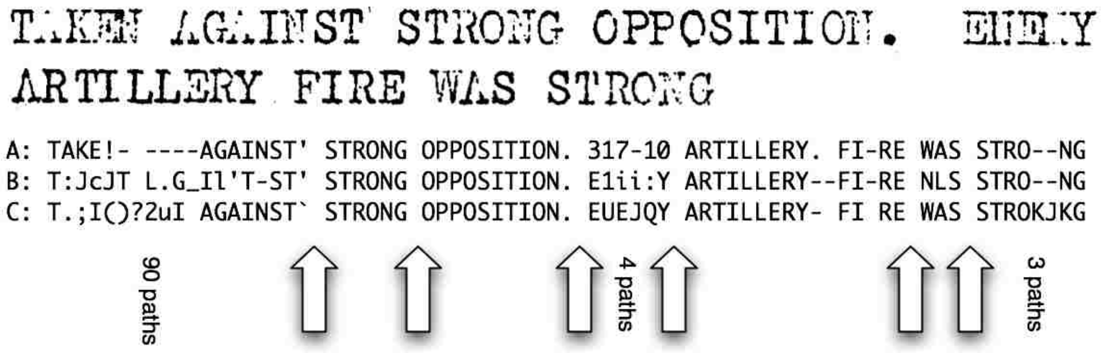
$$Cost(Align^*(S_i(v_i), S_j(v_j)))$$

$$= \min_a(Cost(Align^a(S_i(v_i), S_j(v_j))))$$

and consequently,

$$Cost(Align^*(S_i(v_i), S_j(v_j)))$$

$$\leq Cost(Align^{n*}(S_i(v_i), S_j(v_j))).$$



```
A: TAKE!- ----AGAINST' STRONG OPPOSITION. 317-10 ARTILLERY. FI-RE WAS STRO--NG
B: T:JcJT L.G_Il'T-ST' STRONG OPPOSITION. E1ii:Y ARTILLERY--FI-RE NLS STRO--NG
C: T.;I()?2uI AGAINST` STRONG OPPOSITION. EUEJQY ARTILLERY- FI RE WAS STROKJKG
```

T..KEN AGAINST STRONG OPPOSITION. ENE.Y
ARTILLERY FIRE WAS STRONG
90 paths
4 paths
3 paths

Figure 3.10: Communiqué No. 204, page 2. Points at which all OCR engines agree upon breaks between words are indicated by arrows.

Then by equations (3.7) and (3.8) we may write,

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} Cost(Align^{*}(S_i(v_i), S_j(v_j)))$$
$$\leq \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} Cost(Align^{n*}(S_i(v_i), S_j(v_j)))$$

Therefore, $h_n(\underline{\mathrm{v}}) \leq h_n^*(\underline{\mathrm{v}})$ and the heuristic is admissible.

3.3.3 Building the Word Hypothesis Lattice

Once the optimal alignments have been found, the next step is to build a lattice of word hypotheses. The lattice will represent the possible word alternatives from which the lattice word error rate will be measured. One of the problems with the OCR output is that whitespace (spaces, tabs, end of lines) can incorrectly occur within words, or can be absent, running words together. The word lattice is built on points in the optimal character alignment where all paths of the network converge on whitespace. This is the point where all of the possible alignments agree that there is a break between words. Figure 3.10 shows an example from Communiqué 204, page 2 in which six points of agreement on breaks between words are indicated by arrows. The “-” indicates the $INDEL$ character, which means that there is no corresponding character in the other sequences to be aligned with that location. The optimal alignment of the sequences can result in multiple optimal paths

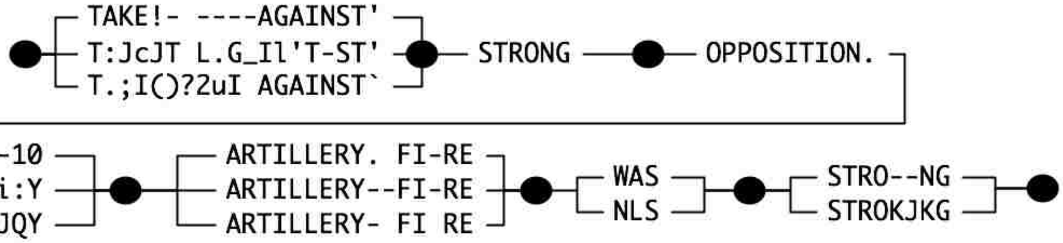


Figure 3.11: Communiqué No. 204, page 2. Word lattice constructed from Figure 3.10.

through the network, indicating alternative alignments. Where more than one path is found between lattice points, the number of paths is shown in Figure 3.10. The lattice of word alternatives is constructed as shown in Figure 3.11.

An example of extraneous whitespace is seen in the fifth lattice grouping in the word "FIRE" where OCR C inserts a space between "I" and "R". The second case, of running words together is seen in the same lattice grouping where "ARTILLERY" and "FIRE" are run together by OCR B, recalling that the "-" character represents an $INDEL$, meaning that the text appears as"ARTILLERYFIRE". Only OCR A correctly interpreted the text.

3.4 Results and Analysis

3.4.1 A* Alignment Results

The following results comparing Dijkstra's algorithm with A* using the RD heuristic do not use the entire data set, as we were limited to documents for which the multi-dimensional alginment problem could be solved by Dijkstra's algorithm in the available RAM (8 GB). Later results in this paper will include the entire development data set. Figure 3.12 compares the number of nodes of the network visited based on three algorithms: Dijkstra's minimum cost algorithm [54], and two implementations of the A* algorithm with different heuristics. (Note, that the A* algorithm with a heuristic value of 0 is Dijkstra's method.) The first, an optimistic heuristic, assumes that all possible matches in the sequences are made regardless of their order in the sequence. The Reverse Dijkstra

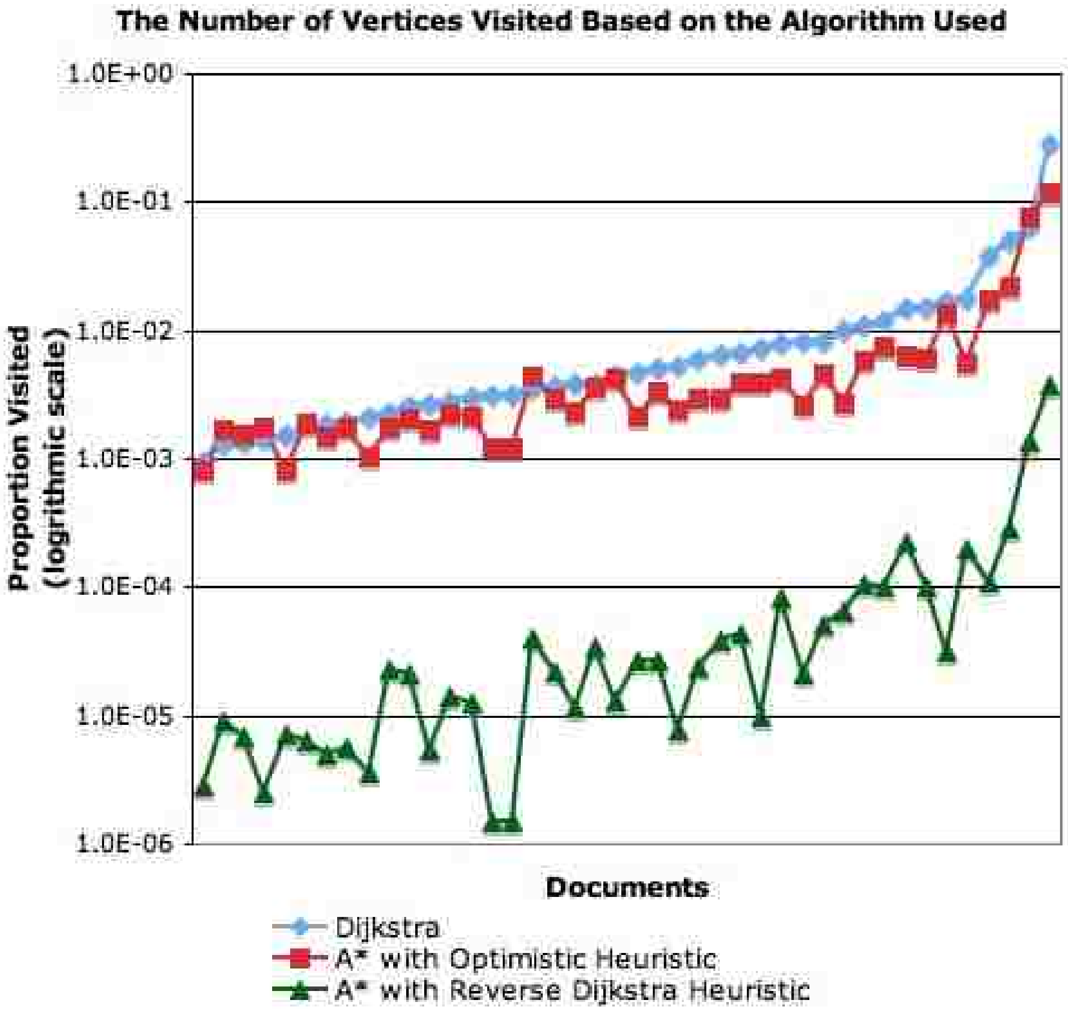


Figure 3.12: Comparing the percent of nodes visited. Sorted by increasing number of visited nodes by Dijkstra's algorithm.

Table 3.2: Comparison of the proportion of nodes visited by algorithm using the set of documents where Dijkstra's algorithm could be completed.

	Dijkstra	A* (Optimistic)	A* (RD)
Min % Visited	0.10%	0.08%	0.0001%
Max % Visited	28.51%	11.74%	0.3837%
Avg % Visited	1.58%	0.59%	0.0079%

heuristic is as described in Section 3.3.2 and shows significant improvement. Selected statistics can be found in Table 3.2.

Another way to view the results is to compare the number of nodes visited by each of the algorithms to the size of the network. Figure 3.13 lists these results for sets where Dijkstra's methodwas practical. It appears that the A* algorithm using the Reverse Dijkstra heuristic is able to maintain its advantage, even as the size of the network problem increases almost three orders of magnitude. A factor which likely affects the the performance of the alignment algorithms is the accuracy of the underlying sequences. Figure 3.14 displays the relationship between the average

OCR word error rate and the percent of the nodes that are visited by the A* algorithm using the Reverse Dijkstra heuristic. The best fit trend line was exponential with an R^2 of 0.663. This would indicate some relationship between the average OCR WER and the efficiency of the A* alignment algorithm. This is not unexpected.

3.4.2 Word and Lattice Error Rate Results

From the aligned character sequences two derivatives are possible: the word lattice of alternatives described in Section 3.3.3 and a sequence of words selected from the lattice based on their presence in a dictionary or place name gazetteer. The word lattice will provide a best case, lower bound on the error rate from the three sequences in that if any of the alternatives match the human transcription *sclite* will count it as a match. The dictionary-based approach determines whether any of the alternatives in the lattice match known English words or European place names, selecting one as the correct word. The resulting sequence is compared to the human transcription by *sclite* calculating

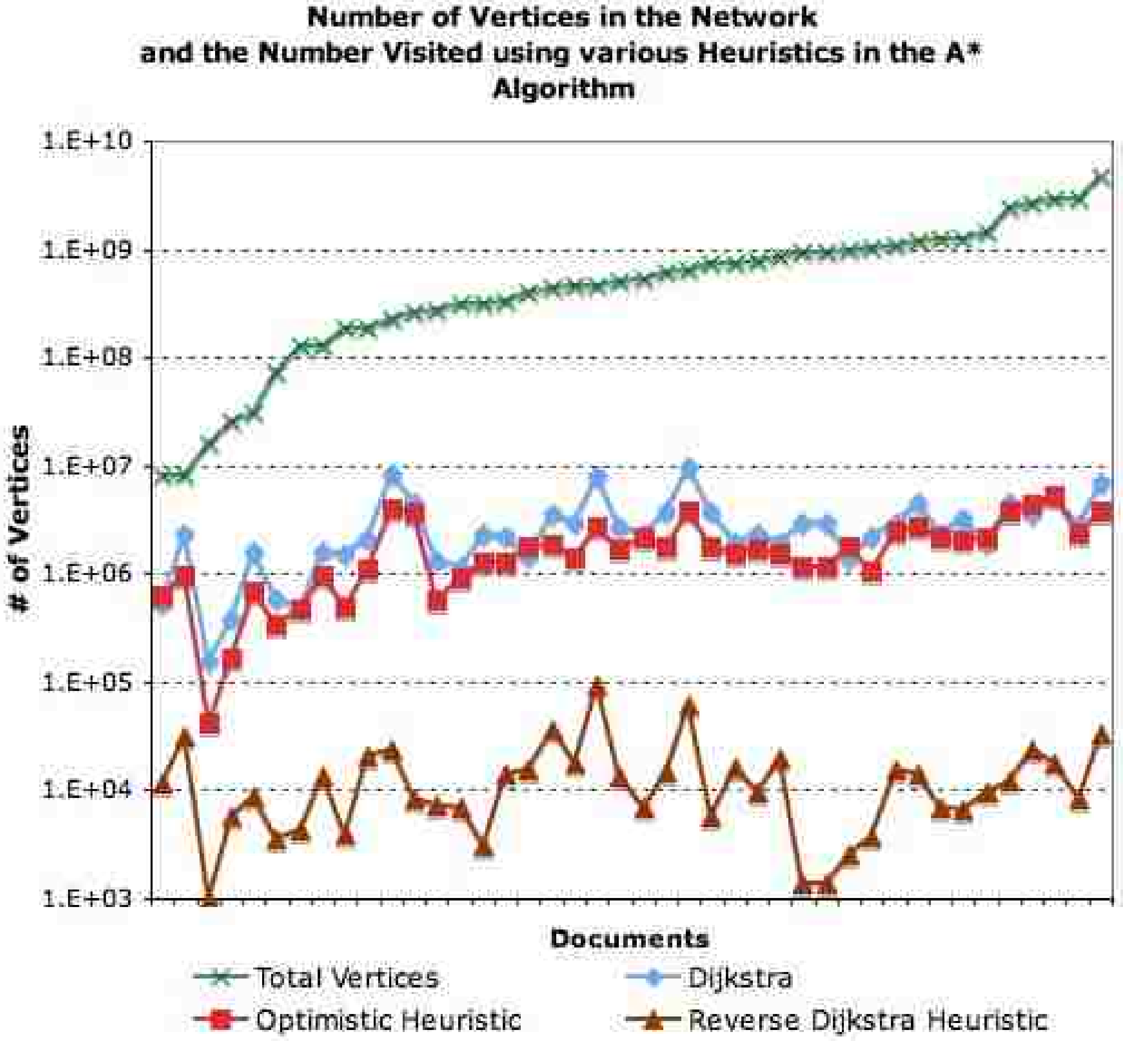


Figure 3.13: Comparing the total number of nodes and those visited by Dijkstra's method and two implementations of the A* algorithm. Sorted by increasing number of nodes in the network.

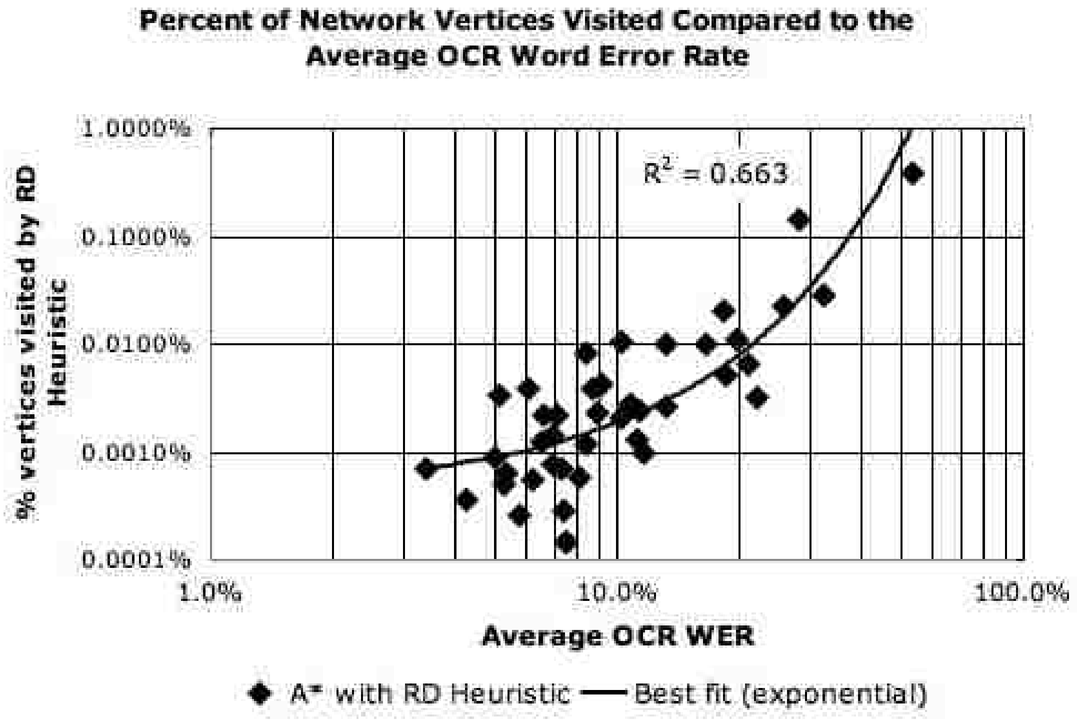


Figure 3.14: The relationship between the average word error rate of the OCR engines and the percent of nodes visited. $R^2 = 0.663$

Table 3.3: WER of OCR engines, the lattice error rate, and the WER of the dictionary-based word selection sequence.

OCR	A	B	C	Lattice	Dictionary
Avg	16.4%	25.9%	26.3%	10.3%	14.7%
Max	120.0%	125.0%	238.1%	105.6%	120.6%
Min	2.3%	1.5%	2.0%	0.6%	1.2%

Table 3.4: Largest and smallest differences between baseline OCR WER and the lattice error rate.

	Diff	A	B	C	Lattice
Largest	85.7%	7.0%	4.9%	7.7%	0.7%
Smallest	0.0%	3.6%	1.5%	5.1%	1.5%
	0.0%	3.1%	31.3%	50.0%	3.1%
	0.0%	13.3%	6.7%	12.4%	6.7%

the WER. It is possible for the WER to be greater than 100% as the method counts any number of insertions as individual errors.

Using the development set of documents, the lattice error rate is on average 55.0% lower (22.9% reduced to 10.3%) than the best individual OCR engine word error rate, while the sequence from the dictionary-based selection method saw an average reduction of 35.8% (22.9% to 14.7%). Table 3.3 extends Table 3.1 comparing the individual OCR WER with the error rate of the aligned word lattices and the dictionary-based word selection sequence. Figure 3.15 shows the range of baseline WER for the OCR output compared to the word lattice and the dictionary-based sequence. As expected, the error rate of the lattice is no greater than the minimum of the baseline WER of the OCR engines. In three documents the error rate of the lattice is equal to the minimum baseline WER of the OCR. Table 3.4 shows these extremes. Note that for three documents there is no difference between the minimum WER of the OCR output and the lattice error rate. In the cases where the lattice error rate was no different than the best of the OCR WERs, the OCR output of the other engines did not provide any additional correctly recognized words beyond those found in the best OCR.

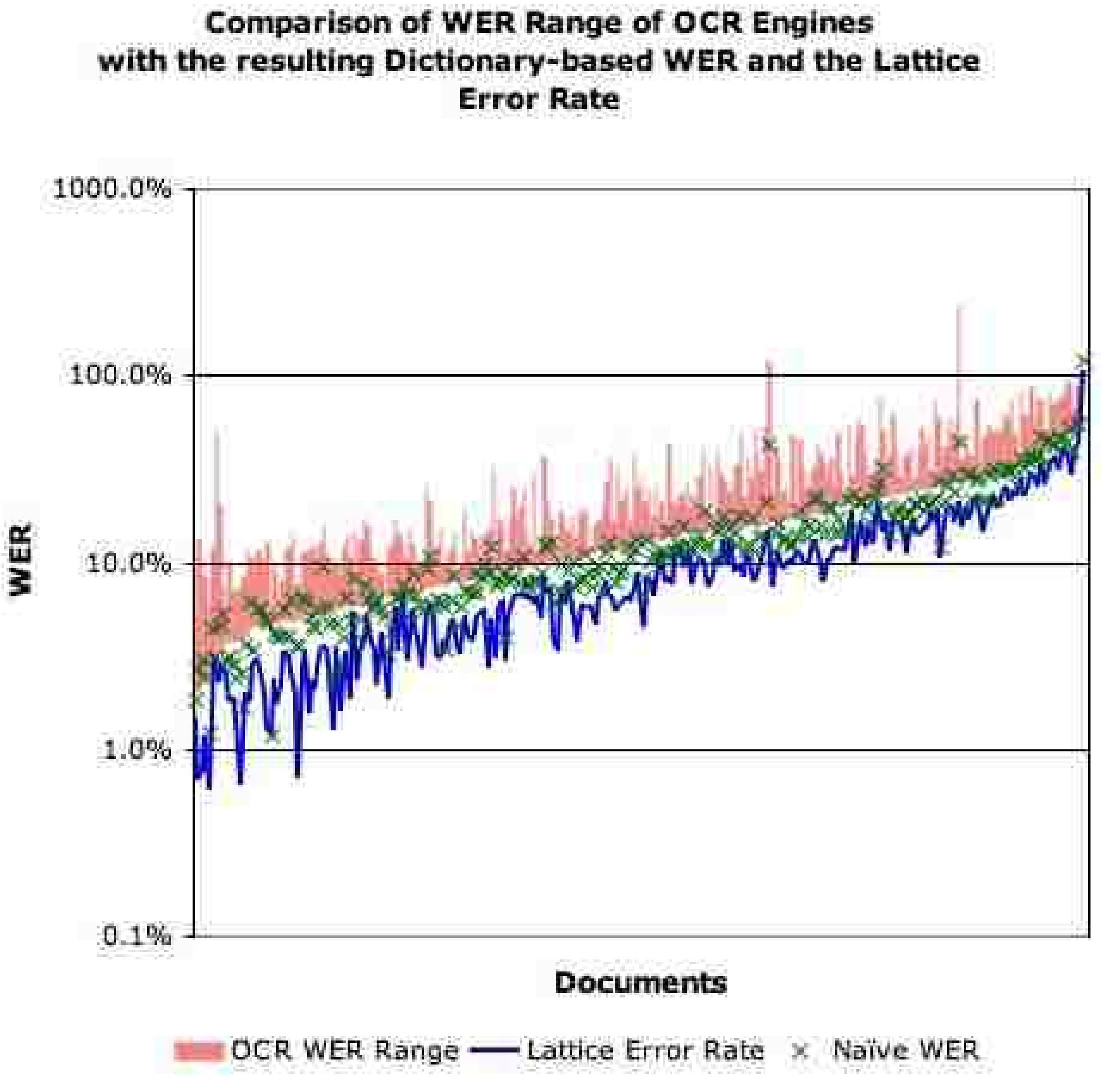


Figure 3.15: Comparing the range of the baseline OCR WER with the error rate of the lattice and dictionary-based word selection sequence. Sorted by the minimum baseline OCR WER of the three OCR engines.

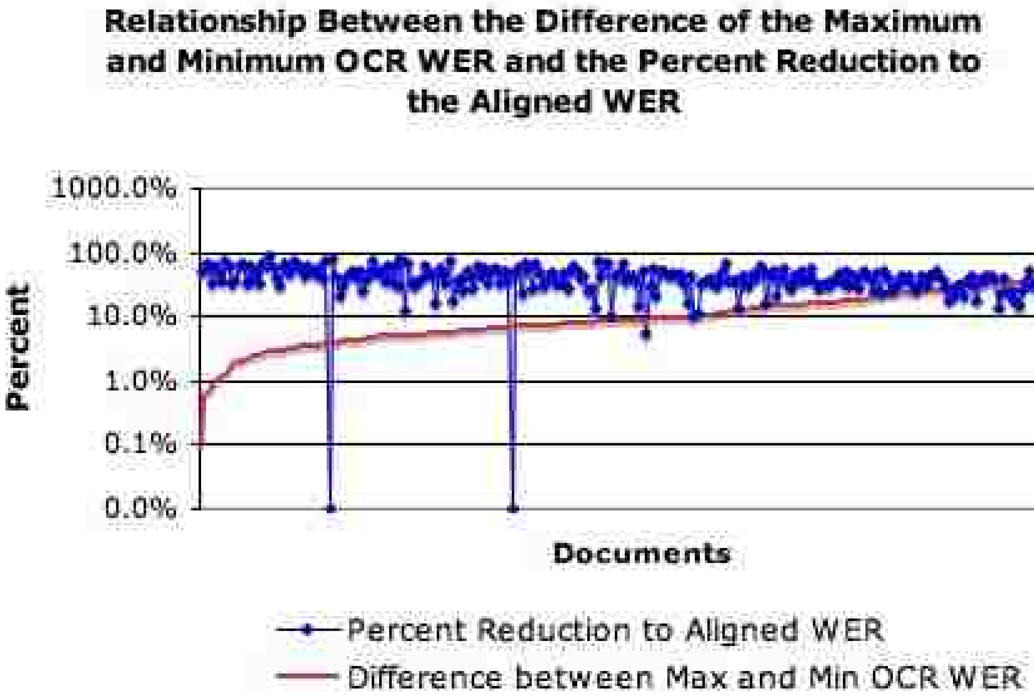


Figure 3.16: Comparing the percent reduction from the difference between the maximum and minimum baseline OCR WERs to the aligned word lattice WER. Sorted by the percent difference between the maximum and minimum WERs.

Figure 3.16 shows the relationship between the difference of the best and worst OCR WERs for a document and the reduction in the lattice error rate. In general as the difference in baseline OCR WERs increases the lattice error rate decreases. This can be interpreted as OCR output with higher WERs providing less help in providing alternate word hypotheses, which would reduce the aligned word lattice WER. The extremes of this are shown in Table 3.5. The table shows the WERs of two documents with the largest difference between the minimum and maximum baseline WERs and one document with the smallest difference between baseline WERs. Of interest is that the document with the smallest differences shows a 50% improvement in the WER of the aligned word lattice over the baseline while the documents with the largest differences show a smaller improvement. Note that the document with the OCR C WER of 238.1% is an example of runaway OCR in which the engine simply failed to recognize words due to extreme noise in the image.

Table 3.5: Differences between baseline OCR WERs and the lattice error rates. Two cases of the largest differences between the baseline OCR WER results and one case of the smallest difference between baseline OCR results.

OCR WER	A	B	C	Lattice
Largest	36.2%	26.7%	238.1%	21.0%
	21.8%	63.2%	78.2%	20.7%
Smallest	6.5%	6.5%	6.5%	3.2%

Clearly the results from the lattice are a best case. Compared to the dictionary-based word sequence, some improvement is still seen. Figure 3.17 shows a general improvement between the average OCR WER and the WER of the diction-ary-based word sequence. As can be seen in Figure 3.15 there are instances where the dictionary-base word selection does not improve upon the best OCR for a document. There are several reasons for this. First, if more than one OCR engine recognizes different English words, there is no mechanism in this work to select between them. An example of this occurs in Communiqué 1 where OCR A recognizes the word "June", which is correct and OCR B recognizes the same character images as "Tune". When this occurs, the current system selects one for the dictionary-based word sequence. If the wrong word is selected, then *sclite* will count this as a substitution error. The lattice does not have this problem. Another source of error is when none of the words in the lattice are found in the English word list or the European gazetteer. Lastly, if the author of the document incorrectly spelled a word, the OCR would attempt to recognize it in its misspelled state rather than attempting to correct it.

3.5 Conclusions and Future Work

The problem of retrieving text through optical character recognition of poor quality documents can be formulated as a noisy channel problem in which the original message of the document is corrupted by the scanning and OCR processes. This paper presents two innovative results in which the differences in OCR engine output can be used to advantage to extract the original text. First, we proposed the Reverse Dijkstra heuristic as an innovative component to the A* algorithm, which

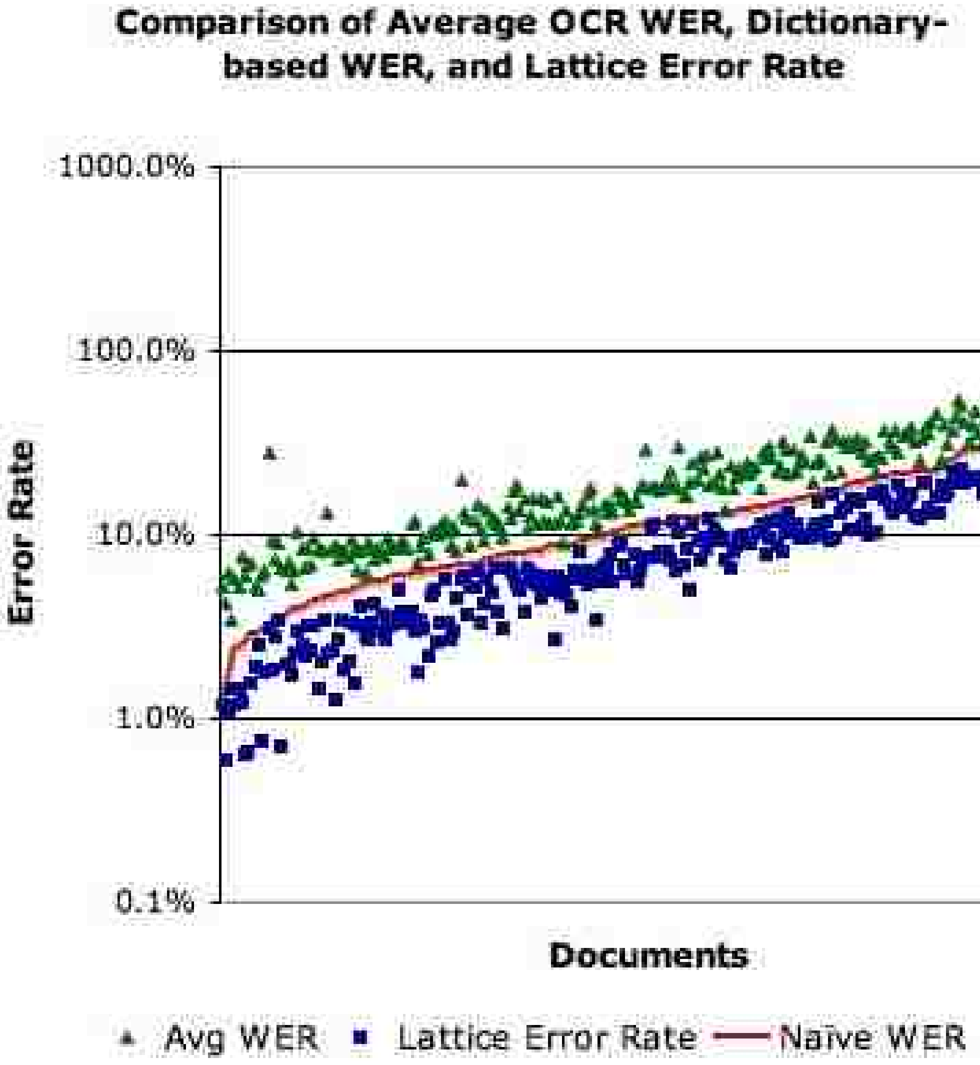


Figure 3.17: Dictionary-base word sequence error rate compared to the OCR average WER and the Lattice Word Error rate.

significantly reduced the number of nodes visited in the minimum cost alignment network. On average 0.0079% of the nodes were visited in the document collection used, making feasible the alignment of multiple text sequences of on the order of 2500 characters each. From the aligned character sequences we constructed a lattice of word hypotheses in which the error rate of the lattice showed an average 55.0% reduction from the best OCR word error rate of each document. Using a simple dictionary and gazetteer-based word selection method on the lattice, we were able to extract a sequence of words from the lattice of each document, in which the resulting WER showed an average reduction of 35.8% from the best OCR WER.

Several possible avenues of future research suggest themselves. First, we will explore whether the use of more than three OCR engines continues to improve the results. Where is the point of diminishing returns? Tuning the scanning and OCR parameters may also provide more hypotheses for the word lattice. As mentioned in Section 3.4.2, runaway OCR provides little improvement to the aligned word lattice. We plan to train on the development set of documents to recognize this and eliminate OCR results, which are not likely to contribute to the overall improvement of the aligned word lattice.

Using Dijkstra's algorithm as the pairwise A* heuristic covers a great deal of space, some of which may not be useful to solving the larger n-dimensional problem. A possible research direction will be exploring the use of other algorithms that would also provide optimal two-dimensional distance measures. Along these lines, Schroedl [93] suggests an interesting measure of efficiency by determining the frequency with which the heuristic misses calculating the value of a node explored by the A* algorithm.

We plan to select additional word hypotheses based on word edit distances to known words and place names, which will augment the word lattice providing a broader selection of words from which to pick. Lastly, we will use a language model to rerank the word hypotheses, selecting the sequence of words with the highest probability of being an English sentence. The ultimate research goal is to provide as good an automated transcription of poor quality documents as possible.

Chapter 4

Error Correction with In-Domain Training Across Multiple OCR System Outputs

4.1 Introduction

Major digitizing efforts are making pre-digital materials available on-line at an unprecedented scale. Our research leverages the variation among OCR engines (see Figure 4.1) and additional features of the OCR hypotheses to improve the output beyond what any single OCR engine is capable of. In this case, where in-domain training data is available, we improve upon our previous work [61] and show how a decision list trained on in-domain data using feature combinations reduces the word error rate beyond what is achieved using consensus voting or dictionary matching alone. Further, we explore using a spell checker to suggest additional words for hypotheses that do not appear in the dictionary or gazetteers..

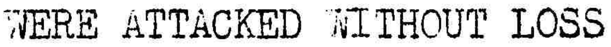
The remainder of this paper is organized as follows. Section 4.2 gives background on the problem, information about the data set used, and related work. Section 4.3 describes our methodology and presents the results in reducing the document OCR error rate using features alone and in combination. And Section 4.4 summarizes our conclusions and proposes future work.

4.2 Approach

4.2.1 Background

The documents used in this paper are the Eisenhower Communiqués [43], a collection of 610 facsimiles of typewritten documents created by the Supreme Headquarters of the Allied Expeditionary

Published in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011)* [63].


WERE ATTACKED WITHOUT LOSS

OCR A: 7JERE ATTACKED WITHOUT LOSS

OCR B: WERE ATTACKED ;ITHOuT LOSS

Figure 4.1: Poor quality text from Eisenhower Communiqué No. 233a along with OCR output.

Force (SHAEF) during the last years of World War II. Having been typewritten and duplicated using carbon paper, the quality of the print is very poor. Many documents have artifacts of the duplication process, further complicating the text recognition task. A manual transcription of these documents serves as the gold standard for evaluating the word error rates of the OCR and the feature weighting process described in Section 4.3. The documents have been randomly divided into three sets, each roughly one third of the total collection: a training set, a development test set and a blind test set. The blind test set of documents is reserved for future work.

4.2.2 Related Work

OCR, particularly for historical documents continues to be an area of open research. Hull [38] notes that even small character error rates result in significantly higher word error rates, which negatively affects the usefulness of the OCR in document searching and other tasks. He calculates that a 1.4% character error rate can lead to a 7.0% word error rate in a document with 2,500 characters and 500 words. Kae and Learned-Miller [44] confirm that although the OCR of modern clean documents is effectively a solved problem, older degraded documents present difficulties.

Many approaches have been taken to reduce the error rate of OCR output. Lopresti et al. [59] use consensus voting of characters between multiple scans of the same document recognized by the same OCR software. The National Library of Medicine [102] selected Prime OCR, a commercial system that votes on responses from multiple OCR engines, to improve word recognition. The authors' previous work [61] also explored voting in conjunction with feature engineering. The approach reported in this work uses distinct OCR engines to find variations in the text recognized by the systems, and similar to Lopresti, we find that voting plays an important role in making good corrections. Using character confusion learned from the document itself, Kae and Learned-

Miller [44] use a weighted English lexicon to provide word hypotheses for unknown tokens. Likewise Strohmaier et al. [99] use a dictionary and a Levenshtein edit distance for post-OCR error correction. Wick et al. [111] explore using a weighted lexicon created from a topic model rather than a dictionary. This research will show that the dictionary matching process in conjunction with aligned hypotheses from multiple engines does in fact result in a lower word error rate, but that using additional features beyond these two improves the results.

Similar to the text error correction task, error correction in speech recognition has employed related methods. For instance, Ringger [86, 87] explored statistical methods in speech recognition post-processing to correct errors. Likewise Mangu et al. [71] took a lattice of alternatives from a speech recognizer and proposed a method for creating a probabilistically consistent lattice for word error minimization. Our work likewise uses a lattice of alternatives, but rather than consisting of alternatives generated by a single source, we use the alignment of multiple sources to populate the lattice.

Combinations of multiple models or systems have been shown to provide improved results. For example, Fiscus [28] votes among multiple speech recognition systems and Nakano et al. [75] use multiple OCR inputs and alignment of lines in the text to improve OCR recognition. We will show that the combination of feature weighting with multiple OCR alignment results in improvements beyond voting or dictionary matching alone.

4.3 Methodology and Results

4.3.1 Baseline OCR Results

Each of the documents in this study was evaluated using three OCR engines, two commercial and one open source, referred to here as OCR A, OCR B, and OCR C respectively. The same digitized image file was evaluated by each OCR engine, and their respective word error rates were calculated with Sclite [3], a tool provided by NIST for use in speech recognition research. Sclite calculated the lattice word error rates and word error rates, shown in Table 4.1.

Table 4.1: Baseline error rates.

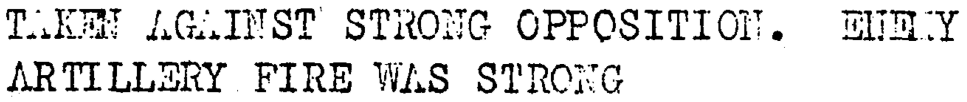
	OCR A WER	OCR B WER	OCR C WER	Mean WER	Lattice WER
Mean	19.9%	30.4%	50.1%	33.5%	13.0%
Minimum	2.3%	1.5%	3.9%	3.4%	0.7%
Maximum	80.7%	111.7%	1,666.4%	602.2%	78.4%

The texts from the three OCR engines are character aligned using the A* algorithm with the Reverse Dijkstra admissible heuristic described by Lund and Ringger [61]. From this character level alignment we construct a lattice of word hypotheses such that wherever there is agreement across all engines on the location of white space we construct a column of hypotheses. The order of hypotheses in the column is determined by the overall accuracy of the OCR engine, with the lowest WER being first. (See Figure 4.2.) From this lattice of word hypotheses we calculate the lattice word error rates shown in Table 4.1 by comparing the true transcription with all of the aligned hypotheses from the OCR output. If any of the hypotheses in the column match the true transcription it is considered a match. This score provides a lower bound on the error rate that is possible using the evidence found only in the OCR outputs.

4.3.2 Features of the Aligned OCR Text

The alignment creates columns of alternative hypotheses suggested by the OCR engines. Each of the hypotheses is evaluated for features that may be indications of its accuracy. Following are the features used in the experiments reported in this paper.

1. **Voting** [V:#]: The count of hypotheses within a column that match the current hypothesis exactly. For example, the feature V:3 indicates that the hypothesis in question matches two other hypotheses in the column.
2. **Dictionary** [D]: A binary indicator for whether a hypothesis appears in the Unix dictionary.
3. **Gazetteer** [G]: A binary indicator for whether a hypothesis appears in a gazetteer of European place names.



TAKEN AGAINST STRONG OPPOSITION. ENEMY
ARTILLERY FIRE WAS STRONG

A:	TAKE!	[D]	AGAINST	[D]	STRONG	[V:3, D]
B:	T:JcJT	[]	L.G_Il'TST'	[]	STRONG	[V:3, D]
C:	T.;I()?2uI	[]	AGAINST‘	[D]	STRONG	[V:3, D]
S:						
A:	OPPOSITION.	[V:2, D]	31710	[N]		
B:	OPPOSiTIOV.	[]	E1ii:Y	[]		
C:	OPPOSiTION.	[V:2, D]	EUEJQY	[]		
S:	OPPOSITION	[S:1]	ENJOY	[S:1]		

Figure 4.2: Aligned output of the OCR engines from part of Communiqué No. 204 with assigned features. “S:” indicates word hypotheses added by the spell checker.

4. **Number** [N]: A binary indicator for whether a hypothesis is a number.

5. **Recurring** [R]: A binary indicator for whether the hypothesis appears in a list of tokens that do not appear in the dictionary or gazetteer but repeat in the corpus.

6. **Spell** [S:#]: For hypotheses that do not appear in the dictionary or gazetteer, we use the GNU Aspell [5] spell checking software to add additional hypotheses, which may suggest correct words not found in the OCR.

Section 4.3.3 below explores the word error rate results when using a single feature to select a hypothesis from the column of alternatives, and Section 4.3.4 presents the results when using a decision list trained on a combination of features.

4.3.3 Untrained Single Feature Success Rates

The first question we address is what is the potential for individual features for identifying the correct hypothesis without the aid of parameter training? The features of each hypothesis are calculated, and each hypothesis itself is compared to the true word from the transcription. The results reflecting the potential usefulness of individual features are shown in the columns of Table 4.3 at the bottom.

Based on our observations and those of related work, we expected that **Voting** would be a powerful indicator of success in identifying the actual word and the results bear this out. **Dictionary**

Table 4.2: Success rates of disjoint and combined features on the training set.

Features	Instances	% Correct
Vote:# and not **Dictionary**	16,2123	71.4%
Dictionary and not **Vote:#**	6,873	41.3%
Vote:# and **Dictionary**	84,640	96.2%

matches were also a strong indicator of the underlying word from the document, with **Gazetteer** matches having fewer instances but still showing promise in indicating the true word from the document. The result using the dictionary is consistent with our earlier paper [61], which showed a reduced WER when using that feature alone.

4.3.4 Combining Features

Encouraged by the potential of our chosen features, we next consider the effects of combining features. Consider instances where a **Vote**:# feature is present (meaning either Vote:3 or Vote:2 is present), but the hypothesis is not found in the dictionary. In Table 4.2 we see that when the **Vote**:# feature is false, combined with the **Dictionary** feature, the potential success rate is significantly lower. Likewise, the **Dictionary** feature combined with the absence of a **Vote**:# feature has a lower success rate. Finally, if we combine just the **Vote**:# features and the **Dictionary** feature the combined success rate is higher than those features alone. These potential success rates lead us to believe that features taken together provide a better indicator of whether a hypothesis is more likely to be the true underlying word from the document than features considered individually.

Using the in-domain training set, we evaluated the success rates of combinations of features, as shown in Table 4.3. The methodology was similar to that described in Section 4.3.3 for individual features; however, in this case we calculate the success rate of combinations of features observed in the training data.

Looking at the results in Table 4.3, note that the combination of the **Vote:3** feature (all three OCR engines output the same hypothesis) combined with the **Dictionary** feature (the hypothesis is found in the dictionary) gives a very high potential success rate of 97.2%, higher than **Vote**:#

Table 4.3: Potential success rates (percent correct) of combinations of features with the cut-off point of the decision list (at the double line) learned from the training set. (See Section 4.3.5.)

	Feature Combinations										Training Results	
	Vote:3	Vote:2	Dictionary	Gazetteer	Number	Recurring	Spell:1	Spell:2	Spell:3	(No Features)	Instances	% Correct
	●								●		46	97.8%
	●		●								52,281	97.2%
	●				●						666	96.4%
	●		●	●							14,409	96.2%
		●	●								14,124	93.6%
	●					●					316	93.0%
		●	●	●							3,826	92.5%
		●			●						362	91.7%
	●			●							1,308	90.4%
		●		●							1,266	90.4%
		●				●					232	80.6%
		●					●				1,251	77.9%
		●						●			26	76.9%
	●										941	69.7%
		●							●		13	69.2%
				●							647	64.6%
		●									9,786	62.2%
			●								5,041	43.1%
			●	●							1,832	36.6%
					●						306	17.0%
						●					489	12.3%
								●			9,061	4.6%
										●	23,178	4.0%
							●				37,886	2.8%
									●		5,766	1.0%
Instances	30,886	69,967	91,513	23,288	1,334	1,037	39,137	9,087	5,825	23,178		
Combined	100,853						54,049					
% Correct	96.4%	82.6%	92.1%	89.4%	76.9%	52.2%	5.2%	4.8%	1.9%	4.0%		
Combined	92.2%						4.8%					

features or the **Dictionary** feature individually. Of greater interest is the fact that the **Vote:2** feature (two of the three OCR engines agree) combined with the **Dictionary** feature still has a high success rate (93.6%), which is also higher than a **Vote**:# or the **Dictionary** features individually. Overall the **Vote**:# feature for all values, when combined with other features, is a strong indicator of the underlying word from the document. **Vote:3** and **Vote:2** alone, without any other features being present, are significantly less indicative of the underlying word.

Another interesting point from Table 4.3 is contra-indicators, specifically feature combinations that would seem to indicate that the hypothesis should be excluded all together. For example, the lack of any features is a strong indicator that the hypothesis is not found in the document.

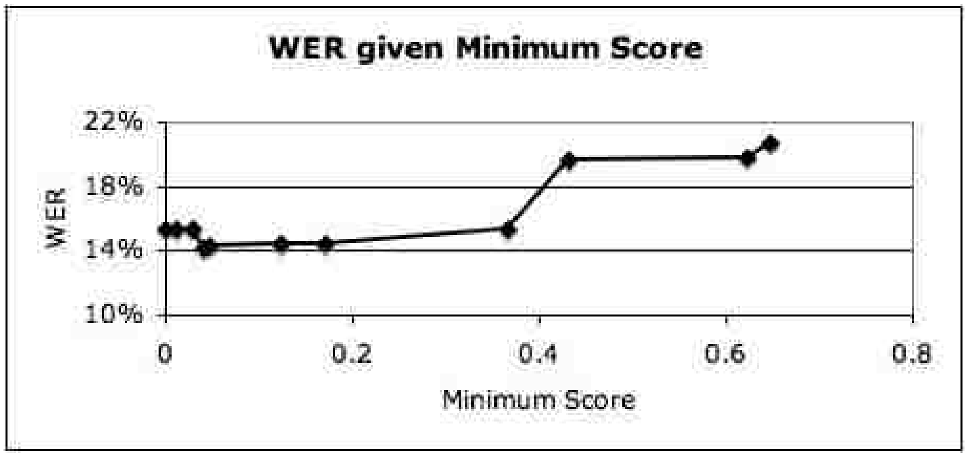
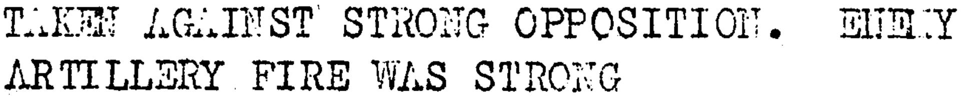


Figure 4.3: WER on the training set for minimum scores required to be included in the output text.

4.3.5 Decision List Training

One way to take advantage of the information learned in Table 4.3 is to employ the table as a decision list (defined by Rivest [88]) to score each hypothesis in each column, creating a two step process: 1) Using the training set, learn the correctness percentage for each combination of features that appears in the training set. 2) Apply these learned rates to the development test set as follows: (a) for each word hypothesis in each document in the development test set, calculate the feature set and look-up the correctness score learned from the training set. (b) From the scores of the hypotheses in an aligned column, select the hypothesis with the highest scoring combination of features. (See Figure 4.4.) To break ties, choose the first hypothesis in the column.

Is it possible that some features should be an indicator that the hypothesis should be discarded? Not all of the feature combinations have strong potential success rates, and hence doubtful that they are indicative of a true word. Using the combined feature accuracy rates shown in Table 4.3, a naïve approach is to reject all hypotheses with features whose combinations have a score below 50%. On the training set we explored varying the lower bound at which a hypothesis may be selected for output. The lowest WER was achieved when the minimum threshold was set at 4.1%, which excluded hypotheses with **No Features**, **Spell:1** and **Spell:2** features. The plot in Figure 4.3 shows the relationship between the score and the resulting WER.



TAKEN AGAINST STRONG OPPOSITION. ENEMY
ARTILLERY FIRE WAS STRONG

A:TAKE!	[D]	(43.1%)	AGAINST	[D]	(43.1%)
B:T:JcJT	[]	(4.0%)	L.G_Il'TST'	[]	(4.0%)
C:T.;I()?2uI	[]	(4.0%)	AGAINST'	[D]	(43.1%)
S:					
O:TAKE!			AGAINST		
T: incorrect			correct		

A:STRONG	[V:3, D]	(97.2%)	OPPOSITION.	[V:2, D]	(93.6%)
B:STRONG	[V:3, D]	(97.2%)	OPPOSiTIOV.	[]	(4.0%)
C:STRONG	[V:3, D]	(97.2%)	OPPOSITION.	[V:2, D]	(93.6%)
S:			OPPOSITION	[S:1]	(2.8%)
O:STRONG			OPPOSITION.		
T: correct			correct		

A:31710	[N]	(17.0%)	ARTILLERY.	[D]	(43.1%)
B:E1ii:Y	[]	(4.0%)	ARTILLERYFIRE	[]	(4.0%)
C:EUEJQY	[]	(4.0%)	ARTILLERY	[D]	(43.1%)
S: ENJOY	[S:1]	(2.8%)	ARTILLERY-FIRE	[S:1]	(2.8%)
O:31710			ARTILLERY.		
T: incorrect			incorrect		

Figure 4.4: Partial aligned output of the OCR engines from Communiqué No. 204 with assigned features and scores. Hypotheses selected for output are underlined. Numbers in parentheses are assigned weights from Table 4.3. "O:" indicates the word selected for output by the system. "T:" indicates whether the selected hypothesis is correct or not.

Table 4.4: Comparing baseline error rates (column group one) to results from using **Vote** and **Dictionary** features alone (column group two), and combined features (column group three).

	OCR A WER	OCR B WER	OCR C WER	Mean WER	Lattice WER	Voting WER	Dictionary WER	Combined Features WER
Mean	19.88%	30.37%	50.13%	33.46%	12.96%	22.61%	18.70%	16.01%
Minimum	2.27%	1.45%	3.85%	3.39%	0.61%	1.21%	1.21%	0.61%
Maximum	80.68%	111.68%	1,666.42%	602.19%	78.41%	95.45%	86.36%	87.50%

If the correctness score of the features of none of the hypotheses in the column exceeded 4.1%, then the hypotheses are excluded from the output. If there is a tie for the highest score, the hypothesis from the first OCR engine in the alignment with that score is selected. All of the selected hypotheses are evaluated against the transcription to calculate the final word error rate for each document. An example of this is shown in Figure 4.4.

4.3.6 Combined Features Results With Training

We applied the method of Section 4.3.5 to the OCR output files of the development test set; the results are shown in Table 4.4 under the heading "Combined Features WER". This method has a lower word error rate than the methods using the **Dictionary** or the **Voting** features alone, with a 52.2% relative improvement on the mean word error rate of the three OCR engines and 19.5% relative improvement on the best OCR word error rate. Of particular interest is that the word error rate for five documents matches the lattice word error rate, the lower bound for accuracy based on the evidence from the aligned OCR outputs. Figure 4.5 compares the system described here using a combination of features and a decision list with our previous work [61] with this corpus, which used the **Dictionary** feature alone. Figure 4.6 shows that for the majority of documents the WER of the single feature **Voting** system is higher than the WER of this combined features system. Feature combinations help substantially.

4.4 Conclusions and Future Work

This paper presents new insight into the process of using multiple OCR engines and the value in exploiting the variations among engines. We relied on multiple features and trained the scores of the features using success rates on the training set as the basis for a decision list. The resulting method achieved a 52.2% relative improvement over the mean OCR baseline and a 19.5% relative improvement over the mean document WER of the best OCR engine. Further, in a few cases the minimum WER actually matched that of the lattice word error rate. Our goal remains to match and hopefully beat the lattice word error rate.

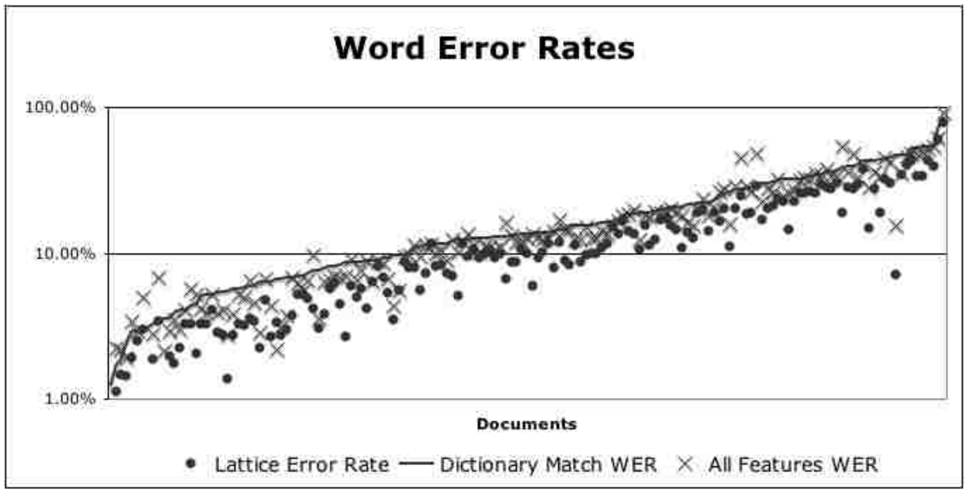


Figure 4.5: Comparison of the lattice word error rate, dictionary-based selection WER, and error rate using all features.

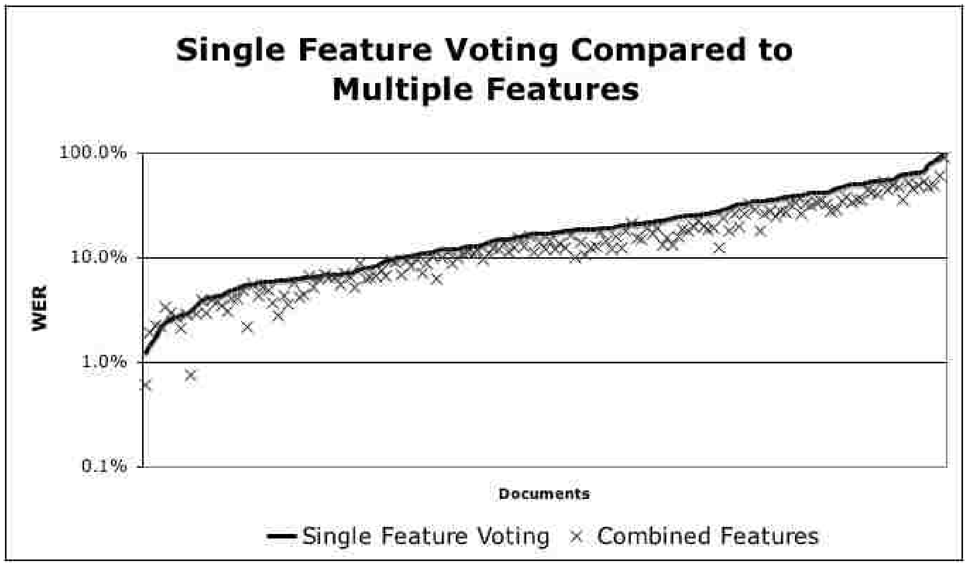


Figure 4.6: Word error rates with Voting as a single feature versus combined features.

The underlying problems with some documents will prevent getting anything close to a transcription. Sometimes the true transcription for certain hypotheses is not to be had; however, in regions of the document where good transcriptions can be recovered, those transcriptions can still provide value. Future work should explore the degree to which regions of document image quality can be identified.

In conclusion this paper has explored the use of in-domain training; where in-domain data is not available, techniques for using out-of-domain training data such as those explored in Lund, Walker, and Ringger (2011) [64] can be used.

4.5 Addendum

Subsequent to the writing of this paper, Volk, et al. [107] published a work which also uses aligned output from multiple OCR engines. Their method consists of three steps. First, a "polisher module" the corrects common errors in the corpus and OCR engine. This requires specific knowledge of the corpus and the confusion matrix of the OCR engines on that corpus. Both of these require an in-domain training set. Second, they align the OCR output sequences using a pair-wise heuristic, similar to our work in [64]. Their approach to alignment assumes that there is very little noise in the OCR hence they do not perform an explicit algorithmic alignment, but rather traverse the OCR sequences until a character mis-match is found. Within forty characters of the mis-match their method considers all possible combinations seeking a sequence with the highest bi-gram probability of the language model, which was created from the output of one of the OCR engines, which also implies in-domain training. Last, they replace unknown tokens with close orthographic variants, in other words a spell checker. This work by Volk, et al. uses similar techniques as this chapter, relying heavily on in-domain training on a printed corpus with limited noise. Subsequent chapters will extent my results beyond the need for in-domain training.

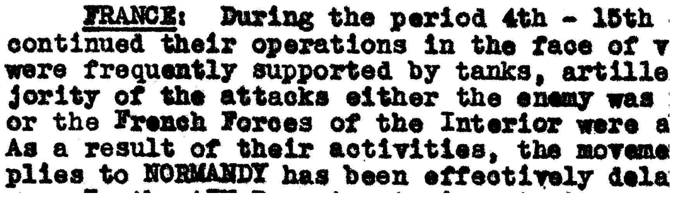
Chapter 5

Progressive Alignment and Discriminative Error Correction for Multiple OCR Engines

Abstract

This paper presents a novel combination of progressive textual alignment from multiple optical character recognition (OCR) engines using maximum entropy methods trained on a synthetic calibration data set to select between aligned hypotheses of the OCR engines. This paper presents a novel method for improving optical character recognition (OCR). The method employs the progressive alignment of hypotheses from multiple OCR engines followed by final hypothesis selection using maximum entropy classification methods. The maximum entropy models are trained on a synthetic calibration data set. The multiple OCR sequences are aligned with a progressive alignment method, originally from bioinformatics, achieving good, although not necessarily optimal results. Although progressive alignment is not guaranteed to be optimal, the results are nonetheless strong. An evaluation of the lattice word error rate (LWER) indicates that the order in which the results from multiple OCR engines are incorporated in the progressive alignment has little effect on the ultimate word error rate of the word lattice. Maximum entropy methods trained on an out-of-domain calibration set consisting of synthetically generated and degraded documents show positive results in leveraging the information found in the lattice to choose the best possible correction. The synthetic data set used to train or calibrate the selection models is chosen without regard to the test data set; hence, we refer to it as “out of domain.” It is synthetic in the sense that document images have been generated from the original digital text and degraded using realistic error models. Along with the true transcripts and OCR hypotheses, the calibration data contains

Published in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011)* [64].



FRANCE: During the period 4th - 15th
continued their operations in the face of v
were frequently supported by tanks, artille
jority of the attacks either the enemy was
or the French Forces of the Interior were a
As a result of their activities, the moveme
plies to NORMANDY has been effectively dela

Figure 5.1: From "Periodical Communiqué No. 1" of the Eisenhower Communiqués. The word error rate of this document across the five OCR engines used in this research varied from 10.63% to 63.41%, with a mean WER of 36.34%.

sufficient information to produce good models of how to select the best OCR hypothesis and thus correct mistaken OCR hypotheses. Maximum entropy methods leverage that information using carefully chosen feature functions to choose the best possible correction. Of the five OCR engines used in this work, our methods show a 24.6% relative improvement over the word error rate (WER) of the best performing OCR engine and a 69.1% relative improvement over the the average WER of all five OCR engines. Our method shows a 24.6% relative improvement over the word error rate (WER) of the best performing of the five OCR engines employed in this work. Relative to the average WER of all five OCR engines, our method yields a 69.1% relative reduction in the error rate. Further, at each step in the progressive alignment together with the maximum entropy-based word selection process, between 39.2% and 52.2% of the documents achieve a new low WER. Furthermore, 52.2% of the documents achieve a new low WER.

5.1 Introduction

In pursuit of high quality digital versions of historical documents, this paper demonstrates the extent to which improvements in the recognized (digital) text are possible as additional OCR hypotheses are incorporated from multiple engines through progressive alignment (cf., [73, 78]). This paper is organized as follows. Section 5.2 discusses related work. Methods used for alignment and the baseline results are in Section 5.3. Section 5.4 discusses the creation of an out-of-domain synthetic data set used to train a maximum entropy model for error correction. Our conclusions are presented in Section 5.5.

5.2 Related Work

There is a significant body of published work on the use of multiple inputs for OCR error correction. Klein and Kopel [49] note that OCR engines show wide variation in the types of errors made and that voting between engines is effective in identifying accurate OCR word hypotheses. Voting among multiple hypotheses has also been explored by Lopresti and Zhou [59], in which multiple scans of the same document were evaluated by the same OCR engine and voting was employed to make the final selection. Lin [56] uses multiple OCR engines to recognize the same document, aligning the OCR text output, with majority voting on the output. Our previous work introduced an efficient exact alignment algorithm [61] and domain-specific training [63] to correct OCR using three engines (two commercial and one open source). Boschetti et al. [10] also align multiple OCR outputs, selecting characters using a naïve Bayes classifier.

In the domain of genetic multiple sequence alignment problems, progressive alignment has been shown to be highly effective in achieving good, although not guaranteed optimal results. (See Moretti et al. [73] and Notredame (2007) [78].) Spencer and Howe [97] apply progressive alignment to textual variants of ancient and historical documents while Feng and Manmatha [27] use a Hidden Markov Model to align the OCR of a full book-length text to an existing electronic version.

A contribution of this paper is the use of supervised, discriminative machine learning methods to choose among all hypotheses. Maximum entropy models have been used previously to select among multiple parses returned by a generative model (e.g. [16]). In this work the models are learned on a synthetic, out-of-domain calibration data set, created and computationally degraded according to the methods proposed by Sarkar, Baird, and Zhang [91] and Baird [7].

5.3 Methods and Results on the Eisenhower Communiqués

5.3.1 Data

The historical documents used in this paper are the Eisenhower Communiqués [43], a collection of 610 facsimiles of typewritten documents created by the Supreme Headquarters Allied Expeditionary

Force (SHAEF) during the last years of World War II. Having been typewritten and duplicated using carbon paper, the quality of the print is poor. (See Figure 5.1 for an example.) A manual transcription of these documents serves as the gold standard for evaluating the word error rates of the OCR. Two-thirds of the documents have been assigned randomly to an evaluation set for this research.[1] One-third of the documents are reserved for future research. We employ no Eisenhower Communiqués data as training data in this work and instead focus on the scenario of recovering document text in the absence of in-domain training data, using an out-of-domain synthetic calibration set.

5.3.2 Baseline OCR

Each of the document images in the Eisenhower Communiqués evaluation set was recognized using five OCR engines: Abbyy FineReader for Windows (version 10), OmniPage Pro X for Mac OS X, Adobe Acrobat Pro for Mac OS X (version 9), ReadIris Pro for Mac OS X (version 11.6), and Tesseract (version 1.03), an open source OCR system. The resulting recognition hypotheses were evaluated using the NIST Sclite [3] tool to compute word error rates (WER) and lattice word error rates (LWER). The baseline WERs for the Eisenhower Communiqués data set can be found in the top half of Table 5.1. These baseline results are a reference point for evaluating the effectiveness of the techniques introduced in this paper. The expectation is that the types of errors as well as the types of successful recognition will vary across engines. Leveraging these variations to correct recognition errors is the goal of this research.

5.3.3 Progressive Alignment

Since exact n-way alignments become exponentially complex in n we turned to greedy progressive alignment heuristics, which are applied successfully in bioinformatics [78] and textual variance analysis [97]. In brief, progressive alignment algorithms begin by selecting two sequences to be aligned that are most similar based on some similarity measure applied to all sequences. Additional

[1] Previous papers [61, 63] using the Eisenhower data set divided the current evaluation set into a training set and a development test set, which accounts for differences in reported development test set WERs.

Table 5.1: Baseline word error rates for the OCR engines on all documents in the evaluation dataset of the Eisenhower Communiqués and the Enron synthetic calibration data set. Note that WERs of greater than 100% are possible due to multiple insertions not found in the reference text.

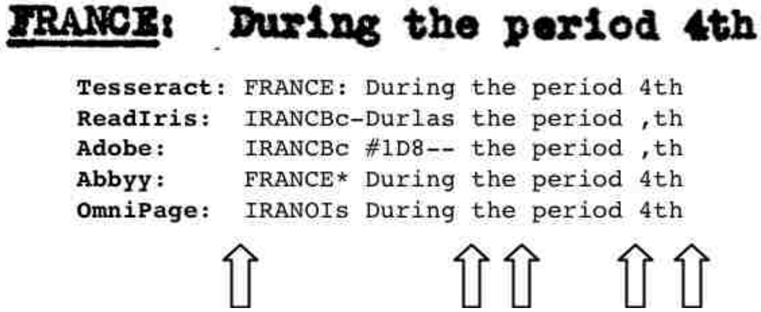
	Word Error Rates				
Eisenhower Communiqués	Abbyy	OmniPage	Adobe	ReadIris	Tesseract
Mean	18.24%	30.02%	51.78%	54.64%	67.78%
	Average WER across all OCR engines: 44.49%				
Minimum	1.87%	1.45%	2.38%	2.38 %	2.01%
Maximum	84.71%	112.68%	151.22%	206.75%	1017.11%
Enron Synthetic Calibration Set	Abbyy	OmniPage	Adobe	ReadIris	Tesseract
Mean	25.02%	31.92%	67.57%	69.62%	56.03%
	Average WER across all OCR engines: 50.03%				
Minimum	0.34%	1.34%	6.02%	5.42 %	4.19%
Maximum	166.34%	205.94%	170.79%	200.00%	176.73%

sequences are aligned, using the same selection criteria as for the first two, until all sequences have been aligned. (Refer to Spencer and Howe [97] for details on progressive alignment in a textual context.) The order of pairwise alignments is specified in a binary tree structure called the guide tree. Due to downstream consequences of greedy choices, a progressive alignment heuristic is not optimal; however, the resulting alignments are good in practice.

In this paper, the order of the alignment, unless indicated otherwise, is a greedy approximation of the guide tree based on sequence similarity of the calibration set (discussed in Section 5.4.1); specifically: Abbyy FineReader and OmniPage Pro X, then Adobe Acrobat Pro and ReadIris Pro, and lastly Tesseract. However, in order to show the effect of adding OCR engines individually, the individual OCR hypotheses were introduced one at a time (progressively) to the overall alignment in a manner consistent with the guide tree.

5.3.4 Lattice Word Error Rates

From the final, overall alignment of the five OCR outputs, we create columns of hypotheses delimited by consensus on white space. Our intention is that each column captures aligned words



FRANCE: During the period 4th

```
Tesseract: FRANCE: During the period 4th
ReadIris:  IRANCBc-Durlas the period ,th
Adobe:     IRANCBc #1D8-- the period ,th
Abbyy:     FRANCE* During the period 4th
OmniPage:  IRANOIs During the period 4th
```

Figure 5.2: From "Periodical Communiqué No. 1", an example of aligned sequence hypotheses from five OCR engines. The arrows indicate points of agreement on white space among the aligned sequences. The text between an adjacent pair of arrows constitutes an aligned column of hypotheses. The hyphen "-" character in a sequence represents a gap aligned with characters in the other sequences.

Table 5.2: Improvements to the Lattice WER for the Eisenhower Communiqués as OCR outputs are progressively added. Row 1: from best to worst. Row 2: from worst to best.

Order by	Number of Aligned OCR Sequences				
OCR WER	1	2	3	4	5
Low to High	18.24%	12.01%	11.28%	11.14%	9.27%
High to Low	67.78%	27.10%	24.08%	16.44 %	9.58%

from the document image. (See Figure 5.2 for an example.) Each aligned column is a list of hypotheses from which to select a single best hypothesis.

The Lattice WER (LWER) is an oracle calculation: for each column, if any of the OCR hypotheses in the column matches the truth in the transcript, it is considered a correct match. As more good hypotheses are added to the lattice, the LWER is reduced, as shown in Table 5.2 and in Figure 5.3. There are several interesting points to observe in the LWER results in row 1 of the table. First, even though Tesseract has the highest overall WER, it still has information to contribute resulting in a decrease in the LWER from 11.14% for a 4-way alignment of Abbyy, OmniPage, Adobe, and ReadIris to 9.27% when Tesseract is added for a 5-way alignment. This result indicates that even higher error rate information sources can potentially contribute to reducing the overall error rate. Second, the order in which the OCR outputs are added to the alignment does little to affect the ultimate Lattice WER when all five engines are included. The small difference between the two ultimate outcomes (shown in the last column of Table 5.2 and the 5-way alignment in

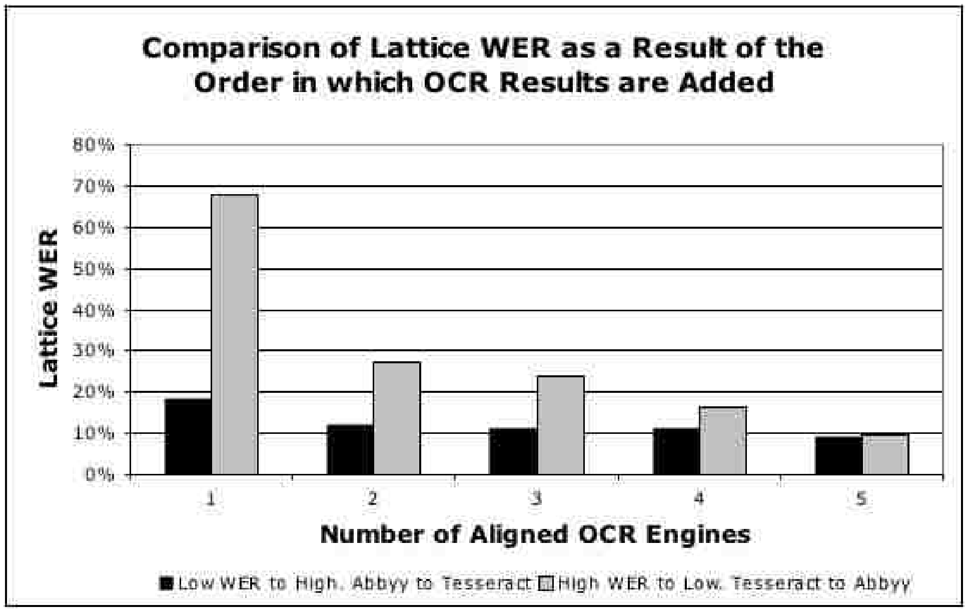


Figure 5.3: Improvements to Lattice WER for the Eisenhower Communiqués

Figure 5.3) can be attributed to the sub-optimalities of the progressive alignment method. The Lattice WER is a lower bound on what is possible given the information contained in the joint alignment of the multiple OCR outputs. Selecting the correct hypotheses within the aligned columns is the remaining task.

5.4 Machine Learning Methods and Results with Out-Of-Domain Training

5.4.1 Enron Synthetically Generated Data Set

Our goal in this research is to explore how well OCR errors can be corrected without requiring a domain-specific training set since such data may be unavailable or too expensive to acquire. In order to use modern discriminative supervised machine learning methods, we used an out-of-domain calibration set. For the calibration set, we created a synthetic data set from the 2001 Topic Annotated Enron Email Data Set [8], a corpus available from the Linguistic Data Consortium (LDC). The choice of the Enron data was essentially arbitrary and reflects our commitment to having a trained model very unfamiliar with our evaluation data. From the digital text of each document, a TIFF document image was generated and randomly degraded using techniques inspired by Baird [7] and Sarkar et al. [91]. Each image was produced in the following way:

1. Create an image from the text as a bi-tonal document at 1500 dots per inch (dpi), which is five times the target resolution of 300 dpi.
2. Introduce spatial sampling error by translating the entire image between one to five pixels in both the x and y axes, which introduces randomness when subsampling.
3. Blur the image using a Gaussian convolution kernel in which the value for each pixel is taken to be the weighted average of its neighboring pixels.
4. Subsample the document to 300 dpi.
5. Simulate pixel sensor sensitivity by adding a value for each pixel individually drawn from a Gaussian distribution with a mean of zero and a standard deviation of 0.025.
6. Choose a random threshold from a truncated normal distribution (between 0.1 and 0.4) with mean 0.225 and a standard deviation of 0.11418, and binarize the document using this threshold.

Note that documents were synthesized with particular parameter ranges chosen independently to reflect the kinds of noise we expected to see in our data, allowing for wide variations. The WERs on the Enron calibration set were comparable to the rates on Eisenhower Communiqués (See Table 5.1) ranging from a mean of 25.02% (Abbyy) to a mean of 69.92% (ReadIris).

5.4.2 Training Maximum Entropy Model Using the Enron Data Set

We employ modern supervised discriminative machine learning methods trained on the calibration set. The role of the machine learning model is to select the proper hypothesis from each aligned column in order to produce the best OCR correction. We prepared training data from the Enron calibration set with the same OCR engines and aligned their output using the same progressive alignment algorithm described above in order to produce aligned columns of hypotheses. We extracted the following kinds of features from each column:

Type of Feature	Feature Values
Word Hypotheses:	T:Precipitation: R:Prccipitalion D:Prcdpitalion A:Precipitation O:Precipitation S:precipitation
Voting	VoteAOS VoteAO VoteAS VoteOS
Dictionary:	DictT DictA DictO
Spell Checker:	Spell
Training Label:	A

Figure 5.4: Example features extracted from a single column of aligned hypotheses.

- *Voting*: multiple features to indicate where multiple hypotheses in a column match exactly,
- *Number*: binary indicators for whether each hypothesis is a cardinal number,
- *Dictionary*: binary indicators for whether each hypothesis appears in the Linux dictionary,
- *Gazetteer*: binary indicators for whether each hypothesis appears in a gazetteer of place names, and
- *Spell Checker*: an additional hypothesis generated by Aspell [5] from words that do not appear in the dictionary or in the gazetteer.

For each training case (an aligned column), the label indicates which OCR engine provided the correct hypothesis. Ties were resolved by selecting the OCR engine with the lowest WER from the calibration set. Consider the following example column from the Enron calibration set: {Abbyy: "Precipitation", OmniPage: "Precipitation", Adobe: "Prcdpitalion", ReadIris: "Prccipitalion", Tesseract: "Precipitation:"}. The Spell Checker also provided the hypothesis: "precipitation". See Figure 5.4 for features are extracted from this column: Note that "DictA" (and so forth) indicates that the entry from each respective OCR engine is found in the dictionary. Leading and trailing punctuation is removed from the hypothesis before checking in the dictionary. To produce a "Voting" feature the match must be exact, including punctuation. During training the label assigned to these features is "A", meaning that Abbyy's is the correct hypothesis to be selected. Once all of the feature vectors have been extracted, we use the maximum entropy learner in the Mallet [72] toolkit

Algorithm 1 recognizeMultipleOCR(d, m, E)

```
INPUT: document: $d$
       model: $m$
       OCR engines set: $E$
$Alignment_d \leftarrow progressiveAlign(d, E)$
$Columns_d \leftarrow splitOnWhitespace(Alignment_d)$
$Transcription_d \leftarrow nil$
for all $c \in Columns_d$ do
  // $c = \{h_a, h_b, h_c, \ldots, h_n\}$
  // $select(c, m) = argmax^*_{h_i \in c} P_m(h_i | features(c))$
  $selection \leftarrow select(c, m)$
  $Transcription_d \leftarrow append(Transcription_d, selection)$
end for
return $Transcription_d$
```

to train a maximum entropy (a.k.a., multinomial logistic regression) model to predict choices on unseen alignment columns.

5.4.3 Machine Learning System Results

The following process is depicted in Algorithm 1. Using the model created with the Enron calibration set, our algorithm assigns a label to each column of hypotheses in each document in the Eisenhower evaluation set. The maximum entropy learner in Mallet indicates which OCR hypothesis to select from the column. The selected hypotheses are then assembled for each document as the corrected output and evaluated by Sclite. It should be noted that Sclite does not distinguish between upper and lower case characters. The function $features()$ in Algorighm 1 returns the features of the aligned column, c, of hypotheses from the OCR engines and $argmax^*()$ performs the expected function while breaking ties as described in Section 5.4.2. The results can be seen in Table 5.3 and in Figure 5.5. From Table 5.3 the lowest WER achieved was 13.76%, which is a 69.1% relative reduction from 44.49%, the mean WER of all OCR engines, and a 24.6% relative reduction from 18.24%, the WER of the best OCR engine (see Table 5.1). Also observe that with each addition of an OCR output, the mean WER on the Eisenhower evaluation set decreases.

Another indication of the ability of the system to take advantage of information provided as new OCR outputs are added is the number of documents that have a reduced WER at each

Table 5.3: WER from the Eisenhower evaluation set for a trained machine learning method to select hypotheses from the aligned lattices as additional OCR outputs are added.

	Word Error Rates				
Method	Abbyy Alone	+OmniPage	+Adobe	+ReadIris	+Tesseract
Machine Learning	18.24%	16.54%	15.09%	14.59%	13.76%

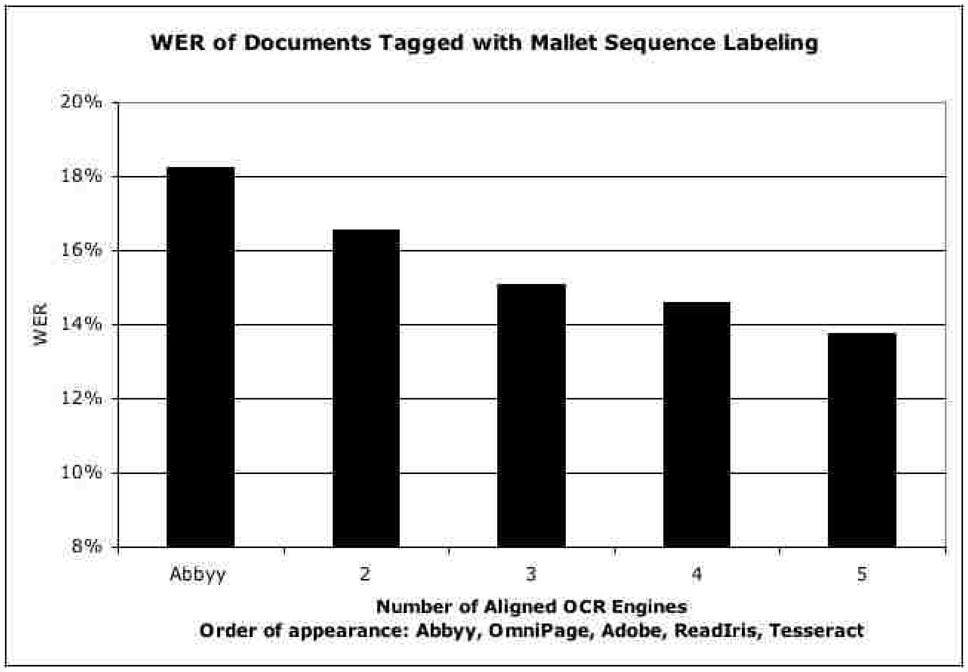


Figure 5.5: Decreasing mean WER on the Eisenhower evaluation set using machine learning methods and an out-of-domain training set. (See the last row of Table 5.3.)

Table 5.4: Percentage of documents reducing WER. Beginning with Abbyy FineReader, as OCR outputs are progressively added to the aligned sequences, at each step there is a significant percentage of documents that reduce their WER either from the previous alignment or as a new overall low WER.

	+OmniPage	+Adobe	+ReadIris	+Tesseract
Reduction from previous alignment	44.16%	83.38%	59.22%	65.19%
New low WER of all previous alignments	44.16%	50.91%	39.22%	52.21 %

progressive step. This is calculated in two ways: first, the percentage of documents that have a lower WER than in the previous step in the progressive alignment, and second, the percentage of documents that achieve a new overall lower WER at that step. Table 5.4 and Figure 5.6 show these results. Note that even after having aligned Abbyy, OmniPage, Adobe, and ReadIris, still 52.21% of the documents have a lower minimum WER with the addition of the Tesseract OCR output, despite Tesseract's 67.78% WER on the Eisenhower evaluation set. We conclude that OCR outputs with even very high WERs can significantly contribute to WER reduction.

5.5 Conclusions

We have documented the degree to which information from multiple OCR engines can be used in an aligned lattice of OCR hypotheses to improve OCR performance. As more OCR engines are included in the alignment, the Lattice WER decreases, even when adding OCR outputs with significant WERs. Thus, progressive alignment provides a usable alternative to exact alignment for processing five OCR sequences. Ultimately, when incorporating multiple OCR engines, the order in which they are added makes little difference on the Lattice WER in the final outcome. Machine learning techniques succeed in leveraging the available information in the lattice: using out-of-domain training data is effective for training a maximum entropy model to select correct hypotheses from the aligned OCR sequences. This research made use of an innovative means for

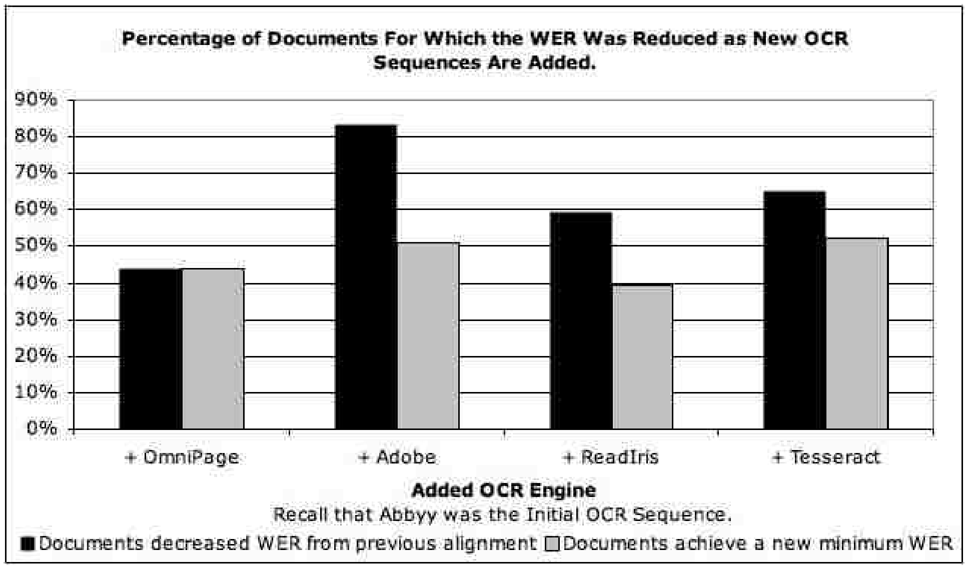


Figure 5.6: As OCR outputs are added to the already aligned sequences, at each step the percentage of documents that reduce their WER, alongside the WER of that OCR engine. (See Table 5.4).

creating a domain-independent calibration training set, which was shown to be successful when used to build models for use with the Eisenhower Communiqués, a historical data set with significant degradation. This work presents a compelling new method for producing digital text from historical documents.

5.6 Addendum

In a recent paper Wernhoener, et al. [110] have an alternative approach to extracting information from multiple sources for OCR error correction by using multiple versions of the same document. For instance, in the digitization of a published work, creating multiple OCR sequences using the same OCR engine, but different editions of the work. Their approach to alignment is similar to ours [64] in using pairwise progressive alignment resulting in hypothesis columns. Voting is used to identify the correct token, but the authors acknowledge that their method breaks down where there is no majority vote.

A thorough introduction to Conditional Random Fields can be found in the tutorial by Sutton and McCallum [100]. Additionally, Conditional Random Fields are introduced in this dissertation in Chapter 2, Section 2.6.2.

Chapter 6

How Well Does Multiple OCR Error Correction Generalize?

Abstract

As the digitization of historical documents, such as newspapers, becomes more common, the need of the archive patron for accurate digital text from those documents increases. Building on our earlier work, the contributions of this paper are: 1. in demonstrating the applicability of novel methods for correcting optical character recognition (OCR) on disparate data sets, including a new synthetic training set, 2. enhancing the correction algorithm with novel features, and 3. assessing the data requirements of the correction learning method. First, we correct errors using conditional random fields (CRF) trained on synthetic training data sets in order to demonstrate the applicability of the methodology to unrelated test sets. Second, we show the strength of lexical features from the training sets on two unrelated test sets, yielding a relative reduction in word error rate on the test sets of 6.52%. New features capture the recurrence of hypothesis tokens and yield an additional relative reduction in WER of 2.30%. Further, we show that only 2.0% of the full training corpus of over 500,000 feature cases is needed to achieve correction results comparable to those using the entire training corpus, effectively reducing both the complexity of the training process and the learned correction model.

Published in *Proceedings of Document Recognition and Retrieval XXI (DRR 2014)* [67].

6.1 Introduction

Historical machine printed document images often exhibit significant noise, making the optical character recognition (OCR) of the text difficult. Our previous work [61, 65] shows that it is possible for combined outputs from multiple OCR engines using machine learning techniques to provide text output with a lower word error rate (WER) than the OCR of any one OCR engine alone. Further, we use methods which are scalable to very large collections, up to millions of images, without document- or test corpus-specific manipulation of training data, which would be infeasible given time and resource constraints.

Ensemble methods are used effectively in a variety of problems such as machine translation, speech recognition, handwriting recognition, and OCR error correction, to name a few. In a paper on pattern recognition frameworks for ensemble methods, Kittler et al. [48] state: ”It had been observed ... that although one of the [classifiers] would yield the best performance, the sets of patterns mis-classified by the different classifiers would not necessarily overlap. This suggested that different classifier designs potentially offered complementary information about the patterns to be classified which could be harnessed to improve the performance of the selected classifier.” Previously we have merged complementary information such as the output of multiple OCR engines [61] and multiple binarizations of the same document image [65] on a single test set and training set. The goal of this paper is to demonstrate the generalizability of the methods involving multiple OCR engines, to introduce a new test and a new training set, to show the results of feature engineering, and to demonstrate the degree to which a large training set may be reduced and still yield results consistent with the full training set.

The remainder of this paper proceeds as follows: Section 6.2 discusses existing work in several fields related to the methods and outcomes of this research. Section 6.3 outlines a brief overview of the methods used to extract corrected text with machine learning techniques, leading to the heart of the paper: the evaluation of the extent to which these methods are applicable across multiple test corpora and synthetic training sets in Section 6.4. Finally, Section 6.5 summarizes the conclusions of this research.

6.2 Related Work

Extracting usable text from older, degraded documents is often unreliable, frequently to the point of being unusable [4]. Kae and Learned-Miller [44] remind us that OCR is not a solved problem and that "the goal of transcribing documents completely and accurately... is still far off." At some point the word error rate of the OCR output inhibits the ability of the user to accomplish useful tasks.

Ensemble methods are used with success in this task as well as in a variety of settings. In 1998 Kitter et al. [48] provided a common theoretical framework for combining classifiers which is the basis for much of the work in ensemble methods. In off-line handwriting recognition Bertolami and Bunke [9] use ensemble methods in the language model. Si et al. [95] use an ensemble of named entity recognizers to improve overall recognition in the bio-medical field. For machine translation, Macherey and Och [69] present a study in how different machine translation systems affect the quality of the machine translation of the ensemble. Maximum entropy models have been used previously to select among multiple parses returned by a generative model [16].

Klein and Kobel [49] as well as Cecotti and Belaïd [14] note that the differences between OCR outputs can be used to advantage. This observation is behind the success of ensemble methods, that multiple systems which are complementary can be leveraged for an improved combined output. The question of how many inputs should be used in an ensemble system is generally "the more the better." Caruana et al. [13] use on the order of 2000 models built by varying the parameters of the training system to create different models. On a smaller number of inputs (five OCR engines), Lund et al. [64] demonstrate that the error rate of the ensemble decreases with each added system. It should be noted that the complementarity [66] of correct responses of the methods is critical. An important point is that even high error rate systems added to an ensemble can contribute to reducing the ensemble error rate where the addition represents new cases or information not included previously. Diversity in the ensemble is critical to improving the system's performance over that of any individual in the ensemble [15, 21, 32, 36]. This paper will expand on the observation regarding complimentary sources, noting that one useful source of diversity is the output of multiple OCR engines.

Necessary for our post-OCR error correction using multiple sequences is an alignment of the text sequences, which can either use exact or approximate algorithms. The multiple sequence alignment problem has been shown to be NP-Hard by Wang and Jiang [109]. Lund and Ringger [61] demonstrate an efficient means for exact alignment; however, alignment problems on long sequences are still computationally intractable. Much of the work in multiple sequence alignment work is done in the field of bioinformatics, where the size of the alignment problems has forced the discovery and adoption of heuristic solutions such as progressive alignment [78]. Elias [23] discusses how a simple edit distance metric, which may be appropriate to text operations, is not directly applicable to biological alignment problems, which means that much of the alignment work in bioinformatics requires some adaptation for use in the case of text.

It is well known from work by Lopresti and Zhou [59] that voting among multiple sequences generated by the same OCR engine can significantly improve OCR. One practical application of voting can be found in the Medical Article Record System (MARS) of the National Library of Medicine (NLM) which uses a voting OCR server for text recognition [101]. Esakov, Lopresti, and Sandberg [25] evaluated recognition errors of OCR systems, and Kolak, Byrne, and Resnik [52] specifically applied their algorithms to OCR systems for post-OCR error correction in natural language processing tasks. OCR error correction with in-domain training [63] as well as out-of-domain training using a synthetic training dataset [64] have been shown to be effective.

Recent work by Yamazoe et al. [114] effectively uses multiple weighted finite-state transducers (WFST) with both the OCR and a lexicon of the target language(s) to resolve the ambiguity inherent in line- and character-segmentation, and character recognition, in which the number of combinations can be very large. Both conventional OCR and post-processing are contained within their system, resolving the difference between various hypotheses before committing to an output string.

A non-ensemble method for improving historical document images prior to OCR is adaptive binarization. Our previous work [65] compared the OCR results of an ensemble of document image binarizations to the results of adaptive binarization. From the perspective of the corpus WER, the

Table 6.1: Baseline corpus word error rates using micro-averaging for datasets by OCR engine. These results are on the original document images without modification. Note that WERs of greater than 100% are possible due to multiple insertions not found in the reference text.

	Abbyy FineReader 10	OmniPage 18	Adobe Pro X	Tesseract 3
Test Set Corpora				
Eisenhower Communiqués	19.98%	30.50%	52.59%	93.99%
	Average WER: 49.26%			
19th Century Mormon Article Newspaper Index	7.44%	11.77%	23.49%	18.35%
	Average WER: 15.26%			
Training Set Corpora				
Enron Synthetic Dataset	24.31%	30.57%	68.95%	56.07%
Reuters-21578 Dataset	15.35%	20.28%	99.77%	82.37%

results of the ensemble methods were superior to those of individual document image adaptive binarizations.

A contribution of this paper is an extension of previous methods in supervised, discriminative machine learning methods to choose among all hypotheses, in which previous methods are shown to be effective in two unrelated historical print corpora and two unrelated training datasets. In this work the models are learned on synthetic, out-of-domain training data sets, created and computationally degraded according to the methods proposed by Sarkar, Baird, and Zhang [91] and Baird [7].

6.3 Methodology

The first step of our methodology prepares the test and training corpora, scanning document images and creating the synthetic training sets. (See Figure 6.1 for a flow chart of this process.) Once the document images have been scanned, the images are recognized with the selected OCR engines; in this case those are Abbyy FineReader 10, OmniPage 18, Adobe Acrobat Pro X, and Tesseract 3 (an open source OCR system). The baseline OCR results of each OCR engine are seen in Table 6.1. The OCR output is character aligned yielding parallel hypotheses from the OCR engines. Where spaces occur in the aligned texts, the process creates a column of text hypotheses. From these columns, features used by the machine learner are extracted as described in Section 6.3.3. The machine

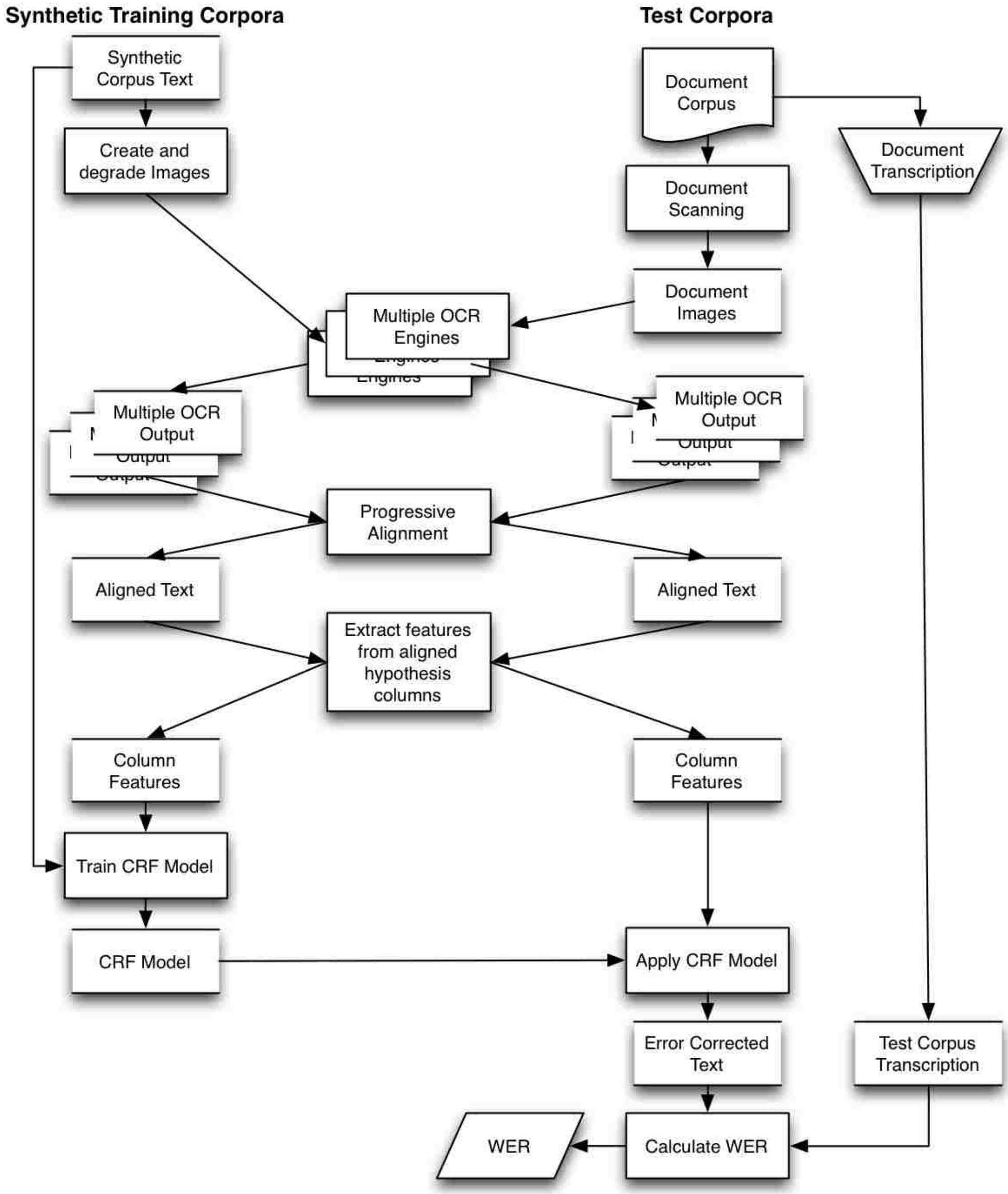
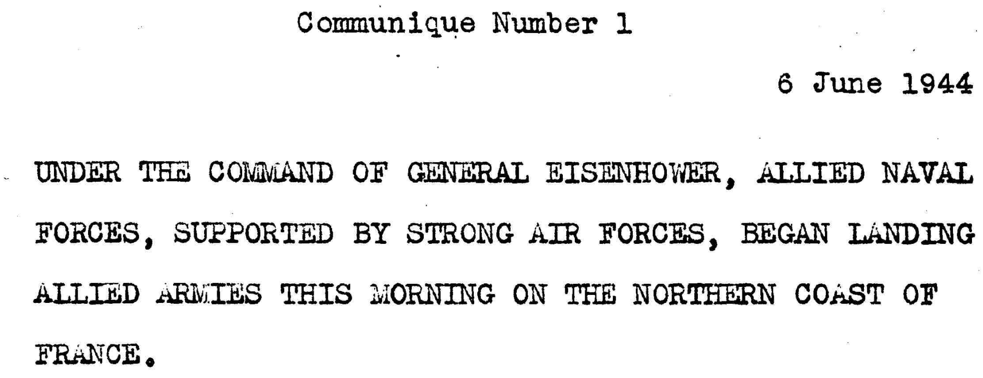


Figure 6.1: The methodology used in this paper to: create the training set and CRF model, prepare the test set for processing by the CRF, and evaluate the results using the test set transcription.



Communique Number 1

6 June 1944

UNDER THE COMMAND OF GENERAL EISENHOWER, ALLIED NAVAL FORCES, SUPPORTED BY STRONG AIR FORCES, BEGAN LANDING ALLIED ARMIES THIS MORNING ON THE NORTHERN COAST OF FRANCE.

Figure 6.2: From "Communiqué No. 1" of the Eisenhower Communiqués. The word error rate of this document across the five OCR engines used in this research varied from 10.63% to 63.41%, with a mean WER of 36.34%.

learner, a trained conditional random field (CRF), labels the aligned hypotheses with the OCR engine to be selected or "NONE", indicating that no output for the column should be selected. The CRF models are trained using the training sets prepared similarly to the test corpora. Collectively, the text associated with each label from all columns constitute the error corrected output. (See Figure 6.4 for an example.) The following sections describe in more detail this process.

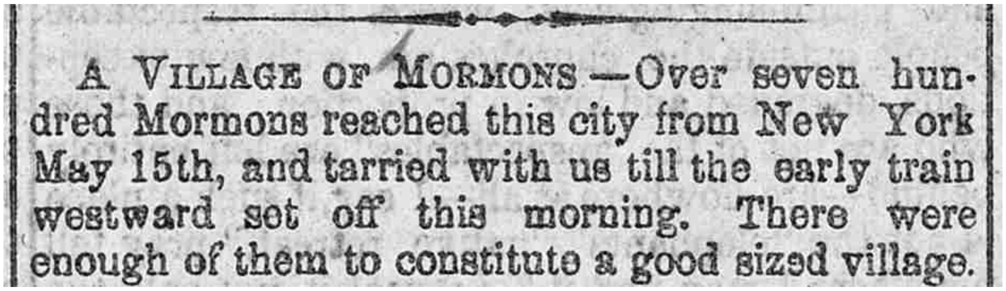
6.3.1 Corpora

Four datasets were used in this work: two test sets, the Eisenhower Communiqués [43] and the Nineteenth Century Mormon Article Newspaper Index [29]; and two training sets, an extraction of the 2001 Topic Annotated Enron Email Data Set and an extraction of the Reuters-21578 Text Categorization Test Collection [55, 108]. The following sections describe each dataset and how it was created.

Eisenhower Communiqués

The Eisenhower Communiqués [43] are a collection of 605 facsimiles[1] of typewritten documents created by the Supreme Headquarters Allied Expeditionary Force (SHAEF) during the last years

[1] An online presentation of The Eisenhower Communiqués is viewable at http://www.lib.byu.edu/digital/eisenhower.



A VILLAGE OF MORMONS.—Over seven hundred Mormons reached this city from New York May 15th, and tarried with us till the early train westward set off this morning. There were enough of them to constitute a good sized village.

Figure 6.3: A fragment from the grayscale scan of document BM_24May1859_p2_c3 from the 19th Century Mormon Newspaper Article collection [29].

of World War II. Having been typewritten and duplicated using carbon paper, the quality of the print is poor. (See Figure 6.2 for an example.) A manual transcription of these documents serves as the gold standard for evaluating the word error rates of the OCR. In the course of duplication the documents have effectively become bi-tonal and are treated as such by this research.

Nineteenth Century Mormon Article Newspaper Index

The Nineteenth Century Mormon Article Newspaper Index [29] (19thCMAN) corpus is a collection of 1055 color images[2] of articles dealing with events and persons of The Church of Jesus Christ of Latter-day Saints (Mormon) from an archive of historical newspapers of the 19th century housed at the Harold B. Lee Library of Brigham Young University. As expected from 19th century newsprint, the quality of the paper and the print was poor when first printed and has further degraded over time. The newspapers were scanned at 400 dots per inch (dpi) in 24-bit RGB color, and the individual articles were segmented and saved as TIFF images. For previous work the RGB images were converted to 8-bit grayscale. The OCR output of each document was manually corrected by two reviewers to act as a gold standard. An example from the document corpus can be seen in Figure 6.3.

Synthetic Training Sets

For the training sets, we created synthetic data sets from the 2001 Topic Annotated Enron Email Data Set [8] and the Reuters-21578 Text Categorization Test Collection [55]. From the digital text

[2] An online presentation of The Nineteenth Century Mormon Article Newspaper Index is viewable at http://lib.byu.edu/digital/19cMormonArticles.

of each document in the test corpora, a TIFF document image was generated and randomly degraded using techniques inspired by Baird [7] and Sarkar et al. [91]. Each synthetic document image was produced using the following steps. First an image is rendered as a bi-tonal document. Spatial sampling error is introduced by translating the entire image stochastically. The image is blurred using a Gaussian convolution kernel. The document is sub-sampled. Gaussian noise is added to simulate pixel sensor sensitivity. To binarize a document, a threshold is applied. For further details on the process, please consult the paper by Walker, Lund, and Ringger (2013) [108].

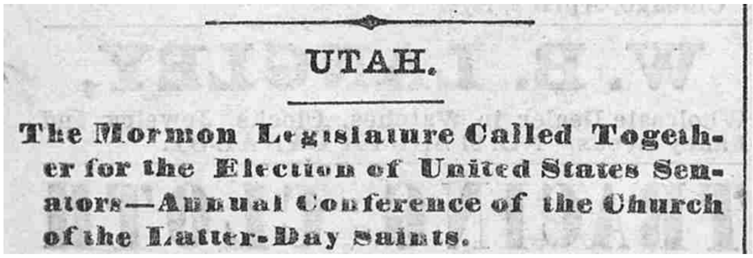
6.3.2 Progressive Alignment

Since exact n-way alignments become exponentially complex in n we turned to greedy progressive alignment heuristics, which are applied successfully in bioinformatics [78] and textual variance analysis [97]. In brief, progressive alignment algorithms begin by selecting two sequences to be aligned that are most similar based on some similarity measure applied to all sequences. Additional sequences are aligned, using the same selection criteria as for the first two, until all sequences have been aligned. (Refer to Spencer et al. [97] for details on progressive alignment in a textual context.) The order of pairwise alignments is specified in a binary tree structure called the guide tree. Due to downstream consequences of greedy choices, a progressive alignment heuristic is not optimal; however, the resulting alignments are good in practice.

In this paper, the order of the alignment, unless indicated otherwise, is a greedy approximation of the guide tree based on sequence similarity of the training set; specifically in the order: Abbyy FineReader and OmniPage Pro X, then Adobe Acrobat Pro, and lastly Tesseract. The incremental results of this alignment order on the WER can be seen in Table 6.4.

6.3.3 Assigning Features in an Alignment Column

We employ modern supervised discriminative machine learning methods trained on the training set. The role of the machine learning model is to select the proper hypothesis from each aligned column in order to produce the best OCR correction. We prepared training data from the training sets with



UTAH.

The Mormon Legislature Called Together for the Election of United States Senators—Annual Conference of the Church of the Latter-Day Saints.

OCR Engine	Hypothesis Columns (Aligned Text)				
Abbyy (A)	Xfee	M-o-r-mon	!.<	-gittiamre	-Called
OmniPage (O)	The-	M-orrison	E-r--gistature		-Called
Tesseract (T)	The-	M-o-r-mon	L-r--gnslauure		-Called
Adobe (D)	The-	lUo-rnton	Lt-˜˜&aslature		()ailed
Features	A:Xfee O:The T:The D:The DictD DictT DictO VoteDOT VoteDO VoteDT VoteOT	A:Mormon O:Morrison T:Mormon D:lUornton DictT DictA DictO VoteAT	A:!.< O:Ergistature T:Lrgnslauure D:Lt-˜˜&aslature	A:gittiamre O: T: D: VoteDOT VoteDO VoteDT VoteOT	A:Called O:Called T:Called D:()ailed DictT DictA DictO VoteAOT VoteAO VoteAT VoteOT
Selected Label	*OmniPage*	*Abbyy*	NONE	*OmniPage*	*Abbyy*
Error Corrected Text	The	Mormon			Called
Transcript	The-	M-o-r-mon	L-e--gislature		-Called

Figure 6.4: An example of an aligned lattice from document CT_2Apr1872_p2_c1. The "dash" character represents a gap or *INDEL* in the alignment, where a character needs to be inserted in order to complete the alignment. Correct hypotheses in the aligned text are underlined. The aligned text is divided into columns of hypotheses on spaces in the aligned sequences. Note that a space occurred in the middle of the Abbyy mis-recognition of the word "Legislature".

the same OCR engines and aligned their output using the same progressive alignment algorithm described above in order to produce aligned columns of hypotheses. (See Figure 6.4.) As a base we extracted the following kinds of feature types from each column:

- *Voting*: multiple features to indicate where multiple hypotheses in a column match exactly.
- *Number*: binary indicators for whether each hypothesis is a cardinal number.
- *Dictionary*: binary indicators for whether each hypothesis appears in the Linux dictionary.
- *Gazetteer*: binary indicators for whether each hypothesis appears in a gazetteer of place names.
- *Lexical Features*: words that appear in the corpus are individually created as features.

For each training case (an aligned column), the label indicates which OCR engine provided the correct hypothesis. Ties were resolved by selecting the OCR engine with the lowest WER from the training set. Note that "DictA" (and so forth) indicates that the entry from Abbyy is found in the dictionary. Leading and trailing punctuation is removed from the hypothesis before checking in the dictionary. To produce a "Voting" feature type the match must be exact, including punctuation. Once all of the feature vectors have been extracted, we use the maximum entropy learner in the Mallet [72] toolkit to train a maximum entropy (a.k.a., multinomial logistic regression) model to predict choices on unseen alignment columns.

6.4 Results

Previous work [61, 63, 64] using the Eisenhower Communiqués test set and the Enron training set was focused on individual document performance in which corpus WERs were calculated as the average of the individual document WERs. A result of this method is that corpus statistics would give more weight to the the tokens of a short document than to the tokens of a long document. This approach may be called a "macro-average" in which each document is given an equal weight, regardless of its size. In contrast, the results reported in this paper are based on tokens, giving an equal weight to each occurrence of a token in the corpus, the OCR output, and the resulting error corrections. All averages and other statistics in this paper, unless stated otherwise, use this "micro-averaging" approach. We believe this approach is more suited for feature engineering although there are important uses of a document-by-document evaluation. Examples where a document-by-document analysis is useful would be observing improvement trends by document or in error analysis related to document WER.

Both for documents and for the corpus as a whole, the word error rate is calculated as

$$WordErrorRate = \frac{Substitutions + Insertions + Deletions}{Number\ Of\ Reference\ Tokens} \tag{6.1}$$

which may be greater than 100% due to the number of insertions, which is not limited.

The remainder of this section is organized as follows. Section 6.4.1 shows the baseline results from previous work reinterpreted using the micro-averaging approach discussed above. New features for this work, recurring features and an order 1 CRF, are described in Section 6.4.2 with the results of incorporating these features in Section 6.4.3. The generalization of these methods on a new test corpus and a new training corpus is shown in Sections 6.4.4 and 6.4.5. Section 6.4.6 wraps up with an evaluation of the effect of the training corpus size on the results of both test corpora.

6.4.1 Baseline Results

As a baseline consider the WERs of the OCR output on the unmodified images of the documents from the Eisenhower Communiqués test set and the Enron training set. Each document image in the Eisenhower Communiqués test set, as well as all corpora in this work, was recognized using four OCR engines: Abbyy FineReader 10, OmniPage 18, Adobe Acrobat Pro X, and Tesseract 3. The resulting recognition hypotheses were evaluated using the NIST Sclite [3] tool to compute the number of correctly recognized tokens, as well as substitutions, insertions, and deletions for each document. These results, shown in the first numerical column of Table 6.4, are a reference point for evaluating the effectiveness of the new techniques and corpora introduced in this paper.

Our previous work [63, 64] showed the improvement in the WER using a trained machine learner with the alignment of the output from multiple OCR engines. Reinterpreting these previous results using micro-averaging techniques, the underlined entries in Table 6.4 show the decreasing WER as additional OCR outputs are added to the alignment. The order of the OCR output alignment was determined by the order of increasing WER on the Enron training set.

6.4.2 New Features

In addition to the baseline feature types described in Section 6.3.3, this paper adds three new feature types for consideration by the machine learner: voting, dictionary lookup, gazetteer lookup, identifying numbers, and the lexical features of the training set. This paper adds three new features:

1. *RecurSim*, a binary feature indicating whether a token occurs more than once.
2. *RecurBucket#*, a multivalued feature dividing the number of times a token appears into one of ten buckets, numbered 0 to 9, with each bucket containing approximately the same number of tokens. The higher the bucket number, the fewer times the individual tokens in that bucket appeared in the corpus. (See Section 6.4.2 for more details.)
3. *Order*, the machine learner is trained using either an order 0 or an order 1 conditional random field (CRF). The order 0 CRF only considers the current column of hypotheses when deciding on the label. The order 1 CRF considers the previous label in addition to the current column features.

Recurring Features

The recurring features (RecurSim and RecurBucket#) are calculated on the training and test sets since the contents of the OCR outputs is available without violating a restriction on using the gold standard transcription of the documents in the corpus. The simple recurring feature (RecurSim) is created for every token that appears more than once anywhere in the corpus. The hope is that by tracking recurring features in the corpus out of vocabulary tokens that do not appear in the dictionary or gazetteer will be captured.

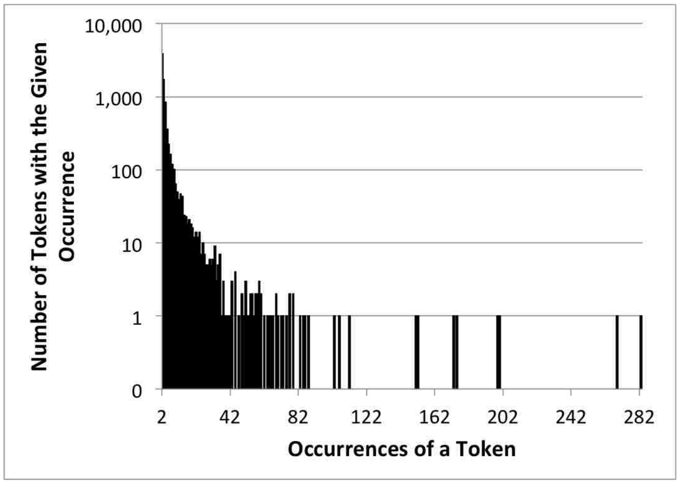


Figure 6.5: A histogram from the Eisenhower test set showing the number of times that a token with a given number of occurrences appears in the OCR of the test corpus. For example there are 3,931 tokens that appear twice in the corpus and one token that appears 283 times.

Table 6.2: Assignments of token recurrences in the Eisenhower test set to feature RecurBucket#.

	Recurring Token Count	
Bucket Label	from	to
RecurBucket0	2	3
RecurBucket1	4	12
RecurBucket2	5	21
RecurBucket3	22	30
RecurBucket4	31	41
RecurBucket5	42	52
RecurBucket6	53	62
RecurBucket7	63	76
RecurBucket8	77	150
RecurBucket9	151	283

The bucket recurring feature (RecurBucket#) divides up the range of recurring feature counts into ten buckets tagged with the labels RecurBucket0 to RecurBucket9. The method for assigning bucket labels calculates the number of times a given token appears in the OCR of the corpus. For example in the combined OCR of the Eisenhower test corpus there are 3,931 different tokens that each appear twice, one of which is "SCHEUERN" and is likely a valid recognition by the OCR engines of a town by that name in Germany. (See Figure 6.5.) The buckets are assigned labels in an ascending order such that there are approximately the same number of recurring tokens in each bucket. For the Eisenhower test set the bucket assignments are found in Table 6.2. The simpler "RecurSim" feature is assigned to all tokens that recur within the corpus, so RecurBucket0 through RecurBucket9 would all be mapped to RecurSim.

CRF Order

Previously, the machine learner used an order 0 conditional random field (CRF), also called a log-linear classifier. This means that only the features of the current column are used to select the label assigned to the features from the hypothesis column. The selected label, which is one of the OCR engines or the label "NONE", determines which OCR hypothesis to select or whether to select none of the OCR outputs.

For this paper we have added a new set of models, trained using the same training sets but modeled using an order 1 CRF. The model considers not only the features found in the current hypothesis column, but also the label that was selected previously. The results will clearly identify whether the order 0 or the order 1 CRF model is being used.

6.4.3 Results of the New Features

One of the goals of this research was to determine the contribution of the lexical features learned from the training set to the overall performance of the machine learner. To this end we group features into sets, including and excluding both the RecurSim, RecurBucket#, and the Lexical feature types.

Table 6.3: The grouping of features used in various model configurations.

Feature Set	Feature Types
Base Set	Voting, Dictionary, Gazetteer, and Number
RecurSim Set	All features found in *Base Set* along with the RecurSim feature type
RecurBucket# Set	All features found in the *Base Set* along with the RecurBucket# feature types
Lexical Set	All features found in the *Base Set* with the Lexical feature type
Lexical+RecurSim Set	A combination of the *Base Set*, *Lexical Set*, and the RecurSim feature type
Lexical+RecurBucket# Set	A combination of the *Base Set*, *Lexical Set*, and the RecurBucket# feature types.

The set names and features found in each set are found in Table 6.3. Refer to Section 6.3.3 for an explanation of the feature names.

The results across all feature set groupings and CRF orders may be seen in Table 6.4. To orient the reader, the previously published results are underlined in the "Order 0 CRF" section of the table. Note the italicized entries which indicate improvements[3] within a given OCR alignment over the previous results.

Observe that the order 1 CRF is not in general an improvement over the order 0 CRF. With the exception of results using the Abbyy+OmniPage alignment all of the other results in the table have an increased WER. Further, the best result of 17.42% in the order 1 CRF is the Abbyy+OmniPage+Tesseract+Adobe alignment with the *Lexical+RecurBucket# Set* is significantly higher than the best result in the order 0 CRF table at 16.23%. Based on this, the order 1 CRF will not be included in the results the follow since it is not showing an improvement over the order 0 CRF.

Clearly the Lexical features, as found in the three feature sets *Lexical Set*, *Lexical+RecurSim Set*, and *Lexical+RecurBucket# Set*, show improvement over the feature sets without the Lexical features, yielding a 6.52% relative improvement between the *Base Set* of the Abbyy+OmniPage+Tesseract+Adobe

[3] The ground truth transcription of the Eisenhower test set consists of 145,346 words in 605 documents. A WER reduction of 0.01% constitutes 15 tokens that are corrected across the corpus, consisting of insertions that are eliminated or words that are corrected.

Table 6.4: Eisenhower test set WERs with the Enron training set including the new features. The underlined entries correspond to the results previously published. Italicized entries indicate improvement over previous results. The bolded entry is the lowest WER in the table.

						Lexical
					Lexical	+ Recur-
Alignment Order	Base	RecurSim	RecurBucket#	Lexical	+ RecurSim	Bucket#
Order 0 CRF	Abbyy OCR WER: 19.98%					
Abbyy + OmniPage	21.91%	22.20%	22.15%	18.31%	*18.13%*	*18.19%*
Abbyy + OmniPage + Tesseract	*17.47%*	*17.15%*	*17.47%*	17.52%	*17.29%*	*17.45%*
Abbyy + OmniPage + Tesseract + Adobe	17.80%	17.55%	17.78%	16.64%	***16.23%***	*16.46%*
Order 1 CRF						
Abbyy + OmniPage	21.56%	21.76%	21.68%	*17.70%*	*17.61%*	*17.70%*
Abbyy + OmniPage + Tesseract	17.68%	17.59%	17.70%	17.80%	17.62%	17.75%
Abbyy + OmniPage + Tesseract + Adobe	17.82%	17.74%	17.79%	17.49%	17.17%	**17.42**%

alignment and the *Lexical Set* as shown in Table 6.4. In addition, the recurring features in conjunction with the Lexical features are superior to the Lexical features alone with the *Lexical+RecurSim Set* having the greatest improvement, showing an additional 2.30% relative improvement over the Abbyy+OmniPage+Tesseract+Adobe alignment and the *Lexical Set* mentioned above. Overall the *Lexical+RecurSim Set* performs best on the Eisenhower test set and the Enron training set. We will proceed with the *Lexical Set* and recurring feature sets as we compare results with the new test corpus, the 19th Century Mormon Article Newspaper Index.

6.4.4 Results on a Different Test Corpus

The 19th Century Mormon Article Newspaper Index, described in Section 6.3.1, consists of 208,630 words[4] in 1,055 documents digitized to 8-bit grayscale, in contrast to the Eisenhower dataset which is effectively bitonal. The results using the selected feature sets from the previous section are found in Table 6.5.

The monotonic improvement in WER seen on the Eisenhower test set using the Lexical+RecurSim feature set is not reflected in the 19thCMAN test set using the Enron training set. In the 19thCMAN test set, the addition of the Tesseract OCR increases the WER above the Abbyy+OmniPage alignment results across the board for all of the feature sets. Unlike the Eisenhower test set, 19thCMAN appears to have a sensitivity to the relatively high WER of the Tesseract OCR (56.08%). Adding the Adobe OCR output improves the resulting WER to a level equal to or below both the Abbyy FineReader WER and the Abbyy+OmniPage alignment, even though the WER of Adobe on the 19thCMAN test set (68.95%) is higher than that of Tesseract. The conclusion here is that although the high WER OCR of Tesseract and Adobe in the training set were able to contribute to lowering the WER for the Eisenhower test set, in combination they did not contribute in the same way with the 19thCMAN treat set. A possible solution may be to eliminate from the training set documents with high WERs. Since the training set includes alignments of documents from multiple OCR engines, this may decrease the size of the training set since if a document is eliminated due

[4] A reduction of 0.01% in the WER on the 19th Century Mormon Article Newspaper Index results in 21 tokens that are corrected across the corpus.

Table 6.5: Results on the Eisenhower and the 19th Century Mormon Article Newspaper Index test sets using a CRF model trained on the Enron, Reuters, and combined training sets. The bold entries are the lowest WERs for each section of the table. The underlined entries are the lowest WERs for their respective test sets.

	Eisenhower Test Set			**19thCMAN Test Set**		
	Feature Sets			**Feature Sets**		
Alignment	**Lexical**	**Lexical +RecurSim**	**Lexical +RecurBucket#**	**Lexical**	**Lexical +RecurSim**	**Lexical +RecurBucket#**
Enron Training Set	Abbyy OCR WER: 19.98%			Abbyy OCR WER: 7.44%		
Abbyy+OmniPage	18.31%	18.13%	18.19%	6.91%	7.03%	6.92%
Abbyy+OmniPage +Tesseract	17.52%	17.29%	17.45%	7.06%	7.75%	7.46%
Abbyy+OmniPage +Tesseract+Adobe	16.64%	**16.23%**	16.49%	**6.83%**	7.03%	6.90%
Reuters-21578 Training Set						
Abbyy+OmniPage	19.91%	20.14%	19.66%	5.99%	5.99%	**5.98%**
Abbyy+OmniPage +Tesseract	16.29%	16.12%	**15.97%**	6.80%	7.64%	7.16%
Abbyy+OmniPage +Tesseract+Adobe	16.41%	16.71%	16.37%	6.99%	7.68%	7.42%
Combined Training Set						
Abbyy+OmniPage	20.01%	20.02%	19.83%	6.04%	**5.97%**	5.99%
Abbyy+OmniPage +Tesseract	16.88%	17.15%	16.96%	6.82%	6.82%	7.28%
Abbyy+OmniPage +Tesseract+Adobe	16.63%	**16.56%**	16.63%	6.83%	6.82%	7.05%

to a high WER from one OCR engine, it would need to be eliminated from all of the OCR engine contributions to the training set. Section 6.4.6 explores whether the full contents of the training set are needed to maintain the level of performance seen so far.

6.4.5 Results on New Training Corpora

New in this paper, the Reuters-21578 training set described in Section 6.3.1 is a synthetic dataset consisting of grayscale images, which is in contrast to the Enron training set which had previously be binarized. The hope was that since the 19thCMAN test set was grayscale, that the Reuters-21578 training set would contribute to improving the over WER. Note that the affects of the high WER OCR outputs from Tesseract and Adobe Pro X seem more pronounced with this training set. The WER results for Abbyy+OmniPage were the best and further adding Tesseract and Adobe Pro X to the alignment each increased the WER. Overall, however, the Reuters-21578 training set showed better results than the Enron training set.

The last rows of Table 6.5 show the results of merging both the Enron and the Reuters-21578 training sets. As more training data is available, there is no improvement on the Eisenhower test set and a small improvement of only 0.01% for the 19thCMAN test set. The conclusion is that there is a point where more training set vectors does not necessarily improve the outcome. Overall the best results were seen with the Reuters-21578 training set, although not consistently with the complete set of OCR alignments.

6.4.6 Results of Sweeping the Size of the Training Sets

Exploring the observation from the last section, that the increased size of the training set does not necessarily improve the WER outcome, we sweep the size of the Reuters-21578 training set from 0.01% to 100% to explore how the WER of the Eisenhower and 19thCMAN test sets varies as the training set increases in size. We selected the best result on the Eisenhower test set across all training sets, which was the Abbyy+OmniPage+Tesseract alignment using the *Lexical+RecurBucket# Set* as seen in Table 6.5. We selected ten proportion values (0.01%, 0.1%, 0.2%, 0.5%, 1%, 2%, 5%, 10%,

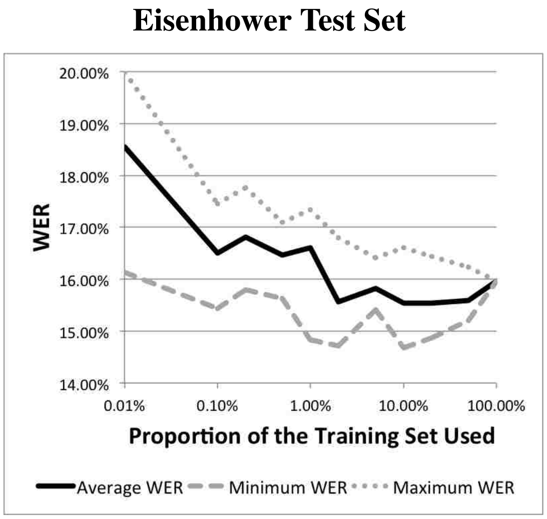


Figure 6.6: Comparing the WER on the Eisenhower test set, with the Reuters-21578 training set, as the proportion of the training set varies from 0.01% to 100%.

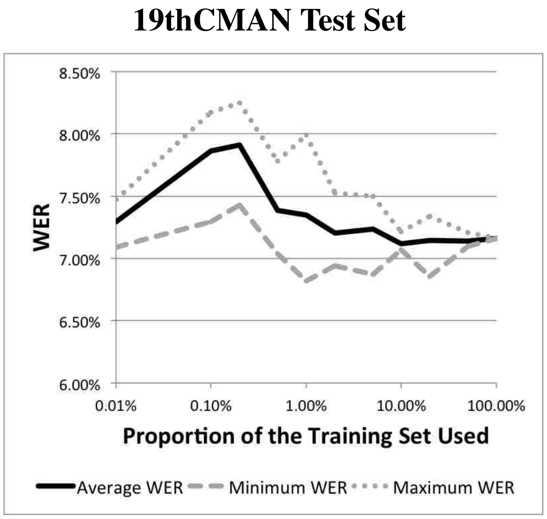


Figure 6.7: Comparing the WER on the 19thCMAN test set, with the Reuters-21578 training set, as the proportion of the training set varies from 0.01% to 100%.

20%, and 50%) of the training set, which has 585,291 feature vectors. At each proportion value we took five random samples from the full training set to create 50 new training sets from which order 0 CRF models were created. The results shown in Figures 6.6 and 6.7 are the average WER, as well as the minimum and maximum WERs for the five models of each proportion value. When selecting a proportion size of only 0.01% of the total training set, only 61 feature vectors on average are included in the models created.

Of interest on the Eisenhower results is that beginning with a training set proportion size of only 2.0% the average WER of the resulting error corrected output is less than the WER using the entire test corpus. Given that the full test set takes a significant amount of time to train, the five 2.0% test sets are considerably faster to train. Further the models created from the 2.0% test sets consist of on the order of 11,500 features while the full model consists of over 217,000 features. Clearly the complexity of the full model does not necessarily reward us with better results.

Regarding the 19thCMAN test set, similar to the Eisenhower test set, the full effect of the Reuters-21578 training set is visible between 1% and 10% of the total training set. Interestingly, superior results are possible given individual training set proportions within the same range.

6.5 Conclusions

In general the results seen in previous work are also seen with the new test set and training set. Although the methodologies used in previous papers still produced good results they do not work consistently across the board. The 19thCMAN test set seemed sensitive to the high WER OCR engines included in the training set, but we demonstrated that the size of the training set can be reduced, potentially eliminating the troublesome high WER documents, potentially improving the end results. Future work will include error analysis of how the models of the Enron and Reuters-21578 training sets differ in their performance on the Eisenhower and 19thCMAN test sets.

6.6 Addendum

A thorough introduction to Conditional Random Fields can be found in the tutorial by Sutton and McCallum [100]. Additionally, Conditional Random Fields are introduced in this dissertation in Chapter 2, Section 2.6.2.

Chapter 7

Combining Multiple Thresholding Binarization Values to Improve OCR Output

Abstract

For noisy, historical documents, a high optical character recognition (OCR) word error rate (WER) can render the OCR text unusable. Since image binarization is often the method used to identify foreground pixels, a significant body of research has sought to improve image-wide binarization directly. Instead of relying on any one imperfect binarization technique, our method incorporates information from multiple global threshold binarizations of the same image to improve text output. Using a new corpus of 19th century newspaper grayscale images for which the text transcription is known, we observe WERs of 13.8% and higher using current binarization techniques and a state-of-the-art OCR engine. Our novel approach combines the OCR outputs from multiple thresholded images by aligning the text output and producing a lattice of word alternatives from which a lattice word error rate (LWER) is calculated. Our results show a LWER of 7.6% when aligning two threshold images down to a LWER of 6.8% when aligning five. From the word lattice we commit to one hypothesis by applying the methods of Lund et al. (2011) achieving 8.41% WER, a 39.1% reduction in error rate relative to the performance of the original OCR engine on this data set.

Published in *Proceedings of Document Recognition and Retrieval XX (DRR 2013)* [65].

7.1 Introduction

Optical Character Recognition (OCR) has reached a point where well-formatted and clean documents are easily recognizable by current commercial OCR products; however, older or degraded documents present problems with word error rates (WER) that severely limit either automated or manual use of the text. Kolak et al. [52] and others have proposed correcting OCR output, aiming for an accurate transcription, as a way to make digitized documents more accessible to natural language processing (NLP) tasks. Many collections of historical printed documents can number into the millions of items, making such manual intervention infeasible. The goal of the research presented here is to improve the quality of OCR output without human intervention on large corpora where manual correction is not possible.

This paper presents new research and a novel approach to extracting text from document images with the goal of directly improving the OCR output, measured by the WER. Our method does not rely on any one imperfect adaptive binarization but incorporates information from multiple global threshold binarizations to improve the final text output.

The remainder of this paper is structured as follows: section 7.2 discusses related work in binarization, optical character recognition, text alignment, and ensemble methods for error correction; the methods and corpus used for this research are covered in Section 7.3; results from our methods are shown in Section 7.4; and conclusions are found in Section 7.5.

7.2 Related Work

Areas in which there is related work are optical character recognition, image binarization, text alignment, and ensemble methods for error correction.

7.2.1 Optical Character Recognition

Printing and duplication techniques of the 19th and mid-20th centuries create significant problems for OCR engines. Examples of problematic documents include typewritten text, in which letters are

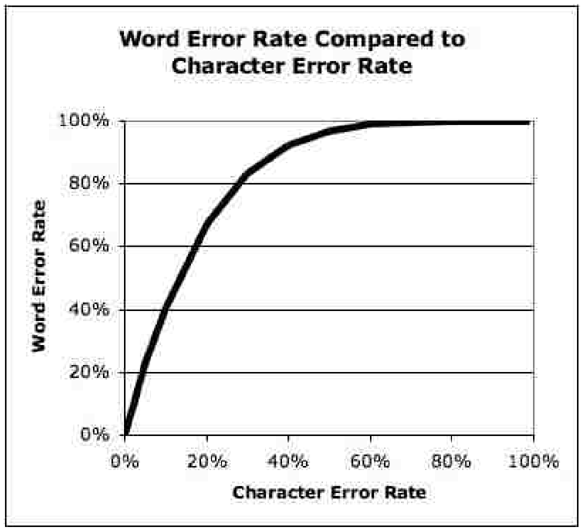


Figure 7.1: Comparing character to word error rates, assuming an average word length of 5 characters.

partially formed, typed over, or overlapping; documents duplicated by mimeographing, carbon paper, or multiple iterations of photographic copying common in the mid-20th century; and newsprint which uses papers that are acidic and type that can exhibit incomplete characters. In addition to original documents which may exhibit problematic text, newspapers may suffer degradation such as bleed-through of type and images, damage due to water, and discoloring of the paper itself. (See Figures 7.2, 7.3, and 7.5 for examples of grayscale images from 19th century newspapers.)

Extracting usable text from older, degraded documents is often unreliable, frequently to the point of being unusable [4]. Even in situations where a fairly low character error rate is achieved, Hull [38] points out that a 1.4% character error rate results in a 7% word error rate on a typical page of 2,500 characters and 500 words (see Figure 7.1).

Kae and Learned-Miller [44] remind us that OCR is not a solved problem and that “the goal of transcribing documents completely and accurately... is still far off.” At some point the word error rate of the OCR output inhibits the ability of the user to accomplish useful tasks. Munteanu et al. [74] determined that in an automated speech recognition task creating text transcriptions of lectures, a transcript with a WER of 50% was no better than having no transcript at all.

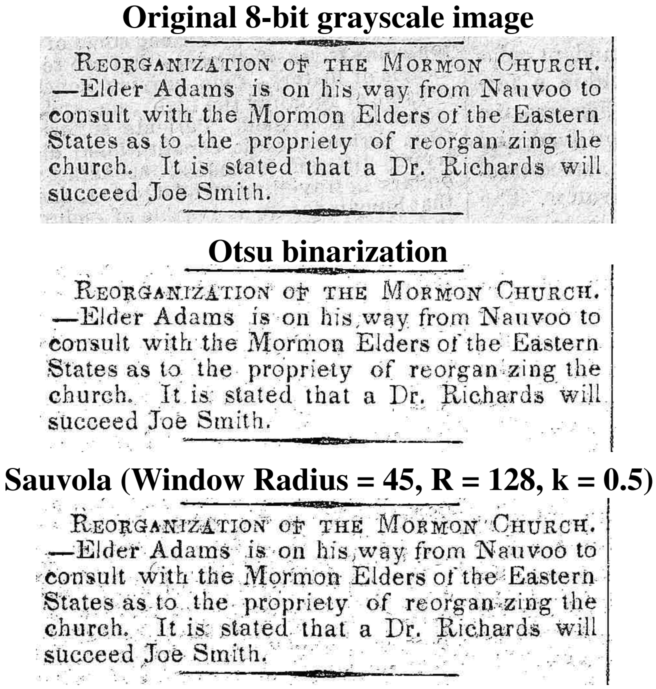
7.2.2 Binarization

Image binarization methods create bitonal (black and white) versions of images in which black pixels are considered to be the foreground (characters or ink) and white pixels are the document background. The simplest form of binarization is *global thresholding*, in which a grayscale intensity threshold is selected and then each pixel is set to either black or white depending on whether it is darker or lighter than the threshold, respectively.

Since the brightness and contrast of document images can vary widely, it is often not possible to select a single threshold that is suitable for an entire collection of images. The Otsu method [79] is commonly used to automatically determine thresholds on a per-image basis. The method assumes two classes of pixels (foreground and background) and uses the histogram of grayscale values in the image to choose the threshold that maximizes between-class variance and minimizes within-class variance. (See Figure 7.2 for an example.) This statistically optimal solution may or may not be the best threshold for OCR, but often works well for clean documents.

For some images, no global (image-wide) threshold exists that results in good binarization. Background noise, stray marks, or ink bleed-through from the back side of a page may be darker than some of the desired text. Stains, uneven brightness, paper degradation, or faded print can mean that some parts of the page are too light for a given threshold while other parts are too dark for the same threshold.

Adaptive thresholding methods attempt to compensate for inconsistent brightness and contrast in images by selecting a threshold for each pixel based on the properties of a small portion of the image (*window*) surrounding that pixel, instead of the whole image. The Sauvola [92] method is a well-known adaptive thresholding method. (For examples see Figure 7.2.) Sauvola performs better than the Otsu method in some cases; however, neither is better in all cases, and in some cases adaptive thresholding methods even accentuate noise more than global thresholding. In addition, the results of the Sauvola method on any given document are dependent on user-tunable parameters. Like global thresholds, a specific parameter setting may not be sufficient for good results across an entire set of documents.



Original 8-bit grayscale image

REORGANIZATION OF THE MORMON CHURCH. —Elder Adams is on his way from Nauvoo to consult with the Mormon Elders of the Eastern States as to the propriety of reorgan zing the church. It is stated that a Dr. Richards will succeed Joe Smith.

Otsu binarization

REORGANIZATION OF THE MORMON CHURCH. —Elder Adams is on his way from Nauvoo to consult with the Mormon Elders of the Eastern States as to the propriety of reorgan zing the church. It is stated that a Dr. Richards will succeed Joe Smith.

Sauvola (Window Radius = 45, R = 128, k = 0.5)

REORGANIZATION OF THE MORMON CHURCH. —Elder Adams is on his way from Nauvoo to consult with the Mormon Elders of the Eastern States as to the propriety of reorgan zing the church. It is stated that a Dr. Richards will succeed Joe Smith.

Figure 7.2: Original grayscale image and two adaptive binarizations of the document AEM_24-Jul1844_p3_col1 from the 19th Century Mormon Newspaper Article collection [29]

Although the Otsu and Sauvola methods are well-known and widely-used binarization methods, a large body of research exists for binarization in general [94] and also specifically for binarization of document images (for example, [30, 31, 47, 115, 118]). While various methods perform well in many situations, no method works infallibly under all circumstances.

7.2.3 Alignment

Implicit in any post-OCR error correction using multiple sequence sources is an alignment of the text sequences, which can either use exact or potentially suboptimal algorithms. The multiple sequence alignment problem has been shown to be NP-Hard in the paper on multiple sequence alignment complexity by Wang and Jiang [109]. Lund and Ringger [61] demonstrate an efficient means for exact alignment; however, large-scale alignment problems are still computationally intractable. The vast majority of multiple sequence alignment work is done in the field of bioinformatics, where the size of the alignment problems has forced the discovery and adoption of heuristic solutions such as progressive alignment. Elias [23] discusses how a simple edit distance metric, which may be appropriate to text operations, is not directly applicable to biological alignment problems, which means that much of the alignment work in bioinformatics requires some adaptation for use in the text sequence case.

Ikeda and Imai [39] and Schroedl [93] represent alignment problems as minimum distance problems in a directed acyclic graph in which edge costs represent the cost of alignment decisions. Further, Ikeda & Imai and Schroedl formulated the cost of DNA base (or character) level alignments in n-dimensions as the sum of the costs of 2-dimensional alignments. Notredame [77] states: "Computing exact [multiple sequence alignments] is computationally almost impossible, and in practice approximate algorithms (heuristics) are used to align sequences, by maximizing their similarity."

7.2.4 Ensemble Methods for Error Correction

Ensemble methods are used effectively in a variety of problems such as machine translation, speech recognition, handwriting recognition, and OCR error correction, to name a few. In a paper on the framework for ensemble methods, Kittler, et al. [48] state: "It had been observed ... that although one of the [classifiers] would yield the best performance, the sets of patterns mis-classified by the different classifiers would not necessarily overlap. This suggested that different classifier designs potentially offered complementary information about the patterns to be classified which could be harnessed to improve the performance of the selected classifier." Cecotti and Belaïd [14] note that combining multiple classifiers has been shown to outperform individual classifiers, but only when they complement each other. This observation is behind the success of ensemble methods, that multiple systems which are complementary can be leveraged for an improved combined output. It should be noted that the complementarity of correct responses of the methods is critical.

Voting among multiple outputs is a common combination method used in OCR research. Lopresti and Zhou [59] use voting among multiple sequences generated by the same OCR engine from different scans of the same document. Their experimental results showed between 20% and 50% reduction in errors. Our method differs from that of Lopresti and Zhou in that although we start with the same digital image, rather than adjusting the scans of the document, we adjust the binariztion threshold. A recent work by Yamazoe, Etoh, Yoshimura, and Tsujino (2011) [114] effectively uses multiple weighted finite-state transducers (WFST) with both the OCR and a lexicon of the target language(s) to resolve the ambiguity inherent in line and character segmentation, and character recognition, in which the number of combinations can be very large. Both conventional OCR and post-processing are contained within their system resolving the difference between various hypotheses before committing to an output string.

7.3 Method

The methods used in this research include image thresholding as introduced in Section 7.2.2, multiple OCR text sequence alignment, hypothesis lattice construction, and the evaluation of error rates.

7.3.1 Document Corpus

The historical documents used in this research are from a collection of 1074 images from the 19th Century Mormon Article Newspaper (19thCMNA) index [29] from Brigham Young University's L. Tom Perry Special Collections of the Harold B. Lee Library. These documents are a collection "of newspaper articles which deal in some way with the Church of Jesus Christ of Latter-day Saints, the Mormons, or with the territory or state of Utah found in national newspapers between 1831 [and] 1900." The quality of the paper and of the print was poor when first printed and has degraded over the years in yellowing of the paper and damage from various sources. The original newspaper pages are scanned at 400 dots per inch (DPI) in 24-bit RGB color. The articles of interest are isolated from the rest of the content on the page and saved as individual TIFF image documents. Originally, the 24-bit RGB document images were OCRed using Abbyy FineReader version 10.0 commercial OCR engine; however, the resulting OCR did not meet the expectations of the collection curator and have been manually corrected. This manually corrected text is the ground truth against which word error rates in this paper are calculated.

Table 7.1: Initial average WERs on a subset of the 19th Century Mormon News Articles image corpus. (The Sauvola parameters are given in parentheses.)

Image Type or Binarization	*Average WER*
RGB	11.19%
Grayscale	11.77%
Otsu	13.81%
Sauvola (radius=7, R=128, k=0.5)	14.37%
Sauvola (radius=45, R=128, k=0.5)	14.01%

For this work, the 24-bit RGB document images are reduced to 8-bit grayscale at 400 DPI using Adobe PhotoShop default settings. This is the starting point for evaluating the grayscale, the adaptive binarized, and the thresholded images of this research. Applying the binarization techniques of Otsu [79] and Sauvola [92] on a subset of the grayscale images in the corpus, we use the Sclite [3] scoring package to calculate the WERs from the OCR output of Abbyy FineReader on the binarized images. The results of this are in Table 7.1.

From the perspective of WERs, the two adaptive binarization techniques are not an improvement over the OCR from either the RGB or grayscale images, which were OCRed as is without modification. As the OCR software used in this research is proprietary, it is not clear what the source of the difference between the WER of the RGB and grayscale images is. For the remainder of this paper, we chose to proceed with the best results using the Otsu and the Sauvola (radius=45, R=128, k=0.5) methods.

7.3.2 Binary Thresholding

Threshold binarization, or thresholding, is a simple process which uses a single numeric value to differentiate between pixels in the digital grayscale image that are assigned the value 0 (for black) or 255 (for white). The seven values we selected for thresholding (31, 63, 95, 127, 159, 191, and 232) divide the range of the 8-bit grayscale into eight equal groups, representing the same incremental difference in threshold value between groups. Within a document all pixels less than or equal to the thresholding value are assigned black, and those greater than the value are assigned white. In order to be useful in the context of a very large digitized corpus, we selected the threshold values to equally divide the grayscale range, rather than individually selecting thresholds based on the characteristics of a single document. Further, we attempt to demonstrate how multiple views or thresholds of the same document can provide improved performance in the word recognition task. Examples of the thresholded output from a portion of a document in the corpus can be seen in Figure 7.3.

Similar to adaptive binarization discussed in Section 7.2.2, where the goal is to account with a single binarized image for variations in background noise or character darkness in the original digitized image, our method of creating multiple thresholded images provides a breadth of image manipulation, capturing character images that may be visible in one thresholding, but not in another.

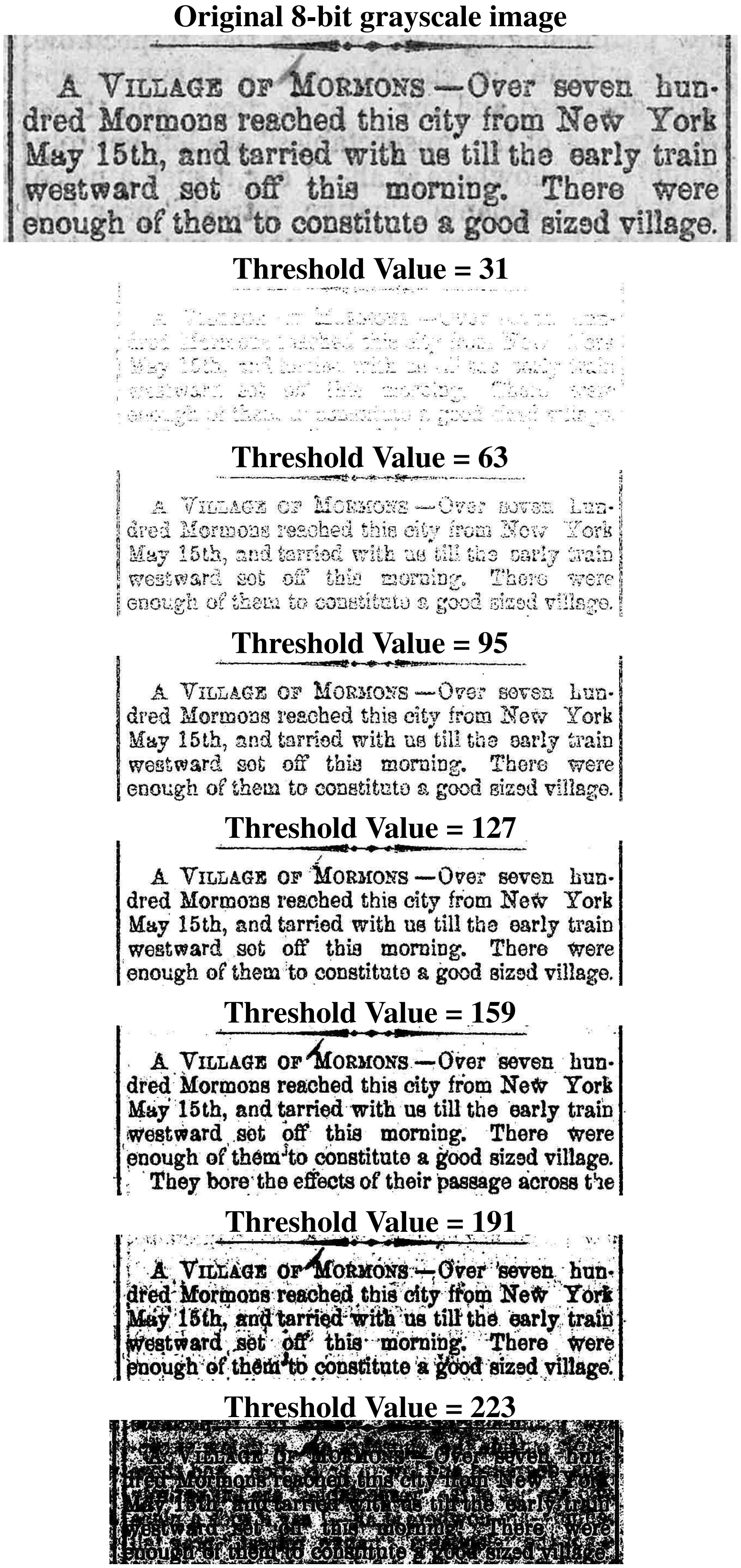
7.3.3 Optical Character Recognition and Word Error Rate

Each thresholded image derived from the grayscale images is OCRed using Abbyy FineReader version 10.0. For this work we have chosen to measure performance by word error rates rather than pixel error rates. Since the goal of this work is ultimately a transcript of the digitized printed document, word error rates more closely match the goal than pixel error rates.

The WER of the thresholded images varies significantly, with the lowest average WER occurring at the threshold value of 127. Although for this corpus the thresholding value with the lowest average WER occurs at the mid-point between 0 and 255, this is not necessarily always the case. Figure 7.4 and Table 7.2 show the averaged WERs of the documents in the corpus. Due to the very high WER for the thresholded values 31 and 223, we exclude these thresholded images in the remaining steps.

7.3.4 Alignment and Hypothesis Lattice

Next in our process is a progressive alignment of the OCR text output of the thresholded images from the same gray-scale image (threshold values: 63, 95, 127, 159, and 191). Progressive text alignment can be approached in a variety of ways. A greedy approach, used by Boschetti, et al. [10],



Original 8-bit grayscale image

A VILLAGE OF MORMONS.—Over seven hundred Mormons reached this city from New York May 15th, and tarried with us till the early train westward set off this morning. There were enough of them to constitute a good sized village.

Threshold Value = 31

Threshold Value = 63

Threshold Value = 95

Threshold Value = 127

Threshold Value = 159

They bore the effects of their passage across the

Threshold Value = 191

Threshold Value = 223

Figure 7.3: Partial original grayscale image and seven thresholded images of the document BM_24May1859_p2_c3 from the 19th Century Mormon Newspaper Article collection [29]

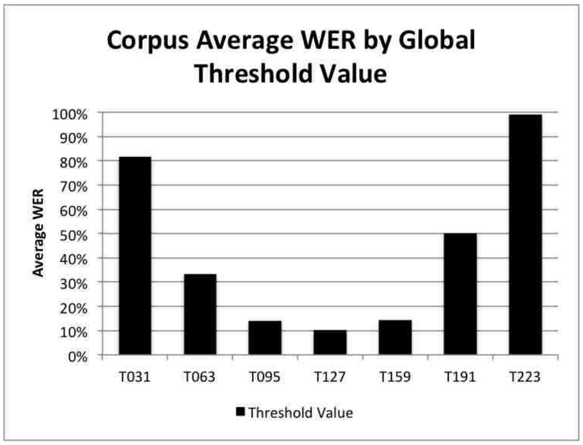
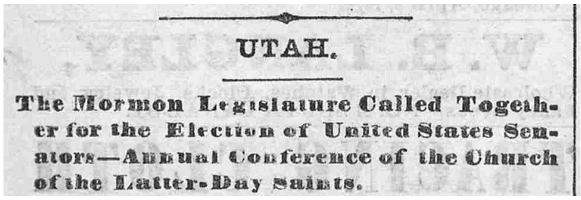


Figure 7.4: Average OCR WER of the document images after being thresholded at the various values between 31 and 223.

Grayscale image



UTAH.

The Mormon Legislature Called Together for the Election of United States Senators—Annual Conference of the Church of the Latter-Day Saints.

Thrshld	OCR Text Output
T127	1’kt **Morm--on** --a---u--t--r-e **Called**
T095	**The** aJorsBsoM [sasins--H--a-e **Called**
T159	Tke -morm--on leg-i-slatu-r-e **Called**
T063	J(- --------3 --------------e ---x--
T191	**The -Morm--on** l<rgi-slaiiir-e **Called**
Transcript	The Mormon Legislature Called
T127	**Togeth- er for the** Uenu -t--sf **United**
T095	Togcih-ti- far **the** litction cf Uvtttd
T159	Togeth--er **for the Election-** f **United**
T063	----->- i) 1 , - w -ij -----"- -1 t [
T191	Togeth--er **for the Election of United**
Transcript	Togeth- er for the Election of United
T127	Stattst **Sen-**-aiort-
T095	Sla-ws- Sew--aiorfc
T159	**States- Sen-**-aioro-
T063	J------ --n--------
T191	**States-** gen> **ators-**
Transcript	States Sen- ators

Figure 7.5: An example of an aligned lattice from document CT_2Apr1872_p2_c1. Correct hypotheses in the lattice are highlighted in “**bold**.” The “dash” character represents an *INDEL* in the alignment, where a character needs to be inserted in order to complete the alignment.

identifies the two sequences (of the n sequences to be aligned) that have the lowest alignment cost. Then from the $n-2$ sequences remaining, the next sequence to be aligned is the one with the lowest alignment cost with the first two previously aligned sequences, where the alignment between the already aligned sequences can not be altered other than to insert a gap (an $INDEL$ character) across the alignment to match the new (third) sequence. This process is repeated until all n sequences have been aligned in a progressive manner. We adapt Boschetti's method by aligning the two sequences that have the lowest WER, adding sequences in the order of their increasing WERs. We demonstrated in Lund, Walker, and Ringger (2011) [64] that the order of the alignment does not have a major effect on the outcome of the resulting WER for this task.

For this paper the thresholded document sets with the lowest average WER are those with threshold values 95 and 127. (Refer to Figure 7.4.) We continue the alignment, one sequence at a time, in order of increasing WER. This continues until all of the OCR output from the sets in use have been aligned. The order of the complete alignment is threshold values: 127, 95, 159, 63, 191. Figure 7.5 shows a portion of an alignment along with the transcript.

Given the character-aligned OCR sequences the next step is to segment parallel hypotheses into a lattice which permits comparisons between the OCR hypotheses from various thresholded images. The aligned text sequences are formed into a lattice divided into "columns" of hypotheses across all aligned sequences where there is complete agreement on spaces. These columns can be seen in the sample alignment shown in Figure 7.5. Note in the lattice that no single OCR text sequence of a thresholded image is correct for all words, and that some words are not found correctly by any sequence. Using the methods of Lund, Walker, Ringger (2011) [64], a single token from the column is selected using a supervised discriminative machine learning tool.

7.4 Results

In the literature both character error rates and word error rates are used to indicate the improvement in underlying OCR results. We choose to use word error rates since the goal of this work is to improve the accessibility of degraded documents whose OCR limits their usefulness. This agrees

Table 7.2: Baseline average WERs on all binarizations of the document image corpus.

Image Type or Binarization	*Average WER*
Grayscale	9.08%
Otsu	14.15%
Sauvola (radius=45, R=128, k=0.5)	14.28%
Binarization Threshold Values	
T031	78.87%
T063	27.39%
T095	12.22%
T127	9.94%
T159	11.75%
T191	42.01%
T223	99.17%

with Kolak et al. [52] who primarily use word error rate for measuring success in NLP tasks. The WER is calculated using the Sclite tool provided by NIST [3].

7.4.1 Baseline Results

As a baseline to compare the results of our method, we binarize the grayscale images of the entire corpus using both the Otsu and Sauvola (radius=45, R=128, k=0.5) methods. The WER results are found in Table 7.2.

7.4.2 Lattice Word Error Rate

The Lattice Word Error Rate (LWER) is an oracle calculation. In each column of the lattice if the correct word (or words) are present in any of the hypotheses, it is considered a correct match. For example, in Figure 7.5 the word from the transcript “The” is found in the OCR from the thresholded images with values 95 and 191. This is considered a match. Note that in the third column, the word “Legislature” in the transcript, is not found in any of the hypotheses. (Note that the “dash” character represents an *INDEL* in the alignment, meaning that there is either an insertion or a deletion of a character in the alignment. Remove all *INDEL*s from a hypothesis to see the text as provided by the OCR engine.) The reason that the word “Legislature” was not matched is that the capital “L” was

Table 7.3: Both the individual WERs of the thresholded files and the cumulative lattice word error rates of the complete document CT_2Apr1872_p2_c3.

Number of Sequences	Threshold Value Added	Threshold Word Error Rate	Resulting Lattice Word Error Rate
1	T127	41.21%	41.21%
2	+T095	110.91%	38.18%
3	+T159	30.91%	26.06%
4	+T063	100.00%	25.45%
5	+T191	22.42%	11.00%
Grayscale WER = 30.30% Otsu WER = 30.30% Sauvola WER = 40.61%			

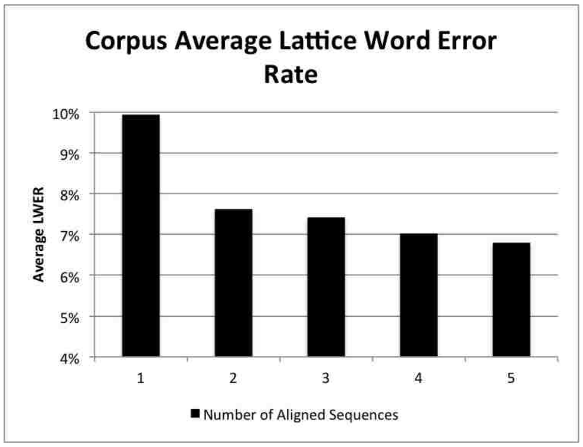


Figure 7.6: The average lattice word error rate as the number of sequences aligned increases.

not a part of the output from the OCR of thresholded image value 159. The LWER of Figure 7.5 is $(14-12)/14 = 14\%$. There are fourteen tokens in the transcript (note that some words are split between lines, creating two tokens), of which two do not have a correct hypothesis from any of the OCR sequences.

Consider document CT_2Apr1872_p2_c3, a portion of which is shown in Figure 7.5. Table 7.3 shows the WERs of the OCR from the various thresholded document images, as well as the LWER as each OCR sequence is added to the aligned lattice. Observe that as more OCR sequences or hypotheses are added to the lattice, the LWER is reduced, even though the WERs of each sequence is significant. Even OCR sequences from thresholded images that have a WER

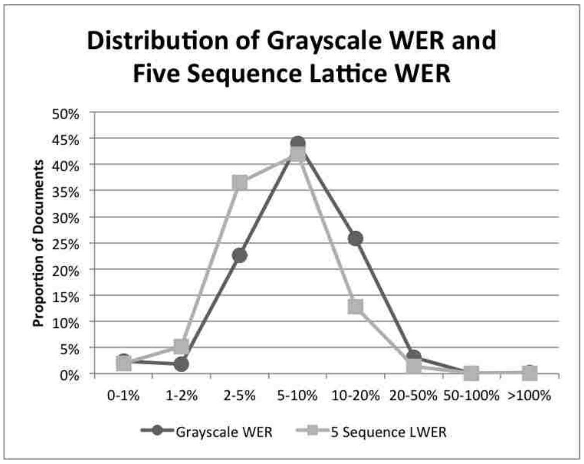


Figure 7.7: Distribution of documents error rates. Grayscale average WER is 9.08%. Five sequence average LWER is 6.79%

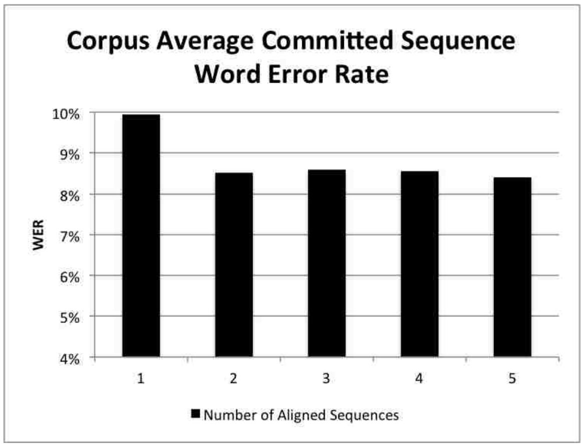


Figure 7.8: The average corpus WER after committing to a single hypothesis from each column of the lattice.

over 100% [1] contribute to lowering the LWER. Ultimately for the document shown in Table 7.3 the LWER is significantly lower than the WER of the OCR output of any one individual thresholded image, or of the grayscale WER.

Considering the entire corpus, we see in Figure 7.6 that on average adding additional sequences continues to improve the LWER at each step. And finally, looking at the distribution of the WERs of the grayscale images compared to the LWERs of the aligned sequences from five thresholded documents in Figure 7.7, we see that the distribution curve has shifted lower. This is an indication that the LWER broadly improves across the entire corpus with five OCR sequences from the thresholded document images.

7.4.3 Committed Word Error Rate

As promising as the LWER may be, ultimately it is only an indication of the lower bound on any methods which may use this lattice approach to select or commit to a single sequence of tokens. (Note that some methods using multiple OCR sequences of the same document will output all OCR hypotheses. Although this is useful for word finding, it is not helpful for other tasks such as machine translation, phrase finding, or transcription creation.) We use the method of Lund, Walker, and Ringger [64] to commit to a single sequence of hypotheses from the lattice, which employs Mallet [72], a supervised discriminative machine learning tool. Since no training set was available for Mallet, we opted to divide the corpus into four segments using round robin training and testing. Each of the segments was used as a testing set, while the remaining three were used for training. The averaged results across all four testing segments are found in Figure 7.8.

In general, the WER of the committed sequences from the lattice of multiple sequences is lower than the average greyscale WER as well as lower than the WERs of the Otsu and Sauvola binarizations. However, an increase in the number of sequences from two to five did not materially reduce the average WER.

[1] A WER of over 100% is possible when the OCR engine interprets document noise as word tokens or when words are incorrectly split.

Table 7.4: Summary of results on the complete corpus

Method	**Average Results**
Grayscale	WER 9.08%
Otsu	WER 14.15%
Sauvola (radius=45, R=128, k=0.5)	WER 14.28%
Threshold Value=127	WER 9.94%
Five Aligned Thresholded Sequences	WER 8.41% LWER 6.79%

7.5 Conclusions

We have shown a novel method for correcting OCR errors using multiple simply thresholded binarizations of a grayscale image of a document. The various threshold values for the document image expose a diversity of characters and words, which when aligned in a lattice of OCR hypotheses, has a lower lattice word error rate than any of the individual thresholded images or the images binarized using current techniques. (See Table 7.4.)

The lattice word error rate indicates a lower limit on the word error rate that is possible given the evidence found in the combination of OCR outputs from the thresholded document images. To achieve this lower bound, perfect decisions need to be made regarding the hypotheses in each column of the lattice as defined in Section 7.3.4. Using an existing method [64] to make these decisions, we selected sequences of OCR hypotheses from the lattice of each document. This method achieved an average WER on the corpus of 8.41% which is lower than any of the baseline results.

7.6 Addendum

Since the submission and publication of this paper, there are papers which relate to this work which need to be cited. First, I have become aware of a patent [18] which uses multiple binarizations and voting between outputs based on unspecified metrics. Without knowledge of the specific metrics used, the method of aligning the outputs, or the methods of extracting competing hypotheses for a given token or sequence of tokens, it is difficult to evaluate how this overlaps with existing work. At the very least, this would seem to be an extension of [59], already cited in this work.

Also related to this research, Rangoni et al. (2009) [83] used multiple binarizations to improve OCR by searching for the single best binarization prior to OCR of the digitized image. For each candidate binarization their process segmented and extracted a line of text as an image from the complete document image. Using multiple runs, varying input parameters, a commercial OCR engine, the extracted image of a line of text is recognized and the resulting text is measured for accuracy using a language model and dictionary which may be manually augmented with out of vocabulary tokens common to the corpus. The binarization run with the best metric is then used for the entire document. A recent paper by Ray, A. et al. (2013) [84] also uses a conditional random field to select the best binarization, where the document image is binarized three times with varying parameters of the Sauvola algorithm. (Refer to Section 7.2.2.) The method selects a segmented character from the three binarizations using a CRF model trained on the OCR recognition of the letter and a language model. In both of the papers cited above, the goal is to find only a single binarization rather than using the information available in multiple binarizations.

Chapter 8

Why Multiple Document Image Binarizations Improve OCR

Abstract

My previous work has shown that the error correction of optical character recognition (OCR) on degraded historical machine-printed documents is improved with the use of multiple information sources and multiple OCR hypotheses including from multiple document image binarizations. The contributions of this paper are in demonstrating how diversity among multiple binarizations makes those improvements to OCR accuracy possible. We demonstrate the degree and breadth to which the information required for correction is distributed across multiple binarizations of a given document image. Our analysis reveals that the sources of these corrections are not limited to any single binarization and that the full range of binarizations holds information needed to achieve the best result as measured by the word error rate (WER) of the final OCR decision. Even binarizations with high WERs contribute to improving the final OCR. For the corpus used in this research, fully 2.68% of all tokens are corrected using hypotheses not found in the OCR of the binarized image with the lowest WER. Further, we show that the higher the WER of the OCR overall, the more the corrections are distributed among all binarizations of the document image.

Published in *Proceedings of the Workshop on Historical Document Imaging and Processing 2013 (HIP 2013)* [66].

Table 8.1: Word error rates of OCRed document images from the 19th Century Mormon Article Newspaper Index.

	Abbyy	Omnipage	Tesseract	Adobe
Average	8.29%	11.24%	20.74%	25.59%
Minimum	0.00%	0.00%	0.00%	0.00%
Maximum	171.05%	173.68%	171.05%	165.79%

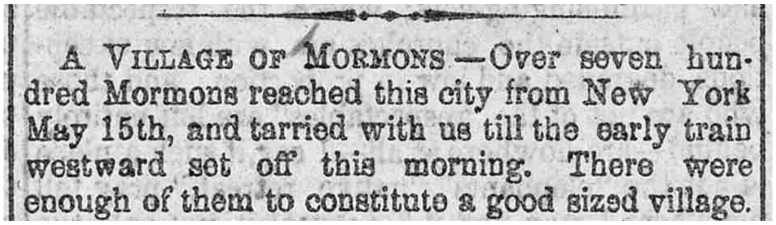
8.1 Introduction

Historical machine printed document images often exhibit significant noise, making the optical character recognition (OCR) of the text difficult as shown in Table 8.1 in which WERs over 100%[1] are observed. Lund et al. (2013) shows that it is possible for multiple binarizations of the same document image to be combined post-OCR using machine learning techniques to provide text output with a lower word error rate (WER) than the OCR of any one binarization alone [65].

There has been much research towards creating a single binarized digital image from a scanned document, which maximizes the correct identification of foreground vs. background pixels. This paper does not solve the binarization problem, rather it explores the extent to which the diversity among the OCR outputs from multiple binarizations can improve the opportunities for error correction using ensemble methods. Further, methods are used which are scalable to very large collections, up to millions of images, without document- or corpus-specific manipulation of training data, which would be infeasible given time and resource constraints.

Ensemble methods are used effectively in a variety of problems such as machine translation, speech recognition, handwriting recognition, and OCR error correction, to name a few. In a paper on the framework for ensemble methods, Kittler et al. [48] state: ”It had been observed ... that although one of the [classifiers] would yield the best performance, the sets of patterns mis-classified by the different classifiers would not necessarily overlap. This suggested that different classifier designs potentially offered complementary information about the patterns to be classified which could be harnessed to improve the performance of the selected classifier.” Previously we have

[1] WERs over 100% are possible due to insertion errors made by the OCR engine.



A VILLAGE OF MORMONS —Over seven hundred Mormons reached this city from New York May 15th, and tarried with us till the early train westward set off this morning. There were enough of them to constitute a good sized village.

Figure 8.1: A fragment from the grayscale scan of document BM_24May1859_p2_c3 from the 19th Century Mormon Article Newspaper Index [29].

merged complementary information such as the output of multiple OCR engines [61] and multiple binarizations of the same document image [65]. The goal of this paper is to study the breadth and degree of the contributions of various binarizations of the same document image to the correction of the original OCR.

The remainder of this paper proceeds as follows. Section 8.2 discusses existing work in several fields related to the methods and outcomes of this research. Section 8.3 outlines the methodology used to create the binarized text images, the OCR of those images, a brief overview of the methods used to extract corrected text with machine learning techniques, leading to the heart of the paper, the evaluation of the extent to which correct word candidates are distributed across all image binarizations. Section 8.4 lays out and discusses the results of this research. Finally, Section 8.5 summarizes the conclusions of this research.

8.2 Related Work

We consider four areas in which there is related work to this research: OCR error correction, text alignment, ensemble methods, and image binarization.

8.2.1 OCR Error Correction

Printing and duplication techniques of the mid-20th century and earlier create significant problems for OCR engines. Examples of problematic documents include typewritten text, in which letters are partially formed, typed over, or overlapping; documents duplicated by mimeographing, carbon paper, or multiple iterations of photographic copying common in the mid-20th century; and newsprint which uses acidic papers and type that can exhibit incomplete characters. In addition to original documents which may exhibit problematic text, they may suffer degradation such as bleed-through of type and images, damage due to water, and discoloring of the paper itself. (See Figures 8.1, 8.2, and 8.6 for examples of grayscale images from 19th century newspapers.)

Extracting usable text from older, degraded documents is often unreliable, frequently to the point of being unusable [4]. Even in situations where a fairly low character error rate is achieved, intolerable word error rates may result: e.g., Hull [38] points out that a 1.4% character error rate results in a 7% word error rate on a typical page of 2,500 characters and 500 words. Kae and Learned-Miller [44] remind us that OCR is not a solved problem and that "the goal of transcribing documents completely and accurately... is still far off." At some point the word error rate of the OCR output inhibits the ability of the user to accomplish useful tasks. In a related study, Munteanu et al. [74] determined that in an automated speech recognition task creating text transcriptions of lectures, a transcript with a WER of 50% was no better than having no transcript at all.

Many approaches have been taken to reduce the error rate of OCR output. It is well known from early work by Lopresti and Zhou [59] that simply voting among multiple sequences generated by the same OCR engine can significantly improve OCR. Esakov, Lopresti, and Sandberg [25] evaluated recognition errors of OCR systems, and Kolak, Byrne, and Resnik [52] specifically applied their algorithms to OCR systems for post-OCR error correction in natural language processing tasks. OCR error correction with in domain training [63] as well as out-of-domain training using a synthetic training dataset [64] have been shown to be effective.

This paper extends our previous work [65] by quantifying the extent to which information already available in the original grayscale image and extracted through multiple binarizations may be leveraged to improve OCR output.

8.2.2 Text Alignment

Implicit in any post-OCR error correction using multiple sequences is an alignment of the text sequences, which can either use exact or approximate algorithms. The multiple sequence alignment problem has been shown to be NP-Hard by Wang and Jiang [109]. Lund and Ringger [61] demonstrate an efficient means for exact alignment; however, alignment problems on long sequences are still computationally intractable. The vast majority of multiple sequence alignment work is done in the field of bioinformatics, where the size of the alignment problems has forced the discovery and adoption of heuristic solutions such as progressive alignment. Elias [23] discusses how a simple edit distance metric, which may be appropriate to text operations, is not directly applicable to biological alignment problems, which means that much of the alignment work in bioinformatics requires some adaptation for use in the case of text.

Ikeda and Imai [39] and later Schroedl [93] represent alignment problems as minimum distance problems in a directed acyclic graph in which edge costs represent the cost of alignment decisions. Further, Ikeda & Imai (and Schroedl) formulated the cost of DNA base (or character) level alignments in n-dimensions as the sum of the costs of 2-dimensional alignments.

This paper will use the same progressive alignment method from our previous work [61], and the alignment order of the binarization threshold values of the training set used in [65]. (Section 8.3.2 describes the binarization method used in this work.)

8.2.3 Ensemble Methods

Voting among multiple outputs is a common ensemble method used in OCR research. Lopresti and Zhou [59] use voting among multiple sequences generated by the same OCR engine from different scans of the same document. Their experimental results showed between 20% and 50% reduction in errors. Our method [65] differs from that of Lopresti and Zhou in that although we start with the same digital image, rather than adjusting the scans of the document, we adjust the binarization threshold. Recent work by Yamazoe et al. [114] effectively uses multiple weighted finite-state transducers (WFST) with both the OCR and a lexicon of the target language(s) to resolve the ambiguity inherent in line- and character-segmentation, and character recognition, in which the number of combinations can be very large. Both conventional OCR and post-processing are contained within their system, resolving the difference between various hypotheses before committing to an output string.

Klein and Kobel [49] as well as Cecotti and Belaïd [14] note the differences between OCR outputs can be used to advantage. This observation is behind the success of ensemble methods, that multiple systems which are complementary can be leveraged for an improved combined output. It should be noted that the complementarity of correct responses of the methods is critical. This paper will expand on the observation regarding complimentary sources, noting that one source of diversity can come from multiple binarizations of the document scan itself.

8.2.4 Image Binarization

Image binarization methods create bitonal images of grayscale or color digital images, with the intention of separating foreground from background information. The simplest form of binarization is global thresholding, in which a grayscale intensity threshold is selected and each pixel is set to either black or white depending on whether it is darker or lighter than the threshold, respectively.

Since the exposure and contrast of document images can vary, it is often not possible to select a single threshold that is suitable for an entire image. (See Figure 8.1 for an example.) Background noise, stray marks, ink bleed-through from the back side of a page, or ink bleed-through from a facing page may be darker than some of the desired text. Stains, uneven brightness, paper fiber, paper degradation, or faded print can mean that some parts of the page are too light for a given threshold while other parts are too dark for the same threshold. Adaptive thresholding methods attempt to compensate for inconsistent brightness and contrast in images by selecting a threshold for each pixel based on the properties of a small portion of the image window surrounding that pixel, instead of the whole image. While various methods perform well in many situations, no method works infallibly under all circumstances. Our previous work [65] compared adaptive binarization to the multiple binarization methods discussed here. Furthermore, the goal of this work is to understand the contributions of multiple threshold binarizations to the OCR error correction problem. No known work considers the contributions of multiple adaptive binarizations.

When digitizing degraded historical text documents our goal is to recognize the text, not to identify whether a pixel is a foreground or background pixel of the image. The goal of this paper is to demonstrate the degree to which multiple threshold binarizations provide complementary OCR sources, which when combined with principled machine learning techniques reduce the resulting WER of the OCR.

8.3 Method

In this section we present the method for studying the contributions of multiple document image binarizations to the task of correcting OCR.

As an overview, the first step in the methodology is the threshold binarization of the document images. The individual binary images are OCRed, and the resulting parallel OCR output from the same document is aligned. In other papers ([64], [65]) we present machine learned models to select individual tokens and thereby commit to a specific hypothesis, from which a WER can be calculated. A Lattice WER can also be calculated as an oracle calculation based on whether any of the possible inputs was correct. Table 8.2 shows these results. Section 8.4 will explore the reasons why these previously published methods are successful.

8.3.1 Document Corpus

The document corpus used in this study is a collection of 1055 color images from an archive of historical newspapers of the 19th century [29]. As expected from 19th century newsprint, the quality of the paper and the print was poor when first printed and has further degraded over time. The newspapers were recently scanned at 400 dots per inch (dpi) in 24-bit RGB color, and the individual articles were segmented and saved as TIFF images. The RGB images were then converted to 8-bit grayscale and subsequently evaluated by an OCR engine, Abbyy FineReader 10, with the default dictionary[2] enabled. The OCR output of each document was manually corrected by two reviewers

[2] The default dictionary was used since one of the goals of the research is to develop methods applicable to very large scale collections, for which creating specialized training sets or dictionaries is not feasible, given time and financial constraints.

Table 8.2: 19thCMAN corpus average word error rates.

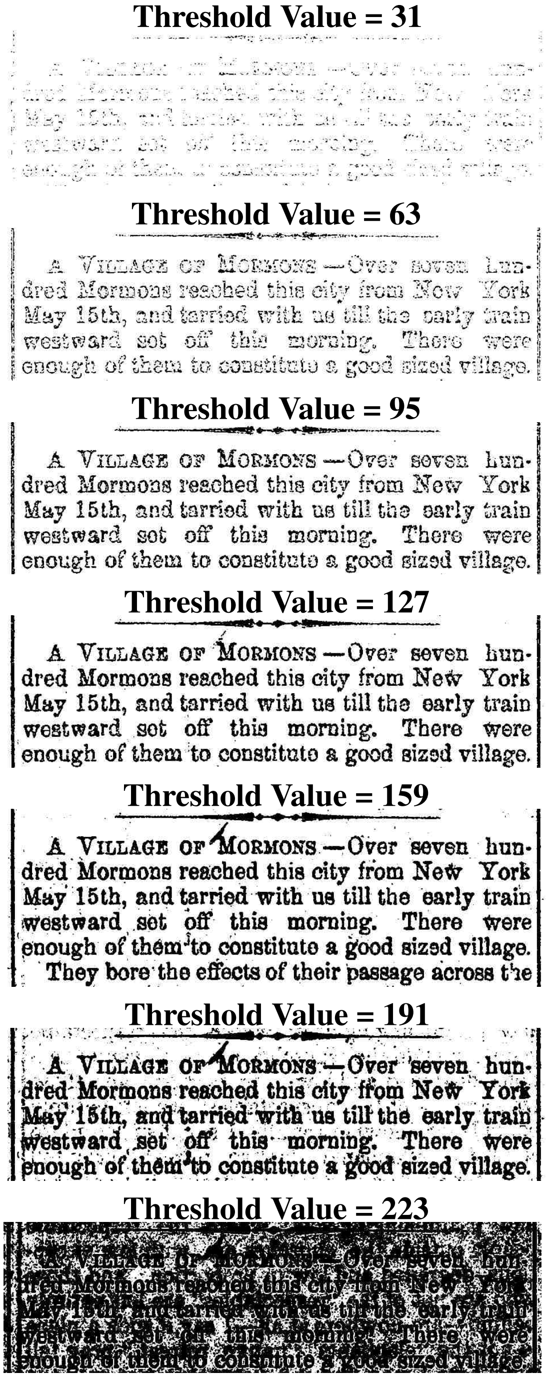
Image Source or Method	**WER**
Grayscale images	11.77%
Single best binarization (Threshold value: 127)	9.94%
Sequence selected by a machine learned model	8.41%
Corpus-wide Oracle Calculations	**WER**
For each document, select the binarization with the lowest WER	7.70%
Lattice Word Error Rate across all binarizations for each document	6.78%

to act as a gold standard. An example from the document corpus can be seen in Figures 8.1, 8.2, and 8.6. Note the noise and variations in background which are common to the entire corpus. The WER on the OCR of the grayscale images, calculated as an average of the WER for each document, is 11.77%. (See Table 8.2, row one.)

8.3.2 Grayscale Documents and Thresholding

For this research we have created fixed threshold binarizations from each grayscale document. Seven evenly spaced threshold values (31, 63, 95, 127, 159, 191, 223) divide the 8-bit grayscale range. Examples of these binarization thresholds for a single document image are shown in Figure 8.2. We selected fixed threshold values to be used across the entire corpus, so that we are able to consistently compare binarizations at the various threshold values. The average WERs of the OCR results on the binarized images is shown in Figure 8.3. There is significant variation in the average WER of the OCR of the binarized images, with the best average WER resulting from the OCR of the binarization threshold value at 127, the mid-point of the range of possible threshold values for 8-bit grayscale. This distribution of WERs shows that there is significant noise in images from all of the binarization threshold values.

Figure 8.4 shows a typical distribution of grayscale pixel values for a document. Note there are two modes, one associated with light values (the "white mode") and another for dark values (the "black mode"). The arrows indicate the seven binarization thresholds used in this study. Further,



Threshold Value = 31

Threshold Value = 63

A VILLAGE OF MORMONS.—Over seven hundred Mormons reached this city from New York May 15th, and tarried with us till the early train westward sot off this morning. There were enough of them to constitute a good sized village.

Threshold Value = 95

A VILLAGE OF MORMONS.—Over seven hundred Mormons reached this city from New York May 15th, and tarried with us till the early train westward sot off this morning. There were enough of them to constituto a good sized village.

Threshold Value = 127

A VILLAGE OF MORMONS.—Over seven hundred Mormons reached this city from New York May 15th, and tarried with us till the early train westward sot off this morning. There were enough of them to constituto a good sized village.

Threshold Value = 159

A VILLAGE OF MORMONS.—Over seven hundred Mormons reached this city from New York May 15th, and tarried with us till the early train westward set off this morning. There were enough of them to constituto a good sized village. They bore the effects of their passage across the

Threshold Value = 191

A VILLAGE OF MORMONS.—Over seven hundred Mormons reached this city from New York May 15th, and tarried with us till the early train westward set off this morning. There were enough of them to constitute a good sized village.

Threshold Value = 223

Figure 8.2: Seven thresholded images of the document BM_24May1859_p2_c3 from the 19th Century Mormon Newspaper Article collection [29]. The original grayscale image is found in Figure 8.1.

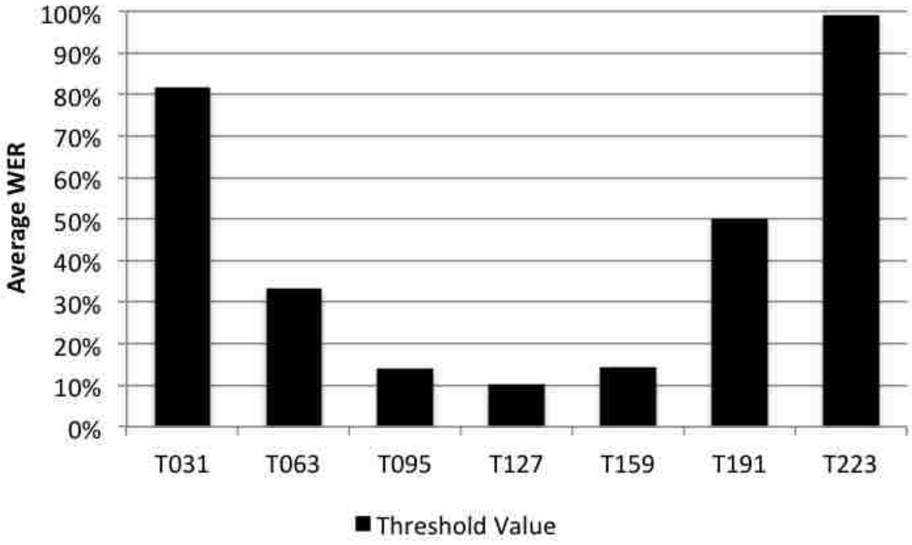


Figure 8.3: Corpus average WER by global threshold value at selected values between 31 and 223.

Figure 8.5 shows that the distributions of the black and the white modes vary significantly across the entire corpus, another indication of the noise and variability in the corpus.

8.3.3 Merging the OCR of Multiple Binarizations

Next in our process is a progressive alignment of the OCR text output for each of the thresholded images from the same gray-scale image. Progressive text alignment can be approached in a variety of ways. A greedy approach, used by Boschetti et al. [10], identifies the two sequences (of the n sequences to be aligned) having the lowest alignment costs. Then from the $n - 2$ remaining sequences, the next sequence to be aligned is the sequence with the lowest cost of alignment against the first aligned pair. In aligning the new sequence the alignment of the already aligned sequences can not be altered beyond the insertion of a gap across the alignment to match a character in the new (third) sequence. This process is repeated until all n sequences have been aligned in a progressive manner. We adapt Boschetti's progressive method by aligning in the order increasing WERs as was found in the training set used in [65]. We have previously demonstrated that the order of the alignment does not have a major effect on the outcome of the resulting WER for this task [64]; however, alignment errors can contribute to correction errors. For this paper the

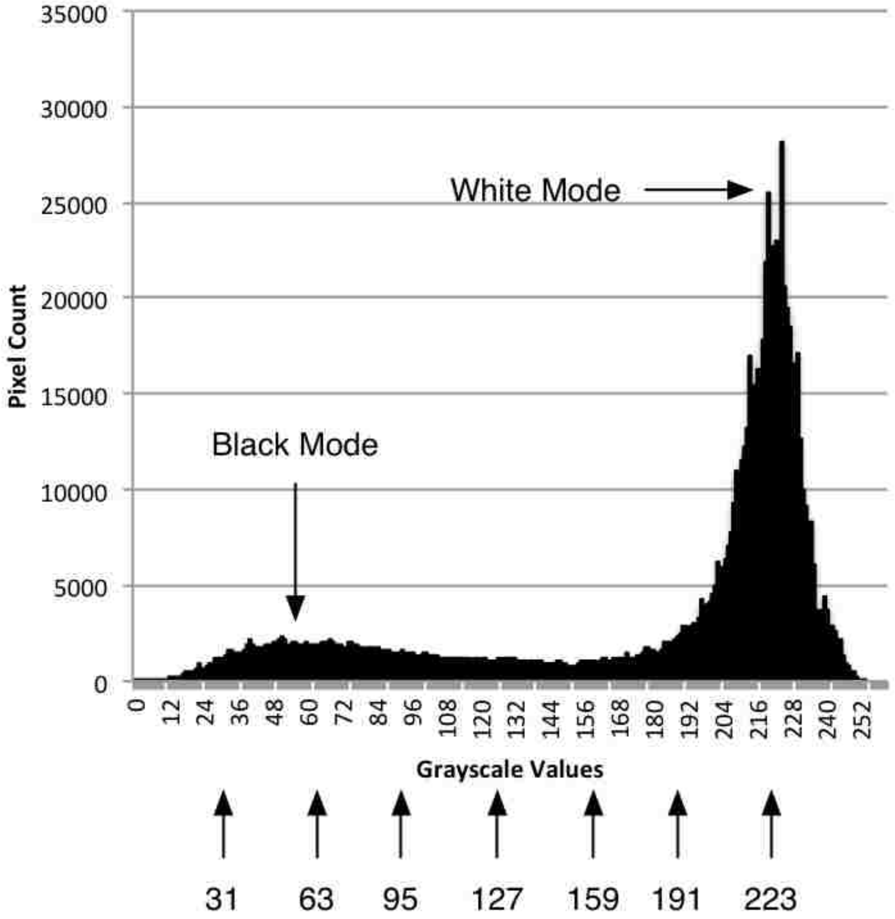


Figure 8.4: The distribution of grayscale values for document BM_24May1859_p2_c3. (Cf. Figure 8.1). The two high points corresponding to pixels which are closest to black (black mode) and to white (white mode) are indicated. The seven threshold values are shown below the graph.

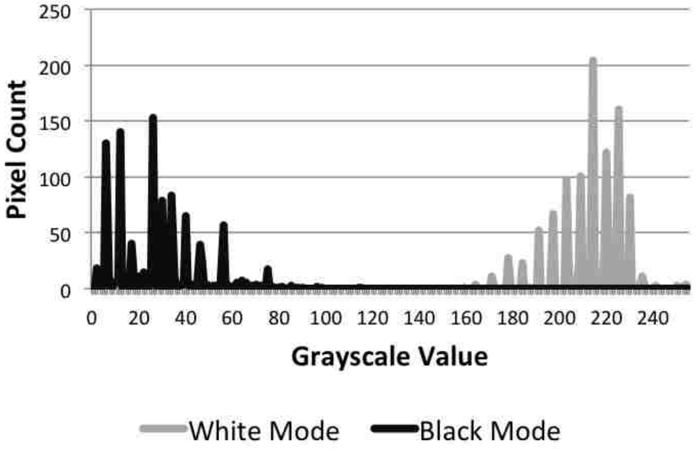


Figure 8.5: The distribution across all documents of the black and the white modes.

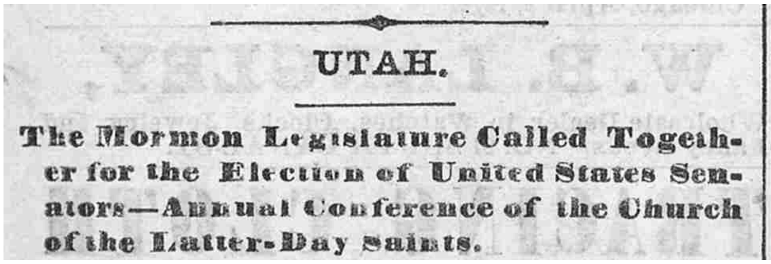
thresholded document sets with the lowest average WER are those with threshold values 95 and 127 (refer to Figure 8.3). This continues until all of the OCR output from the sets in use have been aligned. The order of thresholds in the complete alignment is: 127, 95, 159, 63, 191, 31, 223. Figure 8.6 shows a portion of an alignment along with the transcript.

Given the character-aligned OCR sequences the next step is to segment parallel hypotheses into a lattice which permits token-level comparisons among the OCR hypotheses from various thresholded images. The aligned text sequences are divided into "columns" of hypotheses across all aligned sequences, where column boundaries occur wherever there is complete agreement on white space. These columns can be seen in the sample alignment shown in Figure 8.6. Note in the lattice that no single OCR text sequence of a thresholded image is correct for all words, and that some words are not found correctly by any sequence. Using the methods of Lund et al. [64], a single token from the column is selected using a model learned by supervised discriminative machine learning [72]. The WER result computed from this method is found in Table 8.2, row 3.

For the purposes of this research, we measure the quality of a binarization by the WER of the OCR output. Although it is common to measure the quality of a binarization by its ability to correctly identify foreground vs. background pixels, ultimately for text document images our goal is to extract accurate text. Our previous work [65] has shown how to effectively use machine learning methods on multiple binarization sources for OCR. For the adaptive binarization methods with which we experimented, the multiple threshold method performed better than any adaptive method alone. (No known results are available for multiple adaptive hypotheses.) In particular in our previously cited work, a machine learner selecting a single sequence taken from the lattice of OCR results from the multiple binarizations resulted in 8.41% WER. (See Table 8.2, row 3).

8.4 Results

There are two significant results from this study. First, sources, such as binarizations, with higher error rates can contribute to an overall improvement of OCR. Second, document images with a high OCR WER can benefit from the diversity of information across the OCR of many document



UTAH.

The Mormon Legislature Called Together for the Election of United States Senators—Annual Conference of the Church of the Latter-Day Saints.

<table>
<tr><th>Thresh-old</th><th>OCR Text Output</th></tr>
<tr><td>T127</td><td>1kt | -Morm--on | --a---u--t--r-e | Called</td></tr>
<tr><td>T095</td><td>The | aJorsBsoM | [sasins--H--a-e | Called</td></tr>
<tr><td>T159</td><td>Tke | -morm--on | leg-i-slatu-r-e | Called</td></tr>
<tr><td>T063</td><td>J(- | --------3 | --------------e | ---x--</td></tr>
<tr><td>T191</td><td>The | -Morm--on | l<rgi-slaiiir-e | Called</td></tr>
<tr><td>T031</td><td>--- | --------- | --------------- | ------</td></tr>
<tr><td>T223</td><td>--- | --------- | --------------- | ------</td></tr>
<tr><td>Transcript</td><td>The | Mormon | Legislature | Called</td></tr>
<tr><td>T127</td><td>Togeth- er | for | the | Uenu -t--sf | United</td></tr>
<tr><td>T095</td><td>Togcih-ti- | far | the | litction cf | Uvtttd</td></tr>
<tr><td>T159</td><td>Togeth--er | for | the | Election- f | United</td></tr>
<tr><td>T063</td><td>----->- i) | 1 , | - w | -ij -----"- | -1 t [</td></tr>
<tr><td>T191</td><td>Togeth--er | for | the | Election of | United</td></tr>
<tr><td>T031</td><td>---------- | --- | --- | -------- -- | ------</td></tr>
<tr><td>T223</td><td>---------- | --- | --- | E------- -- | ------</td></tr>
<tr><td>Transcript</td><td>Togeth- er | for | the | Election of | United</td></tr>
<tr><td>T127</td><td>Stattst | Sen--aiort-</td></tr>
<tr><td>T095</td><td>Sla-ws- | Sew--aiorfc</td></tr>
<tr><td>T159</td><td>States- | Sen--aioro-</td></tr>
<tr><td>T063</td><td>----->- | i) 1 , - w</td></tr>
<tr><td>T191</td><td>States- | gen> ators-</td></tr>
<tr><td>T031</td><td>------- | ---- ------</td></tr>
<tr><td>T223</td><td>------- | ---- ---r--</td></tr>
<tr><td>Transcript</td><td>States | Sen- ators</td></tr>
</table>

Figure 8.6: An example of an aligned lattice from document CT_2Apr1872_p2_c1. The "dash" character represents a gap or *INDEL* in the alignment, where a character needs to be inserted in order to complete the alignment. Correct hypotheses in the binarized text are underlined. The aligned text is divided into columns of hypotheses in which column boundaries are where all aligned sequences agree on a space character.

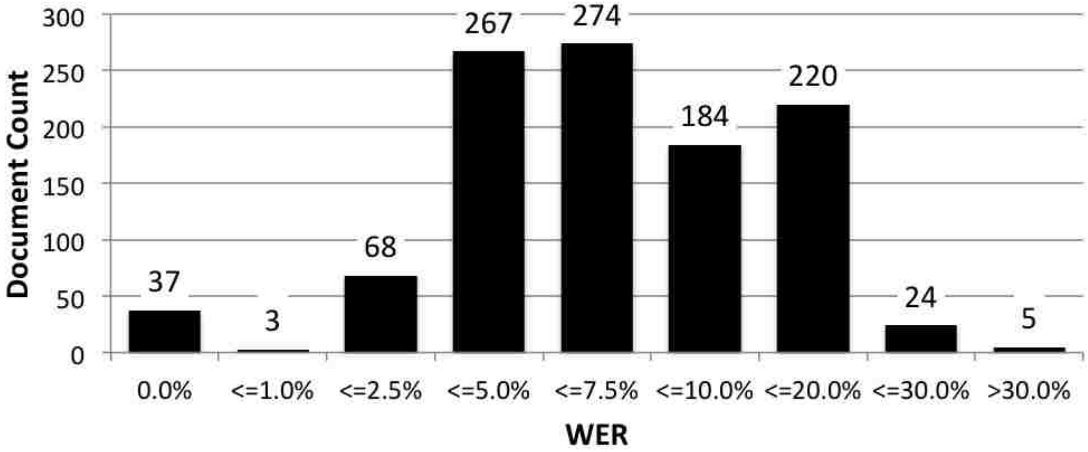


Figure 8.7: The distribution of the lowest WER binarization for all documents.

images with varying binarization threshold values. In fact there is a direct relationship between the percentage of correct tokens found in higher WER binarizations and the WER of the best binarization.

8.4.1 High WER Contributions

The primary question addressed in this paper is: how broadly was the useful information distributed across all of the binarizations? As one lower bound on WER, we evaluated all seven binarizations of a document image and selected (with reference to truth) the binarization with the lowest WER for that document: the average WER of the corpus, using only the individual document binarization with the lowest WER for each document, is 7.70% (see Table 8.2, row 4). This result is an oracle calculation since the binarization having the lowest WER for each document is known only by referring to a gold transcript. In Figures 8.7 and 8.8 we see the distribution of the lowest WER for all documents. For example, in Figure 8.7, there were 267 documents, for which the WER of the best binarization ranged between 2.5% and 5%. More importantly, Figure 8.8 shows the number of documents for each threshold value, where that threshold provided the binarization with the lowest WER. The binarization threshold value of 127, the mid-point between 0 and 255, had the most instances of the lowest WER of all of the binarizations of a document. Note, however,

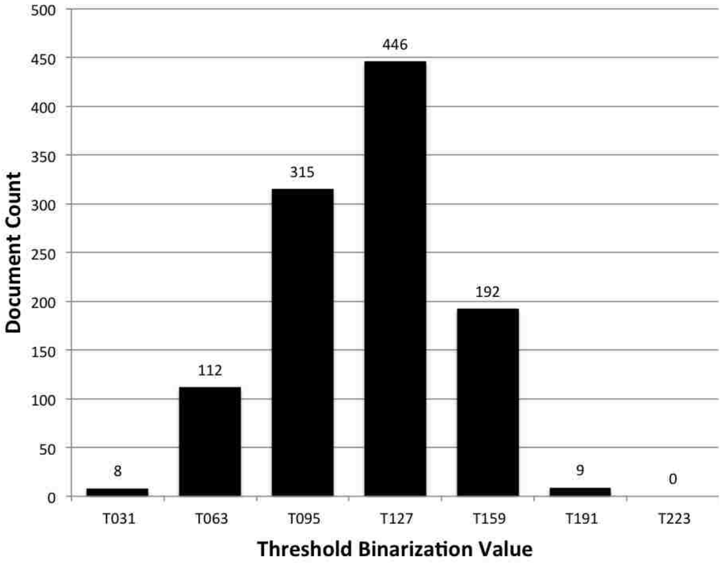


Figure 8.8: The distribution of the binarization thresholds with the lowest WER for each document.

that the other threshold values when summed have more instances of a lowest WER binarization than those at threshold value 127. Further note that extreme threshold values (31, 63, and 191) include a significant number of "best OCR" binarizations as defined by the resulting WER. Since the contribution of the extreme threshold values (31 and 223) was not known prior to the process described here, they were included in the evaluation.

This result is an indication that significant information useful to the OCR error correction task is to be found outside the best overall binarization threshold value. This distribution of lowest OCR WER across almost the full range of binarization threshold values is an indication that selecting a single value for the entire corpus will eliminate data that would contribute to a lower OCR WER. The overall result demonstrates the degree to which correct information is distributed across many threshold binarization values.

Figure 8.9 compares the spread of WERs across all binarization threshold values. For example, the column labeled "lowest WER" shows the spread of WERs for the binarization threshold value with lowest WER of each document. Note that for each document image the threshold binarization value that resulted in lowest OCR WER is not necessarily the same, as seen in Figure 8.8. For the first column in Figure 8.9 the minimum WER found among all "lowest

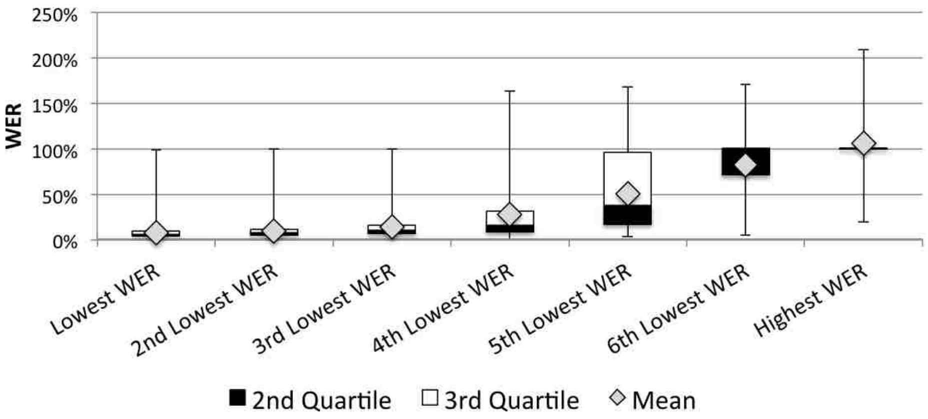


Figure 8.9: The distribution of WERs for each binarization, ordered by increasing WER, not by threshold value. (Note that WERs greater than 100% are possible due to insertion errors which are not limited by the words in a document.)

WER binarization threshold values" for all document images is 0.00%, the first quartile is 4.2%, the median is 6.46%, the third quartile is 9.75%, and the maximum is 98.68%. Compare the first column to the second column, representing the document binarization thresholds with second lowest WER, and the results are similar. Likewise the third column shows similar results, confirming the availability of useful information beyond the best binarization threshold value. It is important to note that binarizations with reasonably low WERs and useful information for OCR error correction can be found across the range of binarization threshold values.

Even the threshold values that result in the binarizations with the highest WER provide information not found in the binarization with the lowest WER, as seen in Table 8.3. These results show that unique information is available that can contribute in reducing the OCR WER from all binarizations of the original grayscale image. Across the entire corpus, even the lowest quality binarizations, those with the highest WERs, can contribute 148 out of 7716 instances of word tokens that were not correctly recognized in the binarized image with the lowest WER. Overall, the binarizations with a higher WER can contribute 6277 word tokens that were not correctly recognized in the lowest WER binarization, for a total of 2.682% of the total correctly identified tokens across all binarizations for all document images. This result can be interpreted to say that in selecting only the best WER threshold binarization value, 2.682% possible correct selections would

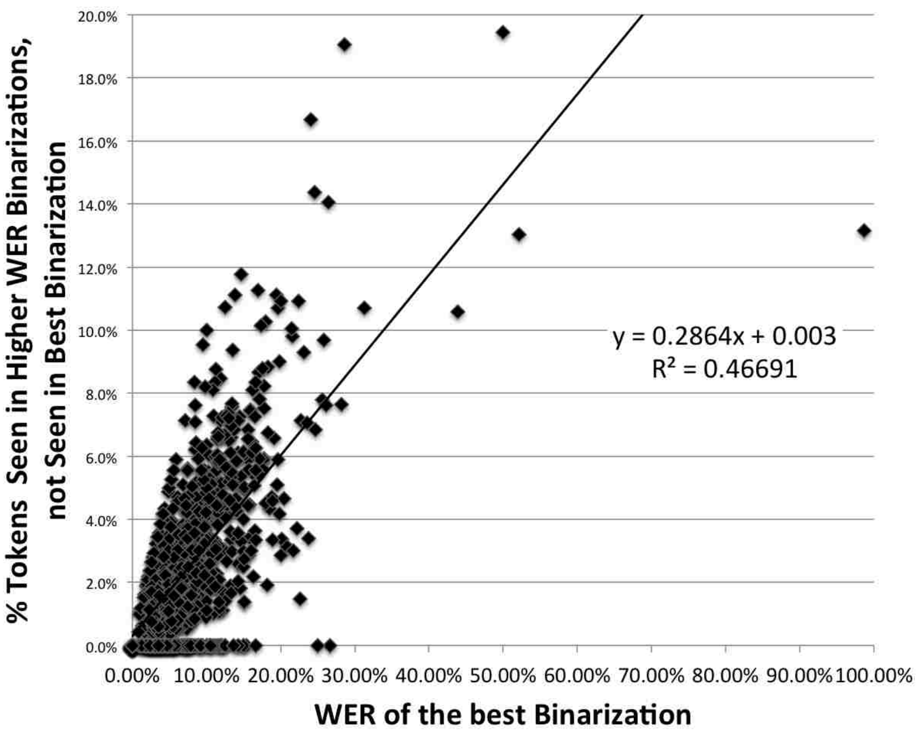


Figure 8.10: The percent of tokens seen in the OCR of binarizations with a higher WER, which are not seen in the best binarization. The line is the least squares fit of the data.

be lost from the other binarizations. Another point, also from Table 8.3, is that only 0.615% of the correctly identified tokens were unique to the binarization with the lowest WER.

8.4.2 Value of Multiple Binarizations

For each document in the corpus, we compute the percentage of correct OCR tokens from each binarization of that document which were not seen in the best binarization. These results are presented in Figure 8.10 and constitute one of the most important results from this study. The figure shows that as the WER (x) of the best binarization increases, the percentage (y) of correct words found in other binarizations increases. The least squares fit of this relationship is $y = 0.2864x + 0.003$ with an R^2 of 0.46691. As the document image becomes noisier (shown as a higher WER), the more “room” there is for multiple binarizations to contribute to the overall error correction of the document. Table 8.3 clarifies this result, displaying the number of tokens correctly identified in the binarizations the threshold values of which ordered by decreasing WER. Across

Table 8.3: Number of tokens correctly identified in higher WER binarizations, that were not correctly identified in the lowest WER binarizations.

Relative WER	**# of Tokens**	**% of Total Correct**
highest WER	148	0.063%
2nd highest WER	657	0.281%
3rd highest WER	1430	0.611%
4th highest WER	1693	0.723%
3rd lowest WER	1434	0.613%
2nd lowest WER	915	0.319%
Total	*6277*	*2.682%*
Unique to Lowest WER	1439	0.615%

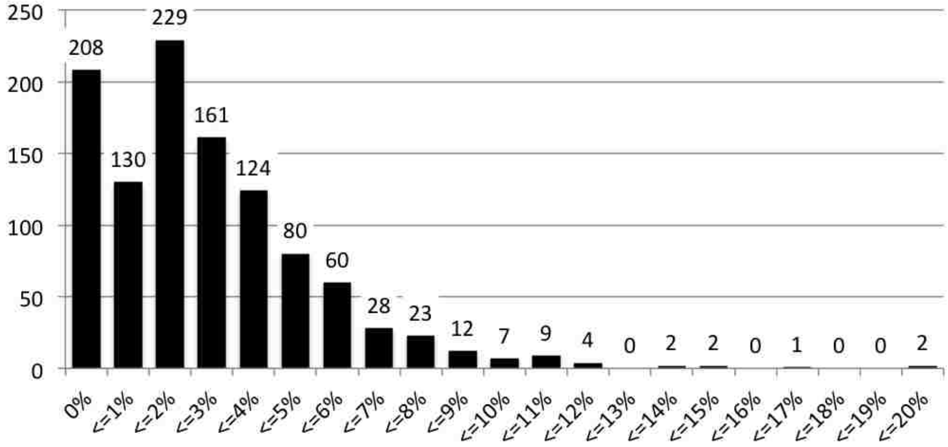


Figure 8.11: The number of documents for which a given percentage of correct word tokens occur in a higher WER binarization but are missing from the lowest WER binarization.

the entire corpus 2.682% of correct words would not have been seen had the OCR of binarizations with a higher WER been excluded. This result is compounded with the uncertainty about which threshold will yield the best binarization depicted in Figure 8.8.

Figure 8.11 shows the distribution of documents by the percentage of correct word tokens which are seen in a higher WER binarization, that are not seen in the lowest WER binarization. For example, in 229 documents (bar #3) there were between 1% and 2% of the words which appeared in higher WER binarizations, but did not appear in the lowest (or best) WER binarization. Note that for a significant number of documents, the percentage of words which were recognized in higher WER binarizations was higher than 5%. This is another indication that using multiple binarizations results in reduced WERs of a combined output.

8.5 Conclusions and Future Work

This research has shown the extent to which information is spread across multiple binarizations of the same document image, which can be leveraged to improve the WER of an aligned and combined output. In the documents of this corpus 2.682% of the word tokens were not recognized in the document binarization with the lowest WER, but were recognized in document binarizations with a higher WER. Not considering these additional binarizations affects the overall WER possible. Further, considering only the binarizations with a higher WER effectively reduces the overall WER by only 0.615%. Using all binarizations provides the best possible result.

Lastly, this study shows the extent to which binarizations with higher OCR WERs can be expected to contribute to the correction of OCR as the WER increases.

Future work will include scaling a fixed number of threshold values to fall between the white and black modes (as described in Section 8.3.2) on a document-by-document basis. As the white and black modes, seen in Figures 8.4 and 8.5, are not the same for each document image, we believe that dividing the region between those modes would be more effective in capturing threshold values likely to provide the most information.

Reducing the time complexity of this process was not a primary goal of this research effort. The quality of the OCR clearly affects the end result; consequently, this research used Abbyy FineReader 10, which has been shown in previous research to provide the lowest WER on these historical documents. This OCR engine supports batch processing, but if desired other OCR engines or versions of this OCR engine which support an API could also be called programmatically. Future work is planned to streamline the entire process for use in large scale OCR efforts.

8.6 Addendum

An important point that was not made in the original published paper is that the order of progressive alignment enables the use of high error rate sequences. By starting the progressive alignment with higher quality sequences, the lower quality sequences have a basis upon which to build. If the alignment started with the lowest quality sequences alignment errors would occur due to the text errors which would then be propagated through the remaining progressive alignments of the higher quality sequences.

Chapter 9

Conclusion

This research has shown novel methods for reducing OCR word error rates, along with the evaluation of how diversity of input sources contributes to the overall improvement. The focus of this work has been on degraded historic printed documents of which two corpora have been developed based on documents from the L. Tom Perry Special Collections of the Harold B. Lee Library from Brigham Young University. Additionally, this research provides a new admissible heuristic for use with the A* minimum path algorithm applied to finding the minimum cost exact alignment of multiple text sequences.

Chapter 3 introduces the core concept of this research; using multiple derivatives of the same document to extract information used to correct OCR. In this chapter the method align the output of three OCR engines using the Reverse Dijkstra Heuristic, a novel admissible heuristic used with the A* exact alignment algorithm. From the aligned OCR output, the method defines the lattice word error rate as an oracle calculation that shows the lowest possible WER given perfect knowledge. This result shows that there is significant information scattered among all the OCR engine outputs. Applying simple voting among token hypotheses from each OCR engine and comparing with a dictionary and gazetteer, the method reduces the average document WER below that of any of the OCR engine results. This chapter also introduces the Eisenhower Communiqués corpus, a collection of effectively bitonal typewritten documents from 1944 and 1945 as a test set. Lastly, this chapter defines the Reverse Dijkstra Heuristic for use with the A* minimum path algorithm in finding the minimum cost alignment of multiple text sequences.

Chapter 4 splits the same corpus as Chapter 3 into test and training sets and explores the use of a trained decision tree in decided which token in the aligned hypothesis column should be selected. This method was an improvement over the naïve approach in Chapter 3 and showed that in some cases the error corrected text equaled the lattice word error rate of the document.

Chapter 5 shows the degree to which multiple OCR engines, even those with high WER outputs, can contribute to correcting the OCR. This chapter shows that as text sequences from additional OCR engines are added to the aligned lattice the lattice WER decreases. This chapter also introduces the use of a progressive heuristic alignment, rather than the exact alignment used in Chapters 3 and 4, which accommodates adding two new OCR engines. This chapter introduces a new synthetic training set based on the a subset of the 2001 Annotated Enron Email Data Set.

Chapter 7 shows an alternate approach to using multiple OCR engines. Rather than using multiple OCR engines on the same digitized document image, a single document image is binarized using various threshold values. This method is seen to be as effective as using multiple OCR engines. This chapter also introduces a new corpus, the Nineteenth Century Mormon Article Newspaper Index.

Chapter 8 focuses on identifying the basis for the OCR error correction observed in Chapter 7. One conclusion was that the sources of error correcting information were scattered among all of the document image binarizations and that a significant number of improvements were found in document image binarizations which themselves did not have a low WER relative to the other binarizations of that same document image. Selecting only the single best binarization of a document image, which itself would be an oracle calculation requiring perfect knowledge, was an inferior approach to the methods shown here. Lastly, the results showed that there is a positive relationship between the overall WER of the binarized document image and the amount of error correcting information available across all binarizations.

Chapter 6 generalizes the results from the previous chapters to see if the successful methods used previously will apply across both test corpora and with a new synthetic training set. Although the methods used still provided some benefit, the 19th Century Mormon Article Newspaper Index corpus appeared to be more sensitive to high WER OCR output included in the training set. Showing that the training sets can be smaller than the full training corpus of hundreds of thousands of feature vectors, it may be possible to identify documents in the training set with high WER OCR output and eliminate them. This chapter also introduces a new synthetic training set based on the Reuters 21578 database from LDC, which uses grayscale rather than bitonal images.

Chapter A explores the potential of space propagation within a hypothesis lattice to identify places in the text sequences where spaces may have been deleted by the OCR engines. Unfortunately, for the corpora used in this research, there appears to be very little benefit to this approach.

9.1 Future Work

Within the scope of ensemble methods and diversity, this work has contributed to a specific area, the error correction of optical character recognition, whereas ensemble methods with diverse inputs are applicable in a variety of tasks, such as image recognition, voice recognition, and handwriting recognition to name a few. Regarding diversity, what are the properties that make this fruitful in an ensemble method? Clearly one principle is that diverse correct inputs to an ensemble method give the method more correct information to work with. As discussed in Chapter 8, even low quality inputs which include unique correct input, contribute to error correction. This observation may be applicable in other tasks using ensemble methods.

Specifically related to OCR error correction, future directions may include the following.

The binarization used in Chapters 7 and 8 was a simple threshold binarization and the state-of-the-art in binarization is adaptive rather than the simple thresholding. Although we demonstrated in Chapter 7 that for our purposes our methods are superior to two of the current adaptive binarization algorithms, it would be possible to apply multiple adaptive binarizations, varying the parameters of the algorithm. In parallel with this it would be useful to explore scaling simple binarization thresholds between the black and white modes, as shown in Figures 8.4 and 8.5 rather than scaling the thresholds between the minimum and maximum pixel grayscale values.

The Google Book n-gram corpus provides an opportunity to compare the hypotheses against known n-grams. As a feature in the conditional random field, this would contribute to measuring the work hypotheses against a language model.

From the perspective of the alignment, the current method in this research does not account for "near misses." Using a confusion matrix derived from the output of the OCR on a test set, it would be possible to devise alignment costs that would differentiate between an "e" being confused with an "o" and an "X".

The computational complexity of these methods needs to be reduced so that they can be applied in real time. Although the use of the Fulton Super Computing Lab has been important to this research, application of these methods in the Lee Library would require being able to run them on common servers. Ultimately, the methods from this research will be applied to the processes of the Digital Imaging Lab of the Lee Library, where we have tens of millions of pages of rare print documents in the L. Tom Perry Special Collections that could conceivably be digitized and made available electronically.

Appendix A

Effects of Space Propagation

A.1 Introduction

Section 1.3.7 mentions the possibility of using space propagation to identify additional token hypotheses from within a hypothesis column.

Space propagation includes propagating both spaces and closures across all aligned sequences. This is a point where character alignment is particularly important to be able to see where spaces may be applied in the middle of tokens. For this document's purposes a column which includes a space internally will be called a "fat column."

A.2 Method

The method behind space and closure propagation relies on the observation that on occasion an aligned sequence may in itself be incorrect, but the location of a space in the incorrect sequence may be accurate. An example of this is shown in Figures A.1 and A.2, a fat column found in the document AEM_24Jul1844_p3_col1 from the 19thCMAN test dataset. Note in Figure A.1 that none of the OCR sequences contain the correct text, and that the OCR output from Adobe contains a space in the correct location, but that the text on either side of the space is incorrect. The method propagates the space found in the Adobe OCR output to the other OCR outputs in the same location found in Adobe. Effectively this splits the fat column into two normal hypothesis columns as seen in Figure A.2. Given the two columns split upon the location of the space in the Adobe sequence, both Abbyy and Omnipage now match the transcription text for the first word "his". This is an

OCR Engine	OCR Output
Transcript	-his wa--y--
Adobe	.his ;-fiy;.
Tesseract	-lihywa--y--
Abbyy	-his,wa--y--
Omnipage	-his,wa--y--

Figure A.1: An example in AEM_24Jul1844_p3_col1 from the 19thCMAN dataset in which none of the OCR outputs provide correct text. The "dash" character represents and INDEL in the alignment.

OCR Engine	OCR Output		Without INDEL	
Transcript	-his	wa--y--	his	way
Adobe	.his	;-fiy;.	.his	;fiy;.
Tesseract	-lih	ywa--y--	lih	yway
Abbyy	-his	,wa--y--	his	,way
Omnipage	-his	,wa--y--	his	,way

Figure A.2: An example in AEM_24Jul1844_p3_col1 from the 19thCMAN dataset in which the space found in the Adobe OCR output is propagated to other OCR outputs in the aligned column.

improvement over the original state of the fat column which would not have provided any correct text.

A.3 Results

From the previous section it is clear that there are instances (at least one) in which space propagation can provide a way to improve the OCR error corrected output. Responding to the question on how much improvement is possible, Table A.1 shows the results from all four datasets used in this work. These results are an oracle calculation in which if a correct response is possible, then it is counted as an improvement. The machine learning techniques used in Chapters 4, 5, 7, and 8 would not be guaranteed to correctly identify the space propagation results as correct.

A.4 Conclusions

Although the notion of propagating spaces between aligned sequences appeared to have promise, the results appear very limited based on two observations. First, the lattice results shown in Table A.1, which are an oracle calculation assuming perfect knowledge, show that the overall improvement in

Table A.1: Lattice results of applying space and closure propagation

Dataset	Total Hypothesis Columns	Total Fat Columns	Dataset Tokens	Columns Improved By Propagation	Percent Improvement
			Test Sets		
19thCMAN	199,907	16,683	211,541	16	0.0076%
Eisenhower	79,395	10,816	145,346	28	0.0193%
			Training Sets		
Enron	138,749	33,157	240,850	42	0.0174%
Reuters21578	75,677	30,892	590,674	4	0.0007%

the two test sets, Eisenhower and 19thCMAN, is minimal. Second, and perhaps more importantly, the availability of training examples that correctly identify instances where space propagation provided correct results are also very limited.

Based on these results, space propagation will not be pursued further in this research.

References

[1] AACR. Homepage of the Anglo-American Cataloguing Rules, 2006. URL `http://www.aacr2.org`.

[2] Ahmad Abdulkader and Matthew R. Casey. Low cost correction of OCR errors using learning in a multi-engine environment. In *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR 2009)*, pages 576–580, Barcelona, Spain, July 2009.

[3] Jerome Ajot, Jon Fiscus, Nicolas Radde, and Chris Laprun. Asclite – Multi-dimensional alignment program., 2008. URL `http://www.nist.gov/speech/tools/asclite.html`.

[4] A. Antonacopoulos and D. Karatzas. Semantics-based content extraction in typewritten historical documents. In *Proceedings of the 8th International Conference on Document Analysis and Recogniction, 2005*, volume 1, pages 48–53, Seoul, South Korea, August 2005. ISBN 0-7695-2420-6. doi: 10.1109/ICDAR.2005.215.

[5] Kevin Atkinson. GNU Aspell, 2008. URL `http://aspell.net/`.

[6] Kartik Audhkhasi, A. Zavou, P. Georgiou, and S. Narayanan. Theoretical analysis of diversity in an ensemble of automatic speech recognition systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 2014.

[7] Henry Baird. The state of the art of document image degradation modeling. In *Digital Document Processing*, pages 261–279. Springer, 2007.

[8] Michael W. Berry, Murray Browne, and Ben Signer. 2001 topic annotated Enron email data set, June 2007. URL `http://www.ldc.upenn.edu/`.

[9] Roman Bertolami and Horst Bunke. Ensemble methods for handwritten text line recognition systems. In *Proccedings of the 2005 IEEE International Conference on Systems, Man and Cybermetrics*, Waikoloa, Hawaii, October 2005.

[10] Federico Boschetti, Matteo Romanello, Alison Babeu, David Bamman, and Gregory Crane. Improving OCR accuracy for classical critical editions. In *Proceedings of the 13th European*

Conference on Research and Advanced Technology for Digital Libraries (ECDL 2009), Corfu, Greece, September 2009. Springer Verlag.

[11] Thomas M. Breuel. The OCRopus open source OCR system. In *Proceedings of Document Recognition and Retrieval XV (DRR 2008)*, volume 6815, 2008. doi: 10.1117/12.783598.

[12] Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 286–293, Hong Kong, 2000. Association for Computational Linguistics.

[13] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18, 2004.

[14] H. Cecotti and A. Belaid. Hybrid OCR combination approach complemented by a specialized ICR applied on ancient documents. In *Proceedings of the 8th International Conference on Document Analysis and Recognition, 2005*, volume 2, pages 1045–1049, Seoul, South Korea, August 2005. ISBN 0-7695-2420-6. doi: 10.1109/ICDAR.2005.130.

[15] Daniel Cer, Christopher D. Manning, and Dan Jurafsky. Positive diversity tuning for machine translation system combination. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 320–328, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

[16] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 173–180, Ann Arbor, MI, June 2005.

[17] Michael Collins and Terry Koo. Discriminative reranking for natural language parsing. *Association for Computational Linguistics*, 31(1):25—70, 2005. ISSN 0891-2017. doi: 10.1162/0891201053630273.

[18] David B. Curtis and Shawn Reid. Multple image input for optical character recognition processing systems and methods, June 2010. Patent Number: 7,734,092 B2.

[19] Rina Dechter and Judea Pearl. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM*, 32(3):505–536, 1985. doi: 10.1145/3828.3830.

[20] John DeNero, Shankar Kumar, Ciprian Chelba, and Franz Och. Model combination for machine translation. In *Proceedings of Human Language Technologies: The 2010 Annual*

Conference of the North American Chapter of the ACL, volume 1, pages 975–983, Los Angeles, Calif., June 2010.

[21] Thomas G. Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*. Springer, 2000.

[22] Dejan Dordevic and Dragan Mihajlov. Weighted voting of multiple ALN classifiers for OCR. In *Proceedings of the 20th International Conference Information Technology Interfaces*, pages 87–92, Pula, Croatia, June 1998.

[23] Isaac Elias. Settling the intractability of multiple alignment. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, 13(7):1323–39, September 2006. ISSN 1066-5277. doi: 10.1089/cmb.2006.13.1323.

[24] Jeffrey Esakov, Daniel Lopresti, Jonathan Sandberg, and Jiangying Zhou. Issues in automatic OCR error classification. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, pages 401–412, Las Vegas, NV, April 1994.

[25] Jeffrey Esakov, Daniel P. Lopresti, and Jonathan Sandberg. Classification and distribution of optical character recognition errors. In *Proceedings of IS&T/SPIE International Symposium on Electronic Imaging*, pages 204–216, San Jose, CA, February 1994.

[26] P. F. Felzenszwalb and D. McAllester. The generalized A* architecture. *Journal of artificial Intelligence Research*, 29:153–190, 2007.

[27] Shaolei Feng and R. Manmatha. A hierarchical, HMM-based automatic evaluation of OCR accuracy for a digital library of books. In *Proceedings of the 6th Annual ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 109–118, Chapel Hill, NC, June 2006. doi: 10.1145/1141753.1141776.

[28] J.G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 347–354, 1997. doi: 10.1109/ASRU.1997.659110.

[29] Patricia Fraud. 19th century Mormon article newspaper index. L. Tom Perry Special Collections, Brigham Young University, 2012. URL `http://lib.byu.edu/digital/19cMormonArticles/`.

[30] B Gatos, I Pratikakis, and S. J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, 39:317–327, 2006.

[31] B Gatos, I Pratikakis, and S. J. Perantonis. Efficient binarization of historical and degraded document images. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, pages 447–454, September 2008.

[32] Kevin Gimpel, Dhruv Batra, Chris Dyer, and Gregory Shakhnarovich. A systematic exploration of diversity in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, Seattle, Washington, USA, October 2013.

[33] V. Goel, S. Kumar, and W. Byrne. Segmental minimum bayes-risk decoding for automatic speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 12(3):234–249, 2004. ISSN 1063-6676. doi: 10.1109/TSA.2004.825678.

[34] M. Govindarajan. Evaluation of ensemble classifiers for handwriting recognition. *International Journal of Modern Education & Computer Science*, 5(11), 2013.

[35] John C. Handley and Thomas B. Hickey. Merging optical character recognition outputs for improved accuracy. In *RIAO 91: Computer aided information retrieval. Conference*, December 1990.

[36] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(10):993–1001, 1990.

[37] L.K. Hansen, C. Liisberg, and P. Salamon. Ensemble methods for handwritten digit recognition. In *Neural Networks for Signal Processing [1992] II., Proceedings of the 1992 IEEE-SP Workshop*, pages 333–342, 1992. doi: 10.1109/NNSP.1992.253679.

[38] J.J. Hull. Incorporating language syntax in visual text recognition with a statistical model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(12):1251–1255, 1996. ISSN 0162-8828. doi: 10.1109/34.546261.

[39] T. Ikeda and T. Imai. Fast A* algorithms for multiple sequence alignment. In *Proceedings of Genome Informatics Workshop 1994*, Yokohama, Japan, 1994. Universal Academy Press.

[40] H. Imai and T. Ikeda. k-group multiple alignment based on A* search. In *Proceedings of the 6th Genome Inform. Workshop*, pages 9–18, 1995.

[41] Rong Jin, Alex G Haupmann, and Chengxiang Zhai. A content-based probabilistic correction model for OCR document retrieval. Technical Report 1-1-2003, Carnegie Mellon University, 2002. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.4225`.

[42] Mark A. Jones, Guy A. Story, and Bruce W. Ballard. Integrating multiple knowledge sources in a bayesian OCR post-processor. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 925–933, 1991.

[43] David Reed Jordan. Daily battle communiques, 1944-1945. Harold B. Lee Library, L. Tom Perry Special Collections, MSS 2766, 1945. URL `http://www.lib.byu.edu/digital/eisenhower/`.

[44] Andrew Kae and E.G. Learned-Miller. Learning on the fly: Font-free approaches to difficult OCR problems. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2009*, Barcelona, Spain, 2009.

[45] Paul B. Kantor and Ellen M. Voorhees. The TREC-5 confusion track: Comparing retrieval methods for scanned text. *Information Retrieval*, 2(2-3):165–176, 2000.

[46] T. Kanungo and R.M. Haralick. An automatic closed-loop methodology for generating character groundtruth for scanned documents. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(2):179–183, 1999. ISSN 0162-8828. doi: 10.1109/34.748827.

[47] I.-K. Kim, D.-W. Jung, and R.-H. Park. Document image binarization based on topographic analysis using a water flow model. *Pattern Recognition*, 35:265–277, 2002.

[48] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, March 1998. ISSN 0162-8828. doi: 10.1109/34.667881.

[49] S. T Klein and M. Kopel. A voting system for automatic OCR correction. In *Proceedings of the SIGIR 2002 Workshop on Information Retrieval and OCR*, August 2002.

[50] Vladimir Kluzner, Asaf Tzadok, Yuval Shimony, Eugene Walach, and Apostolos Antonacopoulos. Word-based adaptive OCR for historical books. In *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR 2009)*, page 501, Barcelona, Spain, July 2009.

[51] Okan Kolak and Philip Resnik. OCR error correction using a noisy channel model. In *Proceedings of the second international conference on Human Language Technology Research*, pages 257–262, San Diego, California, 2002. Morgan Kaufmann Publishers Inc.

[52] Okan Kolak, William J. Byrne, and Philip Resnik. A generative probabilistic OCR model for NLP applications. In *Proceedings of HLT-NAACL 2003*, pages 55–62, Edmonton, Canada, May 2003.

[53] Karen Kukich. Techniques for automatically correcting words in text. *ACM Comput. Surv.*, 24(4):377–439, 1992. doi: 10.1145/146370.146380.

[54] Eugene Lawler. Networks with positive arcs: Dijkstra's method. In *Combinatorial optimization: Networks and Matroids*, pages 70–73. Holt, Reinhart and Winston, 1 edition, 1976. ISBN 0-03-084866-0.

[55] D. D. Lewis. Reuters-21578. Test Collection, 2013. URL `http://www.daviddlewis.com/resources/testcollections/reuters21578/`.

[56] Xiaofan Lin. Reliable OCR solution for digital content re-mastering. In *Proceedings of the SPIE Conference on Document Recognition and Retrieval IX*, San Jose, CA, January 2002.

[57] Rafael Llobet, J. Ramon Navarro-Cerdan, Juan-Carlos Perez-Cortes, and Joaquim Arlandis. OCR post-processing using weighted finite-state transducers. In *Proceedings of the 2010 International Conference on Pattern Recognition*, pages 2021–2024, Istanbul, Turkey, August 2010. doi: 10.1109/ICPR.2010.498.

[58] Daniel Lopresti. Optical character recognition errors and their effects on natural language processing. In *Proceedings of the second workshop on Analytics for noisy unstructured text data*, pages 9–16, Singapore, 2008. ACM. ISBN 978-1-60558-196-5. doi: 10.1145/1390749.1390753.

[59] Daniel Lopresti and Jiangying Zhou. Using consensus sequence voting to correct OCR error. *Computer Vision and Image Understanding*, 67(1):39–47, 1997.

[60] William Lund. Exploring the multiple sequence alignment of text using software designed for genetic alignment. Unpublished manuscript, December 2009.

[61] William B. Lund and Eric K. Ringger. Improving optical character recognition through efficient multiple system alignment. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 231–240, Austin, TX, USA, 2009. ACM. ISBN 978-1-60558-322-8. doi: 10.1145/1555400.1555437.

[62] William B. Lund and Eric K. Ringger. Feature engineering across multiple OCR outputs to improve optical word recognition. Unpublished manuscript, February 2010.

[63] William B. Lund and Eric K. Ringger. Error correction with in-domain training across multiple OCR system outputs. In *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011)*, Beijing, China, September 2011.

[64] William B. Lund, Daniel D. Walker, and Eric K. Ringger. Progressive alignment and discriminative error correction for multiple OCR engines. In *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011)*, Beijing, China, September 2011.

[65] William B. Lund, Douglas J. Kennard, and Eric K. Ringger. Combining multiple thresholding binarization values to improve OCR output. In *Proceedings of Document Recognition and Retrieval XX*, San Francisco, California, February 2013.

[66] William B. Lund, Douglas J. Kennard, and Eric K. Ringger. Why multiple document image binarizations improve OCR. In *Proceedings of the Workshop on Historical Document Imaging and Processing 2013 (HIP 2013)*, Washington, DC, USA, August 2013.

[67] William B. Lund, Daniel D. Walker, and Eric K. Ringger. How well does multiple OCR error correction generalize? In *Proceedings of Document Recognition and Retrieval XXI (DRR 2014)*, San Francisco, California, February 2014.

[68] Huanfeng Ma and D. Doermann. Adaptive OCR with limited user feedback. In *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR 2005)*, volume 2, pages 814–818, Seoul, South Korea, 2005. ISBN 1520-5263. doi: 10.1109/ICDAR.2005.43.

[69] Wolfgang Macherey and Franz Josef Och. An empirical study on computing consensus translations from multiple machine translation systems. In *EMNLP-CoNLL*, page 986, 2007.

[70] Walid Magdy and Kareem Darwish. Arabic OCR error correction using character segment correction, language modeling, and shallow morphology. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 408–414, Sydney, Australia, July 2006. Association for Computational Linguistics.

[71] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, October 2000.

[72] Andrew Kachites McCallum. MALLET: A machine learning for language toolkit., 2002. URL `http://mallet.cs.umass.edu`.

[73] Sebastien Moretti, Fabrice Armougom, Iain M. Wallace, Desmond G. Higgins, Cornelius V. Jongeneel, and Cedric Notredame. The m-coffee web server: A meta-method for computing multiple sequence alignments by combining alternative alignment methods. *Nucl. Acids Res.*, 35(suppl_2):W645–648, July 2007. doi: 10.1093/nar/gkm333.

[74] Cosmin Munteanu, Gerald Penn, Ron Baecker, Elaine Toms, and David James. Measuring the acceptable word error rate of machine-generated webcast transcripts. In *Proceedings of the Ninth International Conference on Spoken Language Processing. INTERSPEECH 2006 (ICSLP 2006)*, page 1756, Pittsburgh, PA, September 2006.

[75] Yasuaki Nakano, Toshihiro Hananoi, Hidetoshi Miyao, Minoru Maruyama, and Ken-ichi Maruyama. A document analysis system based on text line matching of multiple OCR outputs. In *Document Analysis Systems VI*, pages 463–471. Springer-Verlag New York, Inc., 2004.

[76] NIST. NIST scientific and technical databases - NIST fed. reg. document image database: Vol.1, September 2009. URL `http://www.nist.gov/srd/nistsd25.htm`.

[77] Cedric Notredame. Recent progress in multiple sequence alignment: A survey. *Pharmacogenomics*, 3(1):131–144, November 2004. ISSN 1462-2416.

[78] Cdric Notredame. Recent evolutions of multiple sequence alignment algorithms. *PLoS Computational Biology*, 3(8):123, 2007. doi: 10.1371/journal.pcbi.0030123.

[79] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions of Systems, Man, and Cybernetics*, SMC-9(1):62–66, January 1979.

[80] Thomas L. Packer, Joshua F. Lutes, Aaron P. Stewart, David W. Embley, Eric K. Ringger, Kevin D. Seppi, and Lee S. Jensen. Extracting person names from diverse and noisy OCR text. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, AND '10, pages 19–26, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0376-7. doi: 10.1145/1871840.1871845.

[81] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Penn., July 2002.

[82] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley Pub. Co, Reading, Mass., 1984. ISBN 0201055945.

[83] Y. Rangoni, Faisal Shafait, and Thomas M. Breuel. OCR based thresholding. In *Proceedings of the IAPR Conference on Machine Vision Applications*, pages 98–101, Yokohama, Japan, May 2009.

[84] Anupama Ray, Ankit Chandawala, and Santanu Chaudhary. Character recognition using conditional random field based matching engine. In *Proceedings of the 12th International*

Conference on Document Analysis and Recognition, pages 18–22, Washington, DC, USA, August 2013.

[85] Martin Reynaert. Non-interactive OCR post-correction for giga-scale digitization projects. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, number 4919 in Lecture Notes in Computer Science, pages 617–630. Springer Berlin Heidelberg, January 2008. ISBN 978-3-540-78134-9, 978-3-540-78135-6.

[86] Eric Ringger. *Correcting Speech Recognition Errors*. Dissertation, University of Rochester, 2000.

[87] Eric K. Ringger and James F. Allen. A fertility channel model for post-correction of continuous speech recognition. In *Fourth International Conference on Spoken Language Processing (ICSLP'96)*, Philadelphia, PA, October 1996.

[88] R. L Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

[89] David H. Rose and Anne Meyer. *Teaching Every Student in the Digital Age: Universal Design for Learning*. Association for Supervision & Curriculum Development, April 2002. ISBN 0871205998.

[90] Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. Combining outputs from multiple machine translation systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 228–235, Rochester, New York, April 2007. Association for Computational Linguistics.

[91] Prateek Sarkar, Henry S. Baird, and Xiaohu Zhang. Training on severely degraded text-line images. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 1*, page 38. IEEE Computer Society, 2003. ISBN 0-7695-1960-1.

[92] J. Sauvola and M. Pietikinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, February 2000. ISSN 0031-3203. doi: 10.1016/S0031-3203(99)00055-2.

[93] Stefan Schroedl. An improved search algorithm for optimal multiple-sequence alignment. *Journal of artificial Intelligence Research*, 23(January/June 2005):587–623, 2005. ISSN 1076-9757.

[94] Mehmet Sezgin and Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, January 2004. ISSN 1017-9909. doi: 10.1117/1.1631315.

[95] Luo Si, Tapas Kanungo, and Xiangji Huang. Boosting performance of bio-entity recognition by combining results from multiple systems. In *Proceedings of the 5th international workshop on Bioinformatics*, pages 76–83, Chicago, Illinois, 2005. ACM. ISBN 1-59593-213-5. doi: 10.1145/1134030.1134044.

[96] R. Smith. An overview of the tesseract OCR engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633, Parana, Brazil, September 2007. ISBN 978-0-7695-2822-9. doi: 10.1109/ICDAR.2007.4376991.

[97] Matthew Spencer and Christopher Howe. Collating texts using progressive multiple alignment. *Computers and the Humanities*, 38(3):253–270, August 2004. ISSN 0010-4817. doi: 10.1007/s10579-004-8682-1.

[98] Andreas Stolcke, Yochai Knig, and Mitchel Weintraub. Explicit word error minimization in n-best list rescoring. *PROC. EUROSPEECH*, pages 163—166, 1997.

[99] Christian M. Strohmaier, Christoph Ringlstetter, and Klaus U. Schulz. Lexical postcorrection of OCR-Results: The web as a dynamic secondary dictionary? In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03)*, Edinburgh, Scotland, August 2003.

[100] Charles Sutton and Andrew Kachites McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012. doi: 10.1561/2200000013.

[101] G. Thoma and D. Le. Medical database input using integrated OCR and document analysis and labeling technology. In *Proceedings 1997 Symposium on Document Image Understanding Technology*, page 280, 1997.

[102] George R. Thoma. Automating the production of bibliographic records for MEDLINE. R&D report, Communications Engineering Branch, Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, MD, September 2001. URL `http://archive.nlm.nih.gov/pubs/thoma/mars2001.php`.

[103] Xiang Tong and David A. Evans. A statistical approach to automatic OCR error correction in context. In *Proceedings of the Fourth Workshop on Very Large Corpora*, page 88, Copenhagen, Denmark, August 1996.

[104] TREC. Text REtrieval conference (TREC) home page, August 2000. URL `http://trec.nist.gov/`.

[105] Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. Wider pipelines: N-best alignments and parses in MT training. In *Proceedings of The Eighth Conference of the Association for Machine Translation in the Americas*, pages 192–201, Waikiki, Hawai'i, October 2008.

[106] L. Vincent. Google book search: Document understanding on a massive scale. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition, 2007. ICDAR 2007.*, volume 2, pages 819–823, Parana, Brazil, September 2007. ISBN 978-0-7695-2822-9. doi: 10.1109/ICDAR.2007.4377029.

[107] Martin Volk, Lenz Furrer, and Rico Sennrich. Strategies for reducing and correcting OCR errors. In Caroline Sporleder, Antal van den Bosch, and Kalliopi Zervanou, editors, *Language Technology for Cultural Heritage*, Theory and Applications of Natural Language Processing, pages 3–22. Springer Berlin Heidelberg, January 2011. ISBN 978-3-642-20226-1, 978-3-642-20227-8.

[108] Daniel Walker, William B. Lund, and Eric K. Ringger. A synthetic document image dataset for developing and evaluating historical document processing methods. In *Proceedings of SPIE Volume 8297*, volume 8297, Burlingame, CA, January 2012.

[109] L Wang and T Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, 1(4):337–348, 1994. ISSN 1066-5277. doi: 8790475.

[110] David Wemhoener, Ismet Zeki Yalniz, and R. Manmatha. Creating an improved version using noisy OCR from multiple editions. In *Proceedings of the 12th International Conference on Document Analysis and Recognition*, pages 160–164, Washington, DC, USA, August 2013.

[111] Michael L. Wick, Michael G. Ross, and Erik G. Learned-Miller. Context-sensitive error correction: Using topic models to improve OCR. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR) 2007*, 2007.

[112] Tong Xiao, Jingbo Zhu, and Tongran Liu. Bagging and boosting statistical machine translation systems. *Artificial Intelligence*, 195:496–527, February 2013. ISSN 0004-3702. doi: 10.1016/j.artint.2012.11.005.

[113] Pingping Xiu and H.S. Baird. Towards whole-book recognition. In *Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop on*, pages 629–636, 2008. doi: 10.1109/DAS.2008.50.

[114] T. Yamazoe, M. Etoh, T. Yoshimura, and K. Tsujino. Hypothesis preservation approach to scene text recognition with weighted finite-state transducer. In *2011 International Conference on Document Analysis and Recognition (ICDAR)*, pages 359 –363, September 2011. doi: 10.1109/ICDAR.2011.80.

[115] C. Yan and G. Leedham. Decompose-threshold approach to handwriting extraction in degraded historical document images. In *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 239–244, October 2004.

[116] Yong Zhao and Xiaodong He. Using n-gram based features for machine translation system combination. In *Proceedings of NAACL HLT 2009*, pages 205–208, Bourlder, CO, June 2009.

[117] Jiangying Zhou and Daniel Lopresti. Repeated sampling to improve classifier accuracy. In *Proceedings of IAPR workshop on Machine Vision Applications*, pages 346–351, Kawasaki, Japan, December 1994.

[118] Y. Zhu, C. Wang, and R. Dai. Document image binarization based on stroke enhancement. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 955–958, August 2006.