Information Retrieval from CD Covers Using OCR Text

Padraig Kilkenny
B.A. (Mod.) CSLL
Final Year Project
Supervisor: Dr. Carl Vogel
3rd May, 2006
Declaration

I hereby declare that this thesis is entirely my own work and that it has not been submitted as an exercise for a degree at any other university.

___________________________  May 3, 2006
Padraig Kilkenny
Acknowledgements

Firstly, I would like to thank my supervisor Carl Vogel, for his endless support and guidance throughout the project as well as the use of his extensive CD collection!

I would also like to thank my parents for all their emotional and financial support throughout the degree.

Finally, I would like to thank the following people who provided a welcome release from the pressures of final year on countless occasions: Aoibheann, Colman, Cormac, Dave, Edwina, Eleanor, Emma, Hanan, Lieke, Malcolm and of course all the CSLLers (past and present).
“It is a very sad thing that nowadays there is so little useless information.”

*Oscar Wilde (1854 — 1900)*
## Contents

1 Introduction .................................................. 1
   1.1 Introduction ............................................. 1
   1.2 Aims ..................................................... 1
   1.3 Motivation ................................................. 1
   1.4 Overview .................................................. 2
   1.5 Final Introductory Remarks ................................. 3

2 Background Reading ........................................ 4
   2.1 Introduction .............................................. 4
   2.2 The Birth of Optical Character Recognition ............... 4
   2.3 Historical Review of OCR Research and Development ....... 5
       2.3.1 Template-Matching Methods ............................ 6
       2.3.2 Structure Analysis Method .............................. 8
       2.3.3 Slit/Stroke Analysis ................................... 9
       2.3.4 Hybrid of Template Matching and Structural Analysis 10
   2.4 Generations of Commercial OCR ............................. 10
   2.5 Challenges of OCR Processing ............................... 11
   2.6 Pre-Processing Techniques ................................ 13
       2.6.1 Character Recognition ................................ 13
       2.6.2 Connected Component Extraction ....................... 13
       2.6.3 Line Direction Determination .......................... 13
       2.6.4 Line Extraction ....................................... 13
       2.6.5 Character Matching ................................... 14
       2.6.6 Character Direction Determination ..................... 14
       2.6.7 Linguistic Processing .................................. 14
       2.6.8 String Segmentation ................................... 14
       2.6.9 Consistency Checking .................................. 15
       2.6.10 Stochastic Context Free Grammars .................... 15
       2.6.11 Structure Analysis .................................... 17
       2.6.12 Recognition of the Scanned Document ................ 18
       2.6.13 Document Understanding ............................... 18
   2.7 Post-Processing Techniques ................................ 19
       2.7.1 N-Gram Classification ................................ 19
       2.7.2 What is an n-gram? .................................... 19
       2.7.3 Generation N-Gram Frequency Profiles .................. 20
       2.7.4 Comparing and Ranking N-Gram frequency Profiles ..... 20
       2.7.5 Advantages of the N-Gram Frequency Technique ....... 21
       2.7.6 Probabilistic Information Retrieval Systems .......... 21
       2.7.7 Post-Processing System ................................. 22
## CONTENTS

2.7.8 Retrieval Methods for English-Text with Misrecognised OCR characters ........................................... 22
2.7.9 Confusion Matrix Retrieval (CMR) Method ................. 23
2.7.10 Expanded Confusion Matrix Retrieval (ECMR) Method 23
2.7.11 2-gram Matrix Retrieval System ............................ 23
2.8 Conclusion .......................................................... 24

### 3 Design

3.1 Introduction ......................................................... 26
3.2 Goals .............................................................. 26
3.3 System Architecture ................................................ 26
3.3.1 Acquiring the Image ........................................... 26
3.3.2 Flatbed Scanner ............................................... 26
3.3.3 Mobile Phone Device ......................................... 26
3.3.4 Image Pre-Processing ......................................... 27
3.3.5 Optical Character Recognition ............................ 27
3.3.6 Parse I .......................................................... 28
3.3.7 MyDatabase ................................................... 28
3.3.8 Language Model .............................................. 28
3.3.9 Extract Information from Database ....................... 28
3.3.10 Build Bigrams and Compare to Language Model ....... 28
3.3.11 Guess the CD Title ........................................ 28
3.3.12 Compare to FreeDB ......................................... 28
3.4 Why use the N-Gram Approach? ................................. 30
3.5 Conclusion ......................................................... 30

### 4 Pre-Processing

4.1 Introduction ....................................................... 31
4.2 Acquiring the Image ............................................. 31
4.2.1 Scanner ....................................................... 31
4.2.2 Mobile Phone ................................................ 32
4.3 Transferring the Image using a Scanner ....................... 32
4.4 Transferring the Image using a Mobile Phone ................ 33
4.4.1 Infrared ....................................................... 33
4.4.2 Bluetooth .................................................... 34
4.4.3 Mobile Phone Server ....................................... 34
4.5 Image Manipulation .............................................. 35
4.6 Optical Character Recognition ................................. 37
4.6.1 ABBYY FineReader 8.0 Professional ....................... 38
4.6.2 SimpleOCR .................................................. 38
4.6.3 GOCR ........................................................ 40
CONTENTS

4.6.4 Results .................................................. 40
4.7 Problems Still Encountered by OCR .......................... 41
4.8 Conclusion .................................................. 42

5 Implementation ................................................. 43
5.1 Introduction ................................................ 43
5.2 Results of the OCR Data .................................... 43
  5.2.1 Bad .................................................. 43
  5.2.2 Good ................................................ 45
  5.2.3 Excellent ............................................. 46
5.3 Why use Perl? .............................................. 48
  5.3.1 Regular Expressions ..................................... 48
  5.3.2 Perl DBI .............................................. 49
5.4 MySQL ..................................................... 50
5.5 Table Set up ................................................. 50
5.6 First Parse .................................................. 51
5.7 Getting the information from the Database .................. 52
5.8 Deciding on a Music Resource ............................... 53
  5.8.1 Compact Disc Database (CDDB) ......................... 53
  5.8.2 freeDB ............................................... 54
5.9 Problems with freeDB ....................................... 54
5.10 Language Model ............................................ 57
5.11 Bigrams .................................................... 57
5.12 Smoothing the Language Model .............................. 59
  5.12.1 Add-One Smoothing .................................... 59
  5.12.2 Witten-Bell Discounting ............................... 60
  5.12.3 Good-Turing Discounting .............................. 60
  5.12.4 Our Smoothing Technique ............................. 60
5.13 Calculating the Probability of the OCR Results .......... 62
5.14 Guess the Title ........................................... 65
  5.14.1 Defining the Threshold ................................ 65
5.15 Conclusion ................................................ 66

6 Results and Evaluation ......................................... 67
6.1 Introduction ............................................... 67
6.2 Analysis of Results ........................................ 67
  6.2.1 Recall ............................................... 67
  6.2.2 Precision ............................................. 68
6.3 Sample Input/Output ........................................ 68
6.4 Results of Scanned CD Covers ............................... 69
6.5 Results of Mobile Phone CD Covers ......................... 70
6.6 Results of freedb Data ........................................ 71
6.7 Discussion of Results ........................................ 71
6.8 Conclusion ..................................................... 72

7 Conclusion and Future Work .................................. 74
  7.1 Summary ..................................................... 74
  7.2 Achievements ................................................ 75
  7.3 Problems Encountered ...................................... 75
  7.4 Future Work ................................................ 76
    7.4.1 Improving the Parser Quality ......................... 76
    7.4.2 Full Integration with a Mobile Phone Server ....... 76
    7.4.3 Automation and User Interface ....................... 76
    7.4.4 Training the OCR Software .......................... 76
    7.4.5 Partial Matching ..................................... 77

References ....................................................... 78

A Code ............................................................ 80
B CDs Scanned from Scanner .................................... 81
C Pre-Processed CD Images from Scanner .................... 82
D OCR Data: CD Images from Scanner ....................... 83
E System Output: CD Images from Scanner .................. 98
F CDs Scanned from Mobile Phone ............................. 115
G OCR Data: CD Images from Mobile Phone .................. 116
H System Output: CD Images from Mobile Phone .......... 123
I Freedb Sample Tracks ....................................... 128
J System Output: Freedb Sample Tracks ..................... 138
K Tools Utilised ................................................ 148
## List of Figures

1. 2-D Reduction to 1-D by a slit ................................. 6
2. Peephole Method ................................................. 7
3. Application of Stochastic Context-Free Grammars to Business Card Labelling ........................................ 17
5. Layout and Logical Structure of the Document Model .......... 29
6. Epson Photo Perfection 2580 ........................................ 31
7. Sagem MyV-55 Camera Phone ........................................ 33
8. Example of Inverting CD Cover Colours .......................... 36
9. Application of the Median Filter ................................... 37
10. Conversion to Black and White ..................................... 37
11. Conversion to Grayscale ............................................ 38
12. Snapshot of Main ABBY Finereader Window ....................... 39
13. Snapshot of Main ABBY Finreader Window ....................... 40
14. Snapshot of Main OCR Window ..................................... 41
15. Sample Image Producing Bad OCR Results ....................... 44
16. Bad OCR Results .................................................. 44
17. Sample Image Producing Good OCR Results ..................... 45
18. Good OCR Results ................................................ 46
19. Sample Image Producing Excellent OCR Results ................ 47
20. Excellent OCR Results ............................................. 47
21. Communication between the Perl and a MySQL Database ....... 50
22. Entity Relation Diagram of MyDatabase .......................... 51
23. Successful result at Parse 1 ....................................... 52
24. Result of querying Song Database ............................... 53
25. Lowest Relative Frequency Value ................................. 59
26. Probability Formula ............................................... 61
27. \( n^{th} \) Root Probability ........................................ 63
28. Probability Word Results .......................................... 64
29. Probability Word Results .......................................... 65
30. Recall for Proposed System ....................................... 67
31. Precision for Proposed System (%) ............................... 68
32. Sample OCR Data for Input ....................................... 68
33. Sample System Output ............................................. 69
34. Results from Scanned CD Covers .................................. 70
35. Results from Mobile Phone Scanned CD Covers ................ 71
36. Results of Processing freedb CD Covers ......................... 71
37. OCR Data from Test 1 ............................................. 83
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>OCR Data from Test 2</td>
<td>84</td>
</tr>
<tr>
<td>40</td>
<td>OCR Data from Test 3</td>
<td>84</td>
</tr>
<tr>
<td>41</td>
<td>OCR Data from Test 4</td>
<td>85</td>
</tr>
<tr>
<td>42</td>
<td>OCR Data from Test 5</td>
<td>86</td>
</tr>
<tr>
<td>43</td>
<td>OCR Data from Test 6</td>
<td>87</td>
</tr>
<tr>
<td>44</td>
<td>OCR Data from Test 7</td>
<td>88</td>
</tr>
<tr>
<td>45</td>
<td>OCR Data from Test 8</td>
<td>89</td>
</tr>
<tr>
<td>46</td>
<td>OCR Data from Test 9</td>
<td>90</td>
</tr>
<tr>
<td>47</td>
<td>OCR Data from Test 10</td>
<td>91</td>
</tr>
<tr>
<td>48</td>
<td>OCR Data from Test 11</td>
<td>92</td>
</tr>
<tr>
<td>49</td>
<td>OCR Data from Test 12</td>
<td>94</td>
</tr>
<tr>
<td>50</td>
<td>OCR Data from Test 13</td>
<td>95</td>
</tr>
<tr>
<td>51</td>
<td>OCR Data from Test 14</td>
<td>96</td>
</tr>
<tr>
<td>52</td>
<td>OCR Data from Test 15</td>
<td>97</td>
</tr>
<tr>
<td>53</td>
<td>System Output from Test 1</td>
<td>98</td>
</tr>
<tr>
<td>54</td>
<td>System Output from Test 2</td>
<td>99</td>
</tr>
<tr>
<td>55</td>
<td>System Output from Test 3</td>
<td>100</td>
</tr>
<tr>
<td>56</td>
<td>System Output from Test 4</td>
<td>101</td>
</tr>
<tr>
<td>57</td>
<td>System Output from Test 5</td>
<td>102</td>
</tr>
<tr>
<td>58</td>
<td>System Output from Test 6</td>
<td>103</td>
</tr>
<tr>
<td>59</td>
<td>System Output from Test 7</td>
<td>105</td>
</tr>
<tr>
<td>60</td>
<td>System Output from Test 8</td>
<td>106</td>
</tr>
<tr>
<td>61</td>
<td>System Output from Test 9</td>
<td>107</td>
</tr>
<tr>
<td>62</td>
<td>System Output from Test 10</td>
<td>108</td>
</tr>
<tr>
<td>63</td>
<td>System Output from Test 11</td>
<td>109</td>
</tr>
<tr>
<td>64</td>
<td>System Output from Test 12</td>
<td>111</td>
</tr>
<tr>
<td>65</td>
<td>System Output from Test 13</td>
<td>112</td>
</tr>
<tr>
<td>66</td>
<td>System Output from Test 14</td>
<td>113</td>
</tr>
<tr>
<td>67</td>
<td>System Output from Test 15</td>
<td>114</td>
</tr>
<tr>
<td>68</td>
<td>OCR Data from Test 1</td>
<td>116</td>
</tr>
<tr>
<td>69</td>
<td>OCR Data from Test 2</td>
<td>117</td>
</tr>
<tr>
<td>70</td>
<td>OCR Data from Test 3</td>
<td>118</td>
</tr>
<tr>
<td>71</td>
<td>OCR Data from Test 4</td>
<td>119</td>
</tr>
<tr>
<td>72</td>
<td>OCR Data from Test 5</td>
<td>122</td>
</tr>
<tr>
<td>73</td>
<td>System Output from Test 1</td>
<td>123</td>
</tr>
<tr>
<td>74</td>
<td>System Output from Test 2</td>
<td>124</td>
</tr>
<tr>
<td>75</td>
<td>System Output from Test 3</td>
<td>125</td>
</tr>
<tr>
<td>76</td>
<td>System Output from Test 4</td>
<td>126</td>
</tr>
<tr>
<td>77</td>
<td>System Output from Test 5</td>
<td>127</td>
</tr>
<tr>
<td>78</td>
<td>Test 1</td>
<td>128</td>
</tr>
<tr>
<td>79</td>
<td>Test 2</td>
<td>129</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

80 Test 3 .................................................. 130
81 Test 4 .................................................. 131
82 Test 5 .................................................. 132
83 Test 6 .................................................. 133
84 Test 7 .................................................. 134
85 Test 8 .................................................. 135
86 Test 9 .................................................. 136
87 Test 10 ............................................... 137
88 Test 1 .................................................. 138
89 Test 2 .................................................. 139
90 Test 3 .................................................. 140
91 Test 4 .................................................. 141
92 Test 5 .................................................. 142
93 Test 6 .................................................. 143
94 Test 7 .................................................. 144
95 Test 8 .................................................. 145
96 Test 9 .................................................. 146
97 Test 10 ............................................... 147
Abstract

This project presents the steps taken in designing and implementing a system which using the results from running Optical Character Recognition software on CD backs, parses the track titles and removes superfluous data. The system is implemented using an n-gram probabilistic approach. It is developed using Perl, MySQL and an external music resource called freedb. The benefits of such a system would be a means of reliably storing an inventory of a music collection for either personal or commercial use.
1 Introduction

1.1 Introduction

This project designs and implements a system architecture for parsing track titles from CD covers. The data is taken from the results of running Optical Character Recognition software on digital images of CD back covers. This report details the steps involved from pre-processing the images to yield the best quality results right up to the actual statistical parsing of the data. There are many applications which use Optical Character Recognition software for document processing and others which use it in a problem specific domain such as business card recognition. However, one popular application like the one we decided to work on has not yet been constructed, that is, which has such a specific goal of information extraction from multi-coloured and highly variable documents such as those of CD covers. Thus, this project represents a novel application of information extraction strategies.

1.2 Aims

This program aims to build a system to successfully parse track titles from CD covers. The text from the CD covers will be extracted using OCR software. While there will be obvious noise, with respect to the actual clean text contained in the original source, introduced by the OCR software e.g. word insertions, deletions, and substitutions, it is not an explicit goal of this project to rectify the OCR output but merely to use it as a raw material to identify where a track begins and ends. It will also be an aim to identify what constitutes superfluous information such as track durations, copyright or production information. Of course, a larger version of this problem may well be interested in extracting all possible data fields from CD covers, but that was deemed too ambitious for the scope of this project.

1.3 Motivation

As electronic media becomes more and more widespread, the need for transferring hard copy data to an electronic format is growing. The obvious advantages are the convenience and efficiency over traditional storage techniques that use a paper medium. In principle, this format is less dependent on any physical medium; however, electronic data storage has yet to prove its longevity as compared to, say, papyrus. This can lead to better storage, organisation, editing, searching, transmission and retrieving of the stored information.
With the revolution of digital audio players like that of the iPod which has over 42 million users worldwide (apple.com), there is now more than ever a desire to convert old music collections to an electronic format. The possibility of integrating such a system with a mobile device with a built in camera would allow the possibility of taking a picture of a CD and then checking if this album is part of a collection at home.

As well as having a personal inventory of a music collection, such a device would have a commercial application. A record store or even book stores all benefit from an inventory list of stock, however, second hand shops often lack these. Imagine the simplicity of being able to scan old record sleeves and instantly have a searchable and editable database of stock. Immediately, reliable and multiple copies of this electronic data can be stored and backed up, protecting claims for insurance against such disasters as fire or theft.

1.4 Overview

This dissertation is presented in six main parts:

- **Chapter 2:** Provides a comprehensive review of OCR research and development. This will cover the initial birth of Optical Character Recognition, the challenges faced when carrying out OCR, as well as looking at various pre and post processing techniques which have been applied.

- **Chapter 3:** Proposes a system architecture and looks briefly at each stage of the proposed pipeline architecture. The motivations for deciding on a statistical approach based on word n-gram relative frequencies are also considered here.

- **Chapter 4:** Discusses the various pre-processing techniques employed to obtain the best quality image for OCR processing. Also, the possibility of introducing a mobile device to capture and transfer the image to a PC is looked at. Finally, a review of potential off-the-shelf OCR software is examined and various problems encountered are outlined.

- **Chapter 5:** Outlines the overall implementation of the project. This includes a description of the use of Perl and MySQL in the project. The parsing stages are described in depth and also a look at two possible music resources is provided. The building of the language model is described at this point and a look at bigrams and probability calculations
is given. Finally, smoothing techniques and the actual guessing of the track title is described.

• **Chapter 6**: Testing of the system from three different perspectives is carried out. The system is tested using images acquired from a regular flatbed scanner, using a mobile phone camera and finally taking samples from the freedb music resource. An analysis and commentary of these results is provided and a look at possible reasons for variations in the metrics is given.

• **Chapter 7**: Summarises the information given in the preceding chapters and provides directives for future work in the event that someone may want use the report as a basis for another project.

1.5 Final Introductory Remarks

This dissertation represents a combination of my skills and experiences developed while in CSLL. While project limitations and problems are noted so too are original contributions. Many areas of the project also highlight where my learning motivations extend beyond the scope of the degree syllabus.
2 Background Reading

2.1 Introduction

This chapter aims to provide details of some key concepts essential in understanding the objective of this paper. A review of research carried out in the area of Optical Character Recognition (OCR) and in particular with regard to methods employed to obtain data from OCR. The methods looked at will broadly come under a template approach as well as various post-processing probabilistic techniques. Where relevant, I emphasise the role of particular issues general to my task of information extraction for the back of CD covers as outlined in section 2.5.

2.2 The Birth of Optical Character Recognition

Optical Character Recognition (OCR) is the process of using computer software to translate pictures of text into a standard encoding scheme representing them in ASCII or Unicode. This history is drawn from Aim (2000).

In 1950, David Shepard, a cryptanalyst with the American Foreign Service Association (AFSA), known today as the National Security Agency (NSA), was asked to suggest data automation procedures for the Agency. One such automation problem was that of converting printed messages into machine language for computer processing. Shepard decided it must be possible to build a machine to do this, and, with the help of Harvey Cook, built “Gismo”. Shepard then established Intelligent Machines Research Corporation (IMR), which delivered the world’s first OCR systems used in commercial operation. While both Gismo and the later IMR systems used image analysis\(^\text{1}\), as opposed to character matching which resulted in more font variation, Gismo was limited to recognising characters located close together in a vertical manner, whereas the following commercial scanners analyzed characters anywhere in the scanned field. The first commercial system was installed at the Readers’ Digest in 1955.

The second system was sold to the Standard Oil Company for reading credit card imprints for billing purposes, with many more systems sold to other oil companies. Other customers of IMR during the late 1950s were a bill stub reader to the Ohio Bell Telephone Company and a page scanner to the U.S. Air Force for reading and transmitting by teletype typewritten messages. IBM and others later licensed Shepard’s OCR patents.

\(^{1}\)Image analysis is the extraction of meaningful information from images by means of digital image processing techniques whereas character matching tries to match individual characters specific to font and style.
Many national postal services, starting with the United States Postal Service in 1965, use OCR machines to sort mail based on technology developed by Jacob Rabinow. Rabinow’s work included the development of a device which scanned printed material and compared each character to a set of standards in a matrix using the “Best match principle” to determine the original message. Subsequently, OCR systems were capable of reading the name and address of the addressee at the first mechanized sorting centre, and print a routing barcode on the envelope based on the postal code. After that the letters need only be sorted at later centres which read the barcode. To avoid interference with the human-readable address field which can be located anywhere on the letter, special ink was used that is clearly visible under UV light. This ink looks orange in normal lighting conditions. Envelopes marked with the machine readable barcode could then be processed. This practice is still in place today.

2.3 Historical Review of OCR Research and Development

The history of OCR recognition is closely related to that of speech recognition both of which, at the time, came under the heading of pattern recognition. Early research believed that it would be relatively easy to develop an OCR system but the complexity of the task at hand was quickly realised. While great progress was made in the early days, people diversified their interests over various topics including image understanding and 3-D object recognition. Thus, OCR research blends with research in computer vision, but the interest in text extraction from images remains a well defined subfield.

The past 50 years of OCR’s existence can be divided into 2 eras, the research and development of OCR systems and the historical development of commercial OCRs, by Mori, Suen, and Yamamoto (1992). Furthermore, the research and development side can be discussed using two approaches — template matching\(^2\) and structure analysis. This will be described, in turn, below.

In the 1950s, Tausheck’s principle patent introduced the principle of template/mask matching. This was closely related to the technology at the time and used optical and mechanical matching. This involved light been passed through mechanical masks, then captured by a photodetector, and scanned mechanically. When an exact match occurs, light fails to reach the detector and so the machine recognises the characters printed on the paper. In math-

\(^2\)Template matching is a technique for finding small parts of an image which match a template image.
ematical terms, this is based on Euclid’s principle of superposition. This approach is elaborated further in the next section.

2.3.1 Template-Matching Methods

According to Jain, Duin, and Mao (2000), template matching was one of the earliest attempts at automated pattern recognition which aims to find the similarity between two entities of the same type. In template matching, a template of the pattern to be recognised is available. The pattern to be recognised is matched against the stored template while taking into account all possible position and scale changes. The similarity measure, often a correlation, may be optimized based on the available training set. Often, the template itself is learned from the training set.

Initial work by Glauberman (1956) to solve the complexity was achieved by projecting from two-dimensional information onto one using a magnetic shift register. An appropriately placed input character was scanned vertically from top to bottom by a slit through which the reflected light on the printed paper was transmitted to a photodetector. A value proportional to the black area within the slit which segments the input character was achieved, see figure 1. The sampled values were sent to the register to convert the analogue values to digital ones. Template matching is done by taking the total sum of the differences between each sampled value and the corresponding template value, each of which is normalised.

![Figure 1: 2-D Reduction to 1-D by a slit](image)

Hannan (1962) used very sophisticated OCR techniques combining elec-
tronics and optical techniques to look at two-dimensional information. Hannan’s research concluded “In summary, the test results of this program proved that the RCA optical mask-matching technique can be used to reliably recognise all the characters of complete English and Russian fonts (91 channels are necessary).” (Mori et al., 1992, p. 1031). However, no further plans for a commercial project based on this research were continued.

In subsequent years, the development of hardware and complex algorithms greatly aided in the design of OCR. Subsequently, a logical template matching method is introduced. The simplest one is called the peephole method, see Figure 2. Firstly, it is assumed that an input character is binarized. Imagine a two-dimensional memory plane on which a binarized input character is stored and registered in accordance with some rule. Then, for an ideal character, which has constant size and width, black portions are always black and the same is the case for the white background. Subsequently, appropriate pixels are chosen for both black and white regions so that the selected pixels can distinguish the input character from characters belonging to other classes.

![Figure 2: Peephole Method](image)

Listed below are some of the approaches that were used as described in Mori et al. (1992). This paper will not go into these methods as it is beyond the scope of this project.

1. Application of Information Theory — IBM carried out research on automatic generation of discriminate logic functions and applied it to their OCR design.
2. Normalization — Iijima emphasised the importance of normalization in template matching and thus introduced this idea of blurring. It is still a well-known pre-processing technique in Japan.

3. Karhunen-Loeve Expansion — Iijima wrote a second paper devoted to the theory of feature extraction. Based on the consideration that for a given normalized pattern set denoted by $D = h(x, \alpha)$, where $\alpha$ is an assigned/indexing number of the individual pattern, we should choose basic functions for a coordinate system of the pattern representation space.

4. Series Expansion
   a) Moment
   b) Fourier Series

5. Feature Matching

6. Nonlinear template matching

7. Graphical Matching

A commercial OCR device based on the peephole method was released in 1957 and was called ERA (Electric Reading Automation). The characters read were numerals printed by a cash register. Section 3.3 details the pipeline architecture model that I am working with, adopting off the shelf OCR. Such software is intended to be general purpose. As such, and given the range of possible formats of CD covers (see appendix B, it is clear that template matching would offer no obvious advantage to OCR for my project.

2.3.2 Structure Analysis Method

According to Mori et al. (1992), the template matching is really only appropriate for the recognition of printed characters. If one wants to implement a system to include the recognition of hand-printed/cursive characters a different approach had to be taken as the variation of shape of handwritten characters is so large\(^3\). It is at this point that the structure analysis method is introduced. These methods were applied to stylised fonts as well as the recognition of constrained hand-printed characters. A structure can be broken into parts; it can be described by the features of these parts and by the relationships between these parts. The problem then is to choose features

\(^3\)While this is a task routinely carried out by the human brain, it is an enormous challenge for a computer.
and relationships between them so that the description gives each character a clear identification. Feature extraction has thus played a central role in pattern recognition research. Listed below are some of the approaches that were used as described in Mori et al. (1992). This paper will not go into these methods further as it is beyond the scope of this project.

1. Thinning Line Analysis
2. Bulk Decomposition
3. Stream Following Analysis
4. Vectorization
5. Contour Following Analysis
6. Background Analysis
7. Syntactic/Linguistic Approach

The application of this approach to pattern recognition was explored and developed extensively. By scanning the labelled plane in a certain order, a string of labels can be constructed. The problem is how to construct an automaton which accepts the string as belonging to a class. A language whose grammar accepts the string must be constructed. A syntactic approach makes use of formal language theory, but a real image is not so simply recognised. Some typical problems were described by Ali and Pavlidis (1977) as follows:

1. The direct syntactic analysis is faced with the need to handle the effects of noise, which causes rather complicated strings.

2. The parsing of the whole boundary requires the use of context-sensitive grammars for a description of the complete object.

2.3.3 Slit/Stroke Analysis

This is an extension of the peephole method previously discussed. This is because the peephole method is not always limited to a single pixel. Instead, it can be extended into a slit or a window whereby it is not necessary to fix at a specific position on the two-dimensional plane and work based on this approach can be seen in T. Sakai, Nagao, and Shinmi (1963).
2.3.4 Hybrid of Template Matching and Structural Analysis

This is a combination of the best features from both the template matching and the structural analysis i.e. template matching is very sensitive to positional change, but it can be very strong in the sense of global matching. On the other hand, structural analysis has the advantage of detecting the local stroke features of characters. As a result of this, the so-called zoned feature method was born. The general idea is that pixel wise matching is made loose, replaced by subregionwise matching, and the matching objects are local features within the subregion. An example of this method can be seen in Munson (1968).

2.4 Generations of Commercial OCR

By the reckoning of Mori et al. (1992), the first generation appeared in the 1960s. In Japan, by the end of the 1960’s, some mainframe companies announced their OCR’s of the first generation. They included the Facom 6300A from Fujitsu and the H-852 from Hitachi, both of which used the previously mentioned stroke analysis method. On the other hand, the N240D-1 of NEC was somewhat similar to the IBM 1417 and also read the special IBM 407 font.

The second generation of OCR systems can be characterised by hand-printed character recognition capabilities. At the early stage they were restricted to numerals only. Such machines appeared in the middle of the 1960s and early 1970s. The first and a famous OCR system was the IBM 1287, which was exhibited at the World Fair in New York in 1965 as a letter-sorting machine for postal code numbers (see section 2.2). Toshiba was the next big company to get involved with the development of a machine using a now famous structural analysis approach. Two years later, NEC developed an automatic OCR sorter for postal code systems. The algorithm developed by NEC was based on Rabinow’s stream following analysis as looked at in section 2.2. In contrast to Toshiba’s approach, an image was considerably reduced to a 6x10 matrix of pixels. For the printed characters they basically used template matching. For the numerals, they improved the algorithm taking a higher resolution of a 16x20 matrix.

The second generation of OCR can also be described by the recognition capability of a set of regular machine printed characters as well as hand-printed characters. In this respect, a very powerful OCR system, called RETINA was developed by Recognition Equipment Inc. This system read both hand-
printed and machine-printed characters. When hand-printed characters were considered, the character set was constrained to numerals, and the following letters and symbols: C, S, T, X, Z, +, and -. On the other hand, the system that could read printed characters could recognise 40 characters: a complete uppercase alphabet, letters, numerals, and four special symbols. The user specified the font to be read by the machine. A key feature of RETINA, in terms of OCR systems, lies in its parallelism, which is associated with the name RETINA. The processing speed was very high and concerning the recognition of hand-printed characters, strokes were matched, and line crossing and corner types were also taken into account. Positional relations of detected features were all examined to identify an input character (Mori et al., 1992). Early in the 1970s, the main OCR barrier was overcome with the development of a high-performance and low-cost system from Hitachi called H 8959. The reduction in cost can be attributed to using a laser scanner and a special-purpose OCR processor instead of wired logic circuits.

By the end of the 1960’s, the third generation of OCR systems were targeting poor-print-quality characters, and hand printed characters for a large category character set, such as Chinese characters. By document, it is meant that it includes words such as names, addresses, and commands. These targets have been achieved partially and such commercial OCR systems appeared roughly during the decade from 1975 to 1985. Low cost and high performance are always common objectives for these systems.

2.5 Challenges of OCR Processing

OCR involves using an “optical mechanism” comparable to the human eye to recognise text within an image (Aim, 2000). Like the brain, the OCR sees the input but the ability to understand what is presented varies from person to person depending on many factors. By reviewing these problems, one can better understand the challenges faced in the development of an OCR system.

Firstly, if a human reads a page in a language other that their native language then they may recognise the various characters but are unable to recognise words. Of course the exception to this is syllabaric languages such as Chinese or Japanese, thus, a Chinese person is capable of understanding written Japanese. However, on the same page, we are usually able to interpret numerical statements — symbols for numbers are universally used whether it’s a fraction written as 1/2 or $\frac{1}{2}$ or a decimal in Ireland written as 3.14 or as 3,14 in France. This explains why many OCR systems recognise numbers but few understand the full alphanumeric character range — inclusive of
diacritics.

Secondly, there is a similarity between many numerical and alphabetical symbol shapes. For example, while examining a string of characters combining letters and numbers, there is very little visible difference between a capital letter “O” and the number “0”\(^4\). As humans, we can read the sentence or entire paragraph to help us determine an accurate meaning. This procedure, however, is much more difficult for a machine.

Thirdly, we rely on contrast to recognise characters. We may find it very difficult to read text which appears against a very dark background or is printed over other words or graphics.

The ideal conditions for effective OCR processing would be as follows:

- 12 point font size up to approximately 20 point font size.
- Black text on a white background.
- A clean copy; not a fuzzy generation copy removed from an original by successive copying.
- Standard type font (Times New Roman, Arial, etc.) Fancy fonts may pose difficulties during recognition.
- Single column layout.

Conversely, factors which would lead to poor OCR recognition rates would include:

- Variations of foreground and background colours.
- Different sized fonts and unpredictably positioned text.
- Background images.
- Complex formatting.
- Bad quality copy — smudges, creases, low-contrast.
- Misapplication of lexicons or mixing character sets (e.g., when more than one language dictionary is loaded).
- The character sets of certain languages might not be supported.

It is because of these factors and many more that OCR processing of CD covers is such a challenge.

\(^4\)Note that in earlier days of programming, ø was used as a convention to discriminate zero from the alphabetic character “O”.
2.6 Pre-Processing Techniques

2.6.1 Character Recognition

Saiga, Nakamura, Kitamura, and Morita (1993) implements an OCR system for a particular purpose — business card recognition of Japanese business cards. This is relevant because of its proximity to the CD processing problem of my project. The system is implemented on a wide variety of formats and fonts and subsequently classifies recognition results into several predefined categories so that a business card database is built automatically. This is similar to my database construction component and its discussion in section 5.5.

This system is implemented in 2 phases — a character recognition phase (structural analysis) and a linguistic processing phase.

The aims of the character recognition phase is to guess the direction of the lines and then to extract them. Next, they are split into character fragments. The direction of each character on the map is decided by comparing them with the standard patterns. The system works through the text line-by-line while keeping track of their matching scores and candidate categories.

This process involves various steps in the recognition. I shall now briefly outline the details of these steps.

2.6.2 Connected Component Extraction

Firstly, the system extracts connected components of black pixels in the image and subsequently records each connected component by storing the coordinates.

2.6.3 Line Direction Determination

There are many possible directions for a line on a card to have. However, it is assumed that all lines are in the same direction so as to make savings on processing time and also to discount the rarity of the other possibilities. This system also makes use of business card layouts to determine line direction.

2.6.4 Line Extraction

Here, a line is defined as a set of connected components. So, this means splitting the set into several subsets to meet the criteria of what a line is.
2.6.5 Character Matching

The inputs here are the coordinates of 2 points in the image. The matching part extracts the area whose diagonal line ends at the 2 points and turns the area into a vector. The vector is compared to a set of standard patterns and matching scores for each acceptable character calculated. The four highest-scored choices’ character codes and their scores are stored. There are 2 sets of standard patterns: one is for ordinary fonts and the other for company logos.

2.6.6 Character Direction Determination

After the lines have been processed, some of the characters are chosen to decide the direction of the characters on the card face. This is very similar to what has been done in the line direction determination. The system picks one line to be used as a benchmark. The scores for the four vectors are compared and the direction whose vector gives the highest score is considered to be the right one for the fragment.

2.6.7 Linguistic Processing

The system proposed by Saiga et al. (1993), there are three aims of linguistic processing in this system which are as follows:

- To provide a right choices based on the information extracted in the recognition phase.

- Dividing strings into substrings so that each of them represents one of the predefined categories.

- Giving each substring a “tag” that shows which category it stands for.

The linguistic processing is done on a line-by-line basis i.e. every time it receives results from the recognition phase, it is invoked.

Furthermore, this phase can be divided into 4 sub-phases: string segmentation, consistency checking and candidate selection. These are discussed below.

2.6.8 String Segmentation

The first stage for Saiga et al. (1993) was to segment each string by comparing words to those in a previously built database. To do this, delimiters needed to be defined using predefined keywords and changes in character size.
This process is more specific to Japanese as word boundaries are usually more explicit than in European languages.

1. Segmentation by keywords: The system searches the strings for keywords that are predefined. These keywords were chosen in such a way that they become prefixes or suffixes for substrings. Examples of such keywords would include address suffixes, title/position suffixes, fax prefixes/suffixes or phone number prefixes/suffixes.

If one of the keywords is found in the string it becomes a candidate for a delimiter that separates substrings.

2. Segmentation by databases: When the input substring has no keywords in it, the system compares it with some of the database proper nouns which consist of biographical names and organization names. If the substring matches a word in the database, the system gives the substring the category that the database implies.

### 2.6.9 Consistency Checking

This stage is using a document-level grammar for the structure of business cards. The following assumptions are made on business card formats:

- There is at least one person’s name on a card.
- There is at least one organization name on it.
- If an address and a phone number are written closely, they belong to the same place or person.

Person’s Name Checking: If the results don’t contain a person’s name and there are unclassifiable strings the system checks again in a less strict manner if it is not a person’s name. Address and Phone Number Consistency Checking: If an address is recognised correctly it is possible to guess the area code for its phone number by having an address-area code table in the system.

This idea of consistency checking could be applied to CD covers in the sense that it could check, for example, that if a number contained within brackets was found to be close to a track title then it probably is the track duration.

### 2.6.10 Stochastic Context Free Grammars

The solution to the problem of extracting information from OCRd text can be improved by applying a probabilistic approach to develop a grammar for
the structure of a card and associating probabilities with the productions in that grammar. A stochastic context-free grammar (SCFG) is a context-free grammar in which each production is augmented with a probability. The probability of a derivation (parse) is then the product of the probabilities of the productions used in that derivation; thus some derivations are more consistent with the stochastic grammar than others. SCFGs are essentially an extension of context-free grammars. SCFGs are a specialized form of weighted context-free grammars.

This can be done using a variant of the CYK algorithm to find the Viterbi parse of a sequence for a given SCFG. The Viterbi parse is the most likely derivation (parse) of the sequence by the given SCFG.

The Inside and Outside algorithms are analogues of the Forward algorithm and Backward algorithm, and can be used to compute the total probability of all derivations that are consistent with a given sequence, based on some SCFG. This is equivalent to the probability of the SCFG generating the sequence, and is intuitively a measure of how consistent the sequence is with the given grammar.

In a paper by Handley, Namboodiri, and Zanibbi (2005), a document understanding system in which the arrangement of lines of text and block separators within a document are modelled by stochastic context free grammars is presented. A grammar corresponds to a document genre. The system incorporates an optical character recognition system that outputs characters, their positions and font sizes. These features are combined to form a document representation of lines of text and separators. Lines of text are labelled as tokens using regular expression matching. The maximum likelihood parse of this stream of tokens and separators yields a functional labelling of the document lines.

This system is applied to business cards and business letter applications. The aim of the system is to extract enough information, which Handley et al. (2005) class as metadata, to process the document, e.g., in a business letter, appropriate metadata might include the recipient’s name and address.

The system uses stochastic grammar to model documents — each document style or genre having its own grammar and is built using the constraints that it will be modular, trainable and robust.

Structured documents including business cards and letters exhibit a degree of randomness with metadata contained at varying positions. The location of such data may have certain regularities but they are not normally deterministic.

A scanned business document is converted to a list of words with attributes. The word list is converted to a sequence of text line types and text block separators representing a linearization of text blocks found in a
Figure 3: Application of Stochastic Context-Free Grammars to Business Card Labelling

document. Token sequence is then parsed using a SCFG. The most probable parse is used to assign metadata types to text blocks. Detected metadata fields can then be used to store in a document database.

Regular expressions are used to label text lines to avoid the maintenance of a vast database of street names, cities, names, etc.

2.6.11 Structure Analysis

A mixed approach to document structure analysis is presented in a paper by Klink, Dengel, and Kieninger (2000). It makes use of layout as well as textual features of a given document. These features are the base for potential conditions which in turn are used to express matched rules of an underlying rule base.

Rules can be formulated based on features which might be observed within one specific layout object. Rules can also express dependencies between different layout objects.

The document processing in this experiment is divided into two stages: document analysis and document understanding. The approach is taken that a document has two structures, the layout structure and the logical structure. Extraction of the layout structure from a document is referred to document analysis. Mapping the layout structure into a logical structure is referred to document understanding.
Figure 4: Layout and Logical Structure of the Document Model

While structure analysis wasn’t applied to the CD covers, it is certainly an approach which could warrant further investigation as mentioned in section 7.4.

2.6.12 Recognition of the Scanned Document

Once the document has been scanned, it is segmented and the characters and their fonts are recognised by an OCR system.

This resulting file includes text, font and layout attributes has further analysis conducted on it to find the “structured” parts of the document.

The results are added to the document tree and are stored for further analysis and retrieval steps.

From the physical point of view, the document has some pages. A page consists of some blocks, a block consists of some lines, and a line consists of some words. The document must contain at least one object of each type.

From the logical point of view, the document contains logical units, e.g. a heading, a title, tables or lists etc. A logical unit has several characteristic appearances: it can either be a title, heading, abstract, author, footnote, headline etc. In these cases the logical units consist of exactly one block. These logical units can be assigned with various labels each of them carrying a probability value (e.g. 80% heading, 12% title, 8% author).

2.6.13 Document Understanding

The document understanding is divided into two different phases of processing:
1. Common document structures like headings, footnotes and lists are recognised.

(a) Header and footer location and searches for text not too far away from the top and text not too far from the bottom.

2. All domain dependent logical elements are recognised using the appropriate rule base.

(a) Textual relations allows the definition of rules like, “There must be a block which has a least one word in common with the block for which the rule might be applied and furthermore it needs to contain the words ‘Dear’ and ‘Mr.’”. This could for instance express a relation between the recipient field of a letter and the salutation field.

(b) Label relations occur when one logical (labelled) object triggers the labelling of other objects, e.g., if the ‘title’ is found on a scientific paper, then the ‘author’ (or an ‘abstract’) is usually found more or less directly underneath the ‘title’. Thus, if no ‘title’ exists on the (unknown) document, no ‘author’ (or ‘abstract’) will exist too and the risk of a lower precision by labelling an object incorrectly as e.g. ‘author’ is reduced. But this feature could also cause a problem: if the ‘author’ label is depending on the ‘title’ block then an ‘author’ (or ‘abstract’) could never be labelled until the ‘title’ is found. If a ‘title’ object is missed then all other depending labels will not be recognised.

2.7 Post-Processing Techniques

2.7.1 N-Gram Classification

An n-gram based text categorisation system is proposed by Cavnar and Trenkle (1994) to classify documents by language.

2.7.2 What is an n-gram?

An N-gram is an N-character slice of a string. Typically, one slices the string into a set of overlapping N-grams. Sometimes, blanks are appended to the beginning and ending of the string in order to help with matching beginning-of-word and ending-of-word situations.

So, the word “TEXT” would be composed of the following N-grams:
Bi-grams: _T, TE, EX, XT, T_
Tri-grams: _TE, TEX, EXT, XT_, T_
Quad-grams: _TEX, TEXT, EXT_, XT_, T___

N-gram matching has been successful in dealing with noisy input in other areas including the interpretation of postal addresses, in text retrieval and a wide variety of other natural language processing applications.

The main benefits of N-gram-based matching is that since every string is decomposed into small parts, any errors that are present tend to affect only a limited number of those parts, leaving the remainder intact.

### 2.7.3 Generation N-Gram Frequency Profiles

Cavnar’s approach involves:

- Split the text into separate tokens consisting only of letters and apostrophes. Digits and punctuation are discarded. Pad the token with sufficient blanks before and after.
- Scan down each token, generating all possible word N-grams, for N=1 to 5.
- Hash into a table to find the counter for the word N-gram, and increment it.
- When done, output all N-grams and their counts.
- Sort those counts into reverse order by the number of occurrences. Keep just the N-grams themselves, which are now in reverse order of frequency.

At this point, there is an N-gram frequency profile for a document and the following observations can be made: Top 300 correlate to the most frequent language n-grams. Highest are unigrams followed by function words e.g. the, and etc. As the list progresses they become more specific to the subject.

### 2.7.4 Comparing and Ranking N-Gram frequency Profiles.

Two N-gram profiles are taken and a simple rank-order statistic is calculated which they the “out-of-place” measure. This measure determines how far out of place an N-gram in one profile is from its place in the other profile.
2.7.5 Advantages of the N-Gram Frequency Technique

According to Cavnar and Trenkle (1994), suggestions are made that an n-gram approach is suitable for correcting text coming from noisy sources such as email or OCR systems.

It is possible that one could achieve similar results using whole word statistics. In this approach, one would use the frequency statistics for whole words.

However, there are several possible problems with this idea. One is that the system becomes much more sensitive to OCR problems with a single misrecognised character throwing off the statistics for a whole word.

A second possible difficulty is that short passages are simply too short to get representative subject word statistics. By definition, there are simply more N-grams in a given passage than there are words, and there are consequently greater opportunities to collect enough N-grams to be significant for matching.

Other advantages of this approach are the ability to work equally well with short and long documents, and the minimal storage and computational requirements. A further look at the advantages of using an n-gram approach can be seen in section 3.4 and indeed the problems encountered from an n-gram approach can be seen in section 7.3.

2.7.6 Probabilistic Information Retrieval Systems.

Taghva, Borsack, and Condit (1994) evaluates retrieval effectiveness from OCR text databases using a probabilistic IR system. These retrieval results are compared to their manually corrected equivalent showing that there is no statistical difference in precision and recall using graded accuracy levels from three OCR devices. However, characteristics of the OCR data have side effects that could cause unstable results with this IR model. They find individual queries can be greatly affected. Knowing the qualities of OCR text, an automatic post-processing system is applied to improve effectiveness.

In this study, a set of queries are ran against two databases. Each database contained the same document collection, but one database had been manually corrected to a level of 99.8% correctness while the other had been automatically generated by an OCR device. Similarly, in my system a database is constructed to read and write results to and from. This will be looked at further in section 5.5.
2.7.7 Post-Processing System

This system was designed to correct OCR errors in text without human intervention. The post-processing system was run on three databases with varying qualities — best, middle and worst — to see what improvements could be attained.

The original post-processing system consisted of numerous steps. First, a full-text database of documents and an index listing all the unique indexed words in the database was obtained. Then, using word frequency information, an English dictionary, and a domain-specific dictionary, the words were divided into a list of “centroids”, and a list of misspellings. The centroids were the words which were spelled correctly and had high frequency. The misspellings were then grouped around the centroids using approximate pattern matching techniques. The idea was that if a misspelled word was found that was close to a centroid word, then it would be possible to equate those misspelled words to its centroid. This procedure worked for the most part, but sometimes a misspelled word would be close to more than one centroid. Two heuristics were developed to alleviate this problem. The first involved using document frequency information about the words. The second process took information about the kinds of mistakes the OCR device was likely to make, and eliminated unlikely occurrences. Finally, a list of misspelled words and their correct words was produced, the replacements were made, and the database was reloaded.

The most notable problem with using OCR text with a probabilistic system was the amount of “junk” in the text, and subsequently in the index, that caused unreliable term frequencies. One difficulty is determining which terms are “junk” and which terms are misspellings that could be rectified by the post-processing system. After analyzing the list of indexed words the following heuristics were implemented. Limits on the minimum and maximum lengths of words were set and words with four or more identical consecutive characters were removed. Strings that had a high ratio of non-alphabetic characters or an irregular vowel-to-consonant ratio were also removed.

2.7.8 Retrieval Methods for English-Text with Misrecognised OCR characters

Taghva, Borsack, Condit, and Erva (1994) present three probabilistic text retrieval methods designed to carry out a full-text search of English documents containing OCR errors. By searching for any query term on the premise that there are errors in the recognised text, the methods presented can tolerate such errors, and therefore costly manual post editing is not required after
OCR recognition. In the applied approach, confusion matrices are used to store characters which are likely to be interchanged when a particular character is misrecognised, and the respective probability of each occurrence.

In this paper, three new retrieval methods designed for searching OCR-recognised English text were proposed. These methods are briefly described as follows.

2.7.9 Confusion Matrix Retrieval (CMR) Method

This method applies a confusion matrix which expands the query term into multiple search terms that are matched against recognised text containing OCR errors. The confusion matrix was constructed by comparing OCR-recognised text with corresponding error-free edited text. In particular, characters which were likely to be interchanged when a particular character is misrecognised and the respective probability of each occurrence are stored in the confusion matrix. During the retrieval process in which a full-text search is made, the confusion matrix is referred so that the appropriate search terms can be generated. The retrieved terms are assigned validity values based on their calculated occurrence probability and those with a validity value greater than a given threshold are judged as satisfying the input query.

This approach is similar to that of the centroid and misspelling list looked at in Taghva, Borsack, and Condit (1994).

2.7.10 Expanded Confusion Matrix Retrieval (ECMR) Method

This method is the same as the CMR method with the exception that the algorithm used incorporates not only a simplified confusion matrix for interchanged characters but also an expanded confusion matrix which stores all characters likely to be missed, inserted, combined, or decomposed along with their individual probabilities. This allows spacing errors which often break one-to-one correspondence between the recognised text and original text to be handled effectively.

2.7.11 2-gram Matrix Retrieval System

This method is the same as the CMR method with the exception that the algorithm employed incorporates a 2-gram matrix which stores character connection probabilities based on the 2-gram statistics of English text as well as a confusion matrix. This 2-gram matrix is referred to when calculating the validity value of retrieved terms. This method is expected to exceed
the CMR method because soundness values based on both character connection probabilities and error occurrence probabilities are more consistent than those based on error occurrence probabilities only.

2.8 Conclusion

This chapter has presented a thorough overview of OCR development in the past half century.

Various pre and post processing techniques have been looked at and the following conclusions can be made when planning and organising the approach for this project. Post-processing has included information extraction based on information associated with anticipated structure in the text of the documents being processed, whether envelopes or business cards. In 5.6, I provide a description of the structured elements I have concluded necessary, if not sufficient, to a CD cover. The structural analysis has not been incorporated fully into my system, although 5.6 points out aspects of structural analysis which have been utilised (e.g. ignoring track numbers).

While the character recognition processing is interesting as suggested by Saiga et al. (1993), a lot of research has already been carried out in this area. Also, it is a very small part of the optical character recognition process which is linked more to the OCR software itself and has very little to no linguistic relevance. On the other hand, the Stochastic Context Free Grammar by Handley et al. (2005) method which is implemented is very interesting for business documents and business cards but it is believed that this would be difficult to implement as there are no definitive universal layouts for CD covers. The Structure analysis approach is interesting but could also prove difficult to implement because of the need to define CD cover layouts.

From the post-processing perspective, these methods are more relevant to what I hope to implement. In particular, the N-gram classification approach as implemented by Cavnar and Trenkle (1994) should provide interesting results for calculating probability values to predict song titles. The retrieval methods for English-text by Taghva, Borsack, Condit, and Erva (1994) propose some interesting techniques but which relate to the correction of individual characters more so than whole words.

As a result of this, a structural analysis and N-gram approach will be implemented in my system when identifying the track title and predicting the probability of a word being part of a song title.

This chapter has also allowed me an opportunity to explore areas of Computational Linguistics I had not had much contact with previously in my degree. However, the fundamentals of grammar development approaches studied in CSLL were a very useful knowledge when researching areas such
as context free grammars. I have also got to grips with the basics of image analysis techniques as well as various statistical approaches to Natural Language Processing.
3 Design

3.1 Introduction

In this chapter, the project goals as well as system architecture will be outlined. A brief description of each stage of the procedure will always be done as well as the motivations of methodological choices like an N-Gram approach. Where it was possible, ‘off the shelf’ components have been used.

3.2 Goals

The basic goal for the project will be to design and implement a system which will parse track titles from CD covers and remove irrelevant information including disk duration, copyright and artist or record label information.

3.3 System Architecture

The system architecture proposed is shown in figure 5 and outlined in the steps below. This is just a brief overview at the moment and will be looked at closely in subsequent chapters.

3.3.1 Acquiring the Image

Two methods will be employed to acquire the image. Firstly, the most straightforward way would be to use a flatbed scanner connected to a computer and secondly, using a mobile phone device. The use scenario for the latter was outlined in 4.2.2.

3.3.2 Flatbed Scanner

The flatbed scanner would certainly be the easiest to use and produce the highest quality images. It can be connected quickly and effortlessly to any computer, with appropriate software installed, offering many different settings and colour adjustment options.

3.3.3 Mobile Phone Device

The main advantages of using the mobile phone device would be its portability and real life application for taking images while on the move, sending it back to a PC at home or in the office and then connecting, viewing and editing the results via the Internet.
The principal disadvantage is the quality of the images. While cameras are improving and thus quality is improving very quickly, there are still challenges of having enough light, maintaining a steady hold of the camera to avoid distortion and of course the issue of transferring photos from the phone to the computer. There are a few ways this can be done either via infra-red or Bluetooth, setting up a mobile phone server on your PC or laptop to permit sending them via MMS\(^5\). They could also be transmitted by sending them to an online personal space as provided by a network provider such as Vodafone, O2 or Meteor.

### 3.3.4 Image Pre-Processing

The next stage involves the pre-processing of an image. This stage is to best prepare the image so as to yield the best possible recognition rates from the OCR software. This will be employed keeping in mind the ideal conditions for OCR processing.

Challenges here will be to decide on the best software to use to achieve these results, coming up with a somewhat standard way of deciding when and what effect to use on photos, defining thresholds for conversion to black and white and using different filtering methods to improve the quality. This stage involves direct human intervention in my pipeline architecture, as I intend to focus on automating aspects of text processing rather than image processing.

### 3.3.5 Optical Character Recognition

This stage will deal with the actual processing of the image with the OCR software.

The available software (both licensed by Trinity and free) needs to be reviewed to decide which one provides the best ease of use and results. A comprehensive review can be seen in section 4.6, where the description is related to the considerations named in the historical overview provided in chapter 2.

The resulting output of this will be a Unicode text file taken as input to subsequent components in the pipeline.

---

\(^5\)Multimedia Messaging Service (MMS) is a technology for transmitting not only text messages, but also various kinds of multimedia content (e.g. images, audio, and/or video clips) over wireless telecommunications networks using the Wireless Application Protocol (WAP).
3.3.6 Parse I

This first parse is to take any information about the track numbers (if present) and to put this information into a database.

3.3.7 MyDatabase

MyDatabase is, as the name suggests, is a database that I designed. Its structure is detailed in section 5.5. It will contain information on the parse results from the CD covers with the possibility of populating it fully with missing information.

3.3.8 Language Model

This will be a language model of bigrams and their relative frequency using freedb (see section 5.8.2) as a source for instructing the model.

3.3.9 Extract Information from Database

This next stage will query MyDatabase to extract everything from the title field which the first parse has deemed to be a track title.

3.3.10 Build Bigrams and Compare to Language Model

Next, the parsed titles will be split into bigrams and compared to a language model to calculate the probability of a word been in a title.

3.3.11 Guess the CD Title

In this phase, it is hoped that based on the results from the previous phase that the track titles can be successfully predicted. OCR correction is not undertaken, so the task includes guessing correctly that text which is incorrectly scanned is actually correctly part of the title.

3.3.12 Compare to FreeDB

Finally, an evaluation of the system will be carried out by comparing the results to a music database resource.
Figure 5: Layout and Logical Structure of the Document Model
3.4 Why use the N-Gram Approach?

Recall from section 2.7.1, an N-gram is a sub-sequence of $n$ items from a given sequence. Essentially, it is a model of the likelihood of a sequence among all possible sequences. N-grams have become a popular tool in statistical natural language processing. In speech recognition, phonemes can be modelled using an n-gram distribution. For parsing, words are modelled such that each n-gram is composed of n words.

Some of the advantages of the N-gram approach include the fact that it is language independent, unlike a Context Free Grammar, and since our language model (based on track titles from a database) will contain titles from a wide variety of languages, this is an important feature. Secondly, the robustness of the N-gram method against errors in the training data is important. Since the music database resource will probably contain spelling and formatting errors this will be reflected in the n-gram results as a bigram with errors will have a lower relative frequency.

3.5 Conclusion

In this chapter, the goals of the project have been discussed, the architecture for a proposed system has been outlined and the motivations and advantages of implementing a system based on a statistical n-gram approach have been presented. In subsequent sections the components are detailed, with particular emphasis on those I have implemented. This design phase of the project has complemented programming and database course taken throughout CSLL.
4 Pre-Processing

4.1 Introduction

In this chapter, the pre-processing stages up to just after running the OCR software will be looked at. First, a review of the hardware used to do the scanning (both using a flatbed scanner and a mobile phone) will be presented. Next, a discussion of the image manipulation techniques employed to attain the best quality image for OCR will be looked at. Finally, a review of potential OCR software and the actual running of the OCR will be looked at.

4.2 Acquiring the Image

4.2.1 Scanner

The scanner used to convert the CD covers to digital format was one provided by the Computer Science Department. The scanner model was the **Epson Perfection 2580**.

The features of this scanner were well beyond my requirements providing excellent quality images to work with. A brief outline of the relevant technical specifications is provided below.

![Epson Photo Perfection 2580](image)

Figure 6: Epson Photo Perfection 2580

- **Scanner Type:** Flatbed color image scanner
- **Maximum Scan Area:** 8.5” x 11.7” (216mm x 297mm)
- **Optical Resolution:** 2400 dpi
4 PRE-PROCESSING

- **Hardware Resolution:** 2400 x 4800 dpi maximum with Micro Step Drive technology
- **Maximum Resolution:** 12,800 x 12,800 dpi with software interpolation
- **Color Depth:** 48-bit internal/48-bit external
- **Grayscale Depth:** 16-bit internal/16-bit external
- **Scanning Speed:**
  - Monochrome (2400 dpi, draft mode) 11 msec/line (approx.)
  - Full color (2400 dpi, draft mode) 11 msec/line (approx.)

4.2.2 Mobile Phone

The mobile phone used to capture the CD covers was the Sagem myV-55. The myV-55 has an integrated VGA\(^6\) quality camera that is capable of a resolution of 640 x 480 pixels. A smaller sized mobile format is also available for photos, and there is a 4x digital zoom provided for homing in on sections of a picture if required. However, as this is a fairly basic camera, it cannot compete with the cameras on the more expensive models some of which provide quality comparable to handheld digital cameras.

The camera in this phone performed surprisingly well. The qualities of the images taken were better than I expected, yielding satisfying results for the OCR stage. The main difficulty when capturing the image was to ensure appropriate lighting and to maintain a steady hold to avoid distortion of the image.

This secured some confidence in the feasibility of the application as a whole since it is imaginable that the phone could be used to scan CD covers in unlikely scenarios like flea markets on travel abroad in an attempt to tell if a CD track is already represented in an extensive collection back home.

4.3 Transferring the Image using a Scanner

Transferring the image from the flatbed scanner to my PC didn’t pose any problem. It was just a matter of having a USB port, appropriate drivers installed and software as provided by the manufacturer.

---

\(^6\)Video Graphics Array (VGA) is an analog computer display standard by IBM.
4.4 Transferring the Image using a Mobile Phone

There were several possible ways to implement this. The three main ways I considered implementing this were through Infrared, Bluetooth and via mobile phone server.

It was finally decided to use the infrared option as it was the cheapest and easiest to implement.

4.4.1 Infrared

Infrared is a common form of data transmission employed in short-range communication among computer peripherals and personal digital assistants. Infrared devices use infrared light-emitting diodes (LEDs) to emit infrared radiation which is focused by a plastic lens into a narrow beam. The beam is modulated to encode the data. The receiver uses a silicon photodiode to convert the infrared radiation to an electric current.

Nearly all modern laptops and mobile phones have infrared capabilities. The main problem with infrared communication between the phone and the computer is that the signal can be easily interrupted. It is also relatively slow. However, for the purposes of this project it worked well.
4.4.2 Bluetooth

Bluetooth provides a way to connect and exchange information between devices like personal digital assistants (PDAs), mobile phones, laptops, PCs and printers via a secure, low-cost, globally available short range radio frequency. It normally works within a range of 1 to 10 metres. It is particularly useful for the transfer of files between mobile phones. This is enabled through the OBEX\(^7\) communications protocol.

It is essentially the same form of communication as infrared but is faster and more reliant medium. Since my phone didn’t have bluetooth capabilities, it was not possible to implement this.

4.4.3 Mobile Phone Server

The possibility of integrating a mobile phone server with my PC was also explored as done in a final year project by O’Loghlin (2005). This would allow the possibility of sending a MMS message (similar to a text message but for images and videos) from anywhere to a mobile number and then the image would be stored on my computer.

To implement the mobile phone server, a connection must be maintained to the mobile telephone network so as to be able to receive and interpret MMS messages.

The main way of connecting a mobile phone to the network is via a PCMCIA\(^8\) card. These devices come in the form of PC cards which can be slotted into laptops.

A SIM card is then inserted into the PCMCIA card, and thus this mobile phone number is associated with the server used to receive MMS messages. The server can then receive MMS messages and then decode them to have, for example, an image file. Essentially what happens is that when a new MMS arrives, the phone is notified via SMS. This is a specially encoded SMS which isn’t shown to the user, but instead basically includes a URL. The document has to be fetched at this URL, which is the MMS message.

The notification SMS usually runs across at least two messages, so they have to be reassembled into one. Then it is simply a matter of extracting the URL. At this point the URL is not accessible over the general Internet — it can only be retrieved via a GPRS connection connected with the MMS settings of your network. Once the document has been fetched, it just needs to be decoded. Specification documentation and information on decoding

\(^7\)OBEX (abbreviation of OBject EXchange) is a communications protocol that facilitates the exchange of binary objects between devices.

\(^8\)Personal Computer Memory Card International Association.
this message can be downloaded from the European Telecommunications Standards Institute (ETSI).

It was not possible to fully implement this method as there was no PCM-CIA card available to use within the Department. It was decided that the system would be implemented using the infrared mobile phone capabilities.

4.5 Image Manipulation

Once the images were successfully downloaded, the next stage was to manipulate the images so as to have them in the best condition to yield the best OCR results. This was also to identify some techniques that could be generalised so as to eventually automate this stage.

The software which I used here was a product called PhotoHelper. PhotoHelper is a freeware image viewer with an attractive ease of use and graphics manipulation capabilities to try and alleviate some of the challenges faced by OCR recognition of CD covers as looked at in section 2.5.

It was simple to use and could easily perform the following manipulations:

- Zoom
- Decimate
- Mirror
- Rotate
- Shear
- Resize
- Blur
- Filters (Prewit, Sobel, Laplace, Gauss,..)
- Edge detection
- Morphological operations (erode, dilate, close, open)
- Noise addition
- Effects (concavity, fish eye, explosion,..)
- Bit depth conversion
- Invert
4 PRE-PROCESSING

- Histogram equalization
- Contrast enhancement
- Brightness adjustment

Of these, the ones I availed of with the CDs (See Appendix B) were filters, invert, bit depth conversion, contrast and brightness adjustments.

Below are examples of some of the conversions that were done. In general, the approach taken was that if the text was a bright colour then the first thing to do was to invert the colours.

![Figure 8: Example of Inverting CD Cover Colours](image)

Next, if the image was not sharp enough, the median filter setting was applied. This was particularly useful for images taken from mobile phones as noise was usually introduced due to the low quality of the camera.

Finally, the image was converted to black and white or grayscale if there were still too many colours and/or pictures in the background. If the image was converted to black and white, the threshold could be adjusted depending on the image. This is also known as the intensity threshold and is the relation of black pixels to white pixels. This is initially set to 128 but can be increased so as to have more black pixels thus in some cases making text more visible and vice versa this threshold can be decreased so as to have less black pixels thus not having the text look too blurred.
4.6 Optical Character Recognition

In this section, I will provide a review of the Optical Character Recognition software which I considered. Also, I will look at the workings of the software I decided upon and finally the problems encountered using the OCR software. I will also link back to chapter 2 the technologies that these systems make use of.

When deciding on the OCR software to use, I examined the software currently licensed by the college and any freeware products which were available.
I have narrowed this down to the following software choices which are ABBYY FineReader 8.0 Professional, SimpleOCR and GOCR.

4.6.1 ABBYY FineReader 8.0 Professional

FineReader takes images from a variety of sources, including PDF and turns them into editable text. It has to be said this software is fast and quite accurate. The User Interface design is very intuitive. The main screen shows thumbnails of pages on the left, a graphic image of the currently selected page in the middle, and an editing pane with recognised text on the right. The speed is impressive running OCR on images in less than 5 seconds. It also does the recognition in blocks so it correctly identifies columns of text so as to avoid mixing multiple columns. This eliminates one major challenge of OCR processing as identified in section 2.5. Text blocks can also be added or removed easily by the user to finetune the process manually.

The first step of the Abby process evaluates the integrity of the image being scanned, i.e., it looks at the character, word, and sentence as a whole. In the next step, it seeks out the purposefulness of the text and examines different possibilities to build a hypothesis. The hypotheses are analysed in context and verified by use of OCR dictionaries made up of Language Models. Finally, the system adapts to what it has learned. This corresponds well to the literature review in chapter 2.

4.6.2 SimpleOCR

SimpleOCR is a freeware OCR application. The initial start up provides an option to scanning either machine print text or handwritten text. It can also be performed on 4 different languages namely English(UK), English (UK), Dutch and French.
The user interface is very simple to use and can import a number of images at a time. There are however quite a few limitations with the program. Images are automatically converted to black and white when they are imported so this can straight away undo the pre-processing techniques done in the previous stage. This is because it seems to introduce noise that wasn’t previously there. Perhaps this is due to the threshold level that they are using which is much higher that what was previously used. Several factors can also cause the program’s analysis to go haywire including words in all capital letters frequently come out garbled, and the program often reads multi-column images in straight horizontal fashion, across the page, yielding unusable results. Again, this corresponds well to the predicted challenges looked at in section 2.5. It does offer an easy-to-use system for correcting errors with suggestions from its dictionary but the suggestions generally weren’t very relevant. There was also an option to select areas of the image which should be excluded from the OCR processing. Also, there is no training phase so the results don’t improve with experience. The SimpleOCR system is implemented using a template matching approach as reviewed in chapter 2. The character set is also limited reducing the quality of the OCR results.
4.6.3 GOCR

GOCR is an OCR program, developed under the GNU Public License. It converts scanned images of text back to text files. GOCR can be used with many front-ends, which makes it easy to port to different Operating Systems and architectures. It can open many different image formats, and its quality has been improving in recent years.

It is a command line program that allows characters from an image file to be recognised. This front-end for GOCR is graphical and is designed to be simple to use, just select the graphics file that you want to recognise, and the output will be shown to you. You can then copy this output and paste it to somewhere useful.

The GOCR system is also implemented using a template matching approach as reviewed in chapter 2.

The software can also be fine-tuned using various option settings.

4.6.4 Results

The ABBY FineReader provided excellent results, was easy to use and provided other very useful features. Also, it was already licensed by the college.

The GOCR software was relatively easy to use. It was just a matter of giving the file path to the software, tweaking with the settings and then examining the output. However, the results were very disappointing. In all cases, the results were not comparable to what was expected and in some cases were complete gibberish or nothing was returned at all. In general it seemed to perform better on plain scanned documents with no images or background colours.

The SimpleOCR offered an easy to use interface. Unfortunately, images were automatically converted to black and white before the OCR process which introduce noise which was not previously there. Also, information
4 PRE-PROCESSING

4.7 Problems Still Encountered by OCR

While the results were good from the OCR, they were by no means perfect. Some of the problems that were in encountered were with the recognition of brackets. Often curly and braced brackets were confused. Also, the number ‘1’, the letter ‘i’ and the letter ‘l’ were used interchangeably. This also occurred with the letter ‘o’ and the number ‘0’. Other character problems included ‘D’ for ‘O’ or ‘h’ for ‘ri’. This is very much what was expected based on the background reading.

Of course, text which had a cursive style font was impossible to recognise. These results were not recognisable to anything on the CD cover. Also, dark text on a very dark background was not picked up and similarly light text on a light background was not recognised.

Depending on the quality of an image the recognition rate varied from 80% to 98% for a good quality image to 0% to 40% for a poor quality image. The quality of these results will be looked at further in section 5.2.

Finally, there were lots of random characters thrown in at various points but this shouldn’t be a problem as they will be parsed out at the next stage.
4.8 Conclusion

In this chapter, a comprehensive account of the pre-processing stages was given. This included both a review of the hardware used to perform the scanning, the software used and techniques applied for manipulating the images, a review of the OCR software and finally a look at problems which were still encountered by the OCR software. It must be pointed out that the goal of this project was not for error correction but that given off the shelf perfect OCR, that it would be possible to guess where titles stop and start. An evaluation of pretend OCR output which is essentially perfect is given as input it the system. This can be seen in chapter 6. In this chapter the various image manipulation processes were all new to me. Also, the area of mobile technologies is an area which I explored as it was not an aspect of the CSLL syllabus.
5 Implementation

5.1 Introduction
In this chapter, the implementation stage of the project will be dealt with. First, a look at the results of the OCR output will be carried out. Next, the motivations for implementing the system in Perl will be highlighted as well as a look at MySQL and the database design and implementation. Next, an outline of the first parse module will be given. A description of the two music database resources will also be provided. Subsequently, an explanation of the bigram technique will be given followed by a look at possible smoothing algorithm implementations. Finally, the modules which calculate the probability and then guess the title will be explored.

5.2 Results of the OCR Data
The quality of the resulting text files varied considerably. As discussed in the previous chapter, there were a number of reasons for this. If one was to divide it into bad, good and excellent, it would look something like below.

5.2.1 Bad
As can be seen from the example in figure 16 below the problems encountered of running OCR, on the image in figure 15, are as discussed in the previous chapter — the multi-column layout, the small text on a varied background and the various copyright and production information. This is pretty much useless to predict and kind of song titles.
Figure 15: Sample Image Producing Bad OCR Results

Figure 16: Bad OCR Results
5.2.2 Good

As can be seen in figure 18 the results, from the input in figure 17, are quite good. With regards to the track titles, the information has been recognised well apart from one word space for track 8. There is however quite a bit of superfluous information some of which will be easy to remove by the parsing and other words which will not be as easy to categorise as titles words e.g. “Public Performance”.

Figure 17: Sample Image Producing Good OCR Results
5 IMPLEMENTATION

5.2.3 Excellent

As can be seen in figure 20 the results, from the input in figure 19, are excellent. Nearly all tracks have been identified perfectly. The superfluous information is in the form of numbers and gibberish which should be easily removed.
5 IMPLEMENTATION

Figure 19: Sample Image Producing Excellent OCR Results

Figure 20: Excellent OCR Results
5 IMPLEMENTATION

5.3 Why use Perl?

According to an official Perl web site, Perl\(^9\) is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming and GUI development.

Perl, also called ‘Practical Extraction and Report Language’ is a dynamic procedural programming language and was designed by Larry Wall in 1987. Wall, as a linguist, used his knowledge extensively in the design and development of the language. It is because of this that Perl remains an attractive tool to many for natural language processing as it offers a number of built-in functions for character string manipulation. Mastering these functions makes it possible to write string code quickly and concisely. Thus, it is the ideal choice to accomplish OCR correction.

Perl has many varied applications including its web use for writing CGI scripts. It is often used as a “glue language”, tying together systems and interfaces that were not specifically designed to interoperate. Finally, it is widely used in finance and bioinformatics mainly for its ability to handle large data sets, and the availability of many standard and third-party modules.

Of course, Perl isn’t the ultimate programming language and there are criticisms of it including that it produces ugly code with harsh looking syntax. Another criticism is that Perl is excessively complex and compact, and that it leads to “write-only” code that is virtually impossible to understand after it has been written. The language is intended to be easy to use and efficient rather than beautiful, small and elegant.

5.3.1 Regular Expressions

The Perl language includes a specialised syntax for writing regular expressions, and the interpreter contains an engine for matching strings to regular expressions (Deitel & Deitel, 2001).

Below is a brief look at some of the useful operators which were used extensively in my project. The m//(match) operator introduces a regular expression match. In the simplest case, an expression evaluates to true if and only if the string \$x matches the regular expression abc.

\[
\$x =~ m/abc/\]

\(^9\)Perl was originally named “Pearl”, after “the pearl of great price” of Matthew 13:46. Larry Wall wanted to give the language a short name with positive connotations; he claims that he looked at (and rejected) every three- and four-letter word in the dictionary. Wall discovered before the language’s official release that there was already a programming language named PEARL and changed the spelling of the name.
Portions of a regular expression may be enclosed in parentheses; correspon-
ding portions of a matching string are captured. Captured strings are assigned
to the sequential built-in variables $1$, $2$, $3$, ..., and a list of captured
strings is returned as the value of the match.

```perl
\$x =~ m/a(.)c/;  \# capture the character between ‘a’ and ‘c’
```

The `s///` (substitute) operator specifies a search and replace operation:

```perl
\$x =~ s/TTITLE/title/;  \# Change TTITLE to title
```

Regular expressions can also take modifiers. These are single-letter suffixes
that modify the meaning of the expression:

```perl
\$x =~ m/song/i;  \# case-insensitive pattern match
\$x =~ s/TTITLE/title/g;  \# global search and replace
```

### 5.3.2 Perl DBI

In order for a Perl program to communicate with the MySQL database, a DBI
(Database Interface) had to be used as discussed by Wall, Christiansen, and
Orwant (2000). It is necessary to know this as it has to be used to connect
to the database described in 3.3.7. The DBI has a very simple interface for
saying what SQL queries you want to make, and for getting the results back.
The DBI doesn’t know how to talk to any particular database, but it does
know how to locate and load in DBD (Database Driver) modules. The DBD
modules have the vendor libraries in them and know how to talk to the real
database. When you ask DBI to make a query for you, it sends the query
to the appropriate DBD module, communicates with the real database and
when it gets the results back, it passes them to DBI. Then DBI gives you the
results. To communicate with the database in Perl, a database handle needs
to be opened so as to let the program know the details of the database type,
server name, database name, username and password.

```perl
($db_handle = DBI->connect("dbi:mysql:database=kilkennp_db2;
host=mysql.cs.tcd.ie;user=kilkennp;password=XXXXXXX")
or die "Couldn’t connect to database: $DBI::errstr\n");
```

Figure 21 illustrates the communication between the Perl program and
the MySQL database taking place.
5 IMPLEMENTATION

5.4 MySQL

MySQL is an open source relational database management system (RDBMS) that uses Structured Query Language (SQL) for adding, accessing, and processing data in a database. Because it is open source, anyone can download MySQL and tailor it to their needs in accordance with the general public license. MySQL is noted mainly for its speed, reliability, and flexibility. It runs on many different platforms (Windows, Linux, Mac OS X, Unix, etc). The version of MySQL used in this implementation (4.0.24) was available on the TCD macneill server.

As mentioned above, MySQL is an implementation of the relational database model whereby relations are organised into tables. This works well for a music database because of the relation that exists between album names, artists, track titles, genre etc.

The structure of the MySQL server is so that it consists of a number of databases, each database consisting of a number of tables. A table consists of rows and columns, it can be empty but must have at least one field associated with it i.e. at least one column must be created.

Various commands can be executed on a table including INSERT, ALTER and UPDATE. SELECT can also be used to query the database and retrieve necessary information.

5.5 Table Set up

The normalised database is as in figure 22 below. There are 3 tables containing different information. Each table has PRIMARY KEYS (where one or more columns of a table uniquely identify every row in that table) and FOREIGN KEYS (where a column in a table does not uniquely identify rows in that table, but is used as a link to matching columns in other tables).

The first table called ALBUM contains information like the name of the album, the year it was released, the album genre, a unique disk id and an auto incremented id. The second table SONG contains information about a song namely the song name, its tracknumber on an album (if available) and an auto incremented id which acts as the primary key. The third table is

![Diagram of communication between Perl and MySQL database](image-url)
called ARTIST and contains information about the artist name and assigns an auto incremented id as the primary key.

The primary key can never be null. However, for other information where it is not possible to fill it in, this will contain a Null value. The id, track-number and year values are all INTEGER while the other variables are of type VARCHAR which is the MySQL equivalent of a String. The length for each VARCHAR column is set to 160 which is still well above the length of any songtitle\(^\text{10}\).

The arrows indicate the constraints and relations between the tables. An album can have many songs. Similarly, an artist can have many albums and an album may also have many artists. Finally, an artist may have many songs.

Figure 22: Entity Relation Diagram of MyDatabase

5.6 First Parse

As CD covers sometimes have the track number directly before the track title, it was decided that this first parse would check if there was a track number there. This is a similar approach as taken with the structural analysis of business cards and documents in section 2.6.11. It was believed if the information is there, then it should be kept rather than losing it now and looking it up later.

The method for this was to check if the line started with a number and if it did then it would send the track number to a database assuming that

\(^{10}\text{The longest know song title is “I’m a Cranky Old Yank in a Clanky Old Tank on the Streets of Yokohama with my Honolulu Mama Doin’ Those Beat-o, Beat-o Flat-On-My-Seat-o, Hirohito Blues” written by Hoagy Carmichael in 1943.}\)
the remainder of the line is a track title. Similarly, if there was no number at the start of the line then the entire line was sent to the track title column of the database.

\[(\text{Line}) \rightarrow (\text{tracknumber}) (\text{tracktitle})\]

\[(\text{Line}) \rightarrow (\text{tracktitle})\]

An example result of a successful parse can be seen in Figure 23.

![Figure 23: Successful result at Parse 1](image)

### 5.7 Getting the information from the Database

```perl
my $sth = $db_handle->prepare('SELECT name FROM song')
    or die "Couldn't prepare statement: " . $db_handle->errstr;
$sth->execute();
while(my $ref = $sth->fetchrow_hashref())
{
    print OUT "$ref->{'name'}\n";
}
```

This module opened a database handle and got everything from the name field in the song table and printing it to a file. This involved preparing an SQL query and executing it as below.

The resulting output from this module would be like in figure 24.
5 IMPLEMENTATION

5.8 Deciding on a Music Resource

5.8.1 Compact Disc Database (CDDB)

CDDB is a database for software applications to look up audio CD information over the Internet. This is performed by a client which calculates a (nearly) unique disc ID and then queries the database. The client is then able to display the artist name, CD title, track list and other additional information.

The database is used primarily by Microsoft’s media player and CD ripper software.

CDDB was originally developed under the GNU General Public License, and so many people submitted CD information for free. Later, however, the project was sold and the license conditions were changed thus making it no longer a free service.

CDDB was developed for recognising entire CDs. This is due to the identification algorithm it implements whereby the process involves creating a ‘discid’ of a CD created by performing calculations on the track duration information stored on the CD. This discid is used with the Internet database, either to download track names for the whole CD or to submit track names for a newly-identified CD.

It is claimed that the database contains over 4 million CDs spanning over 80 languages.

For the purposes of this project, the database would contain a large
amount of information relating to track titles, artist names etc. but would also hold a lot of unrequired extra information.

5.8.2 freeDB

Another resource is freedb which is a database of compact disc track listings where all the content is under the GNU General Public License. It was originally based on the now commercial CDDB as described above. It is based on the exact same principle for looking up CD information over the Internet as for CDDB.

Essentially freeDB and CDDB are the same with a few exceptions. These include:

- That a license fee must be paid to use the CDDB service.
- The CDDB is continuously maintained and updated and currently has of 2 million CDs.
- CDDB provides faster searching and easier integration with other applications.

For the purpose of this project, the freeDB provided more than enough track titles to build a language model so it was decided to go with this option.

5.9 Problems with freeDB

The first challenge was in getting the freedb database downloaded. The database was huge and initially took hours to download. There was also the problem of where to store the database when it was downloaded. This was resolved thanks to the Computer Science technicians who allotted space for it on the server. The next thing was to figure out how to access this database. What was downloaded and eventually unzipped contained folders with genre names which each contained thousands of entries for CDs. It was decided to concatenate all these files in to one large file which eventually amounted to 1.8GB. Of course, it was not possible to open this file in a text editor. However, it was feasible to view it using the ‘more’ command in Unix. This provided an insight into how the file was structured thereby seeing what information was there and what we needed etc. An entry for an album was as shown below.

```bash
# xmcd CD database file
# Copyright (C) 1995-1998 Ti Kan
#```
5 IMPLEMENTATION

# Track frame offsets:
#  150
#  16220
#  26977
#  33497
#  49575
#  69232
#  116972
#  132072
#  143862
#  154775
#  169232
#  192342
#
# Disc length: 2875 seconds
#
# Revision: 1
# Processed by: cddbd v1.4PL0 Copyright (c) Steve Scherf et al.
# Submitted via: CDex 1.40Beta3
#
DISCID=a50b390c
DTITLE=Various Artists / Put your Tongue to the Rail (Disc 1)
DYEAR=1987
DGENRE=Data
TTITLE0=Nothing Is True - Mae Pang
TTITLE1=It’s too Late - Psyclone Rangers
TTITLE2=Three Sisters - Thorazine
TTITLE3=People Who Died - Bootleg Thorazine Nightmare
TTITLE4=Catholic Boy - Bottom
TTITLE5=City Drops into the Night - Stomaboy w/ Marrow
TTITLE6=I Want the Angel - Jen Hess
TTITLE7=Day and Night - Iota
TTITLE8=Differing Touch - Marah
TTITLE9=Crow - Del Pess
TTITLE10=Plain Division - Mia Johnson
TTITLE11=Wicked Gravity - Sensitive Pricks
EXTD= ID3G: 254
EXTT0=
EXTT1=
EXTT2=
EXTT3=
As can be seen from the entry above, there was a lot of superfluous information. It was thus decided to do two things. Firstly, the file could be stripped down to just contain just DISCID, DTITLE, DYEAR, DGENRE and of course TTITLE.

It was also decided to put this information into a MySQL database for later use so that it would be more easily accessible, searchable and the information wouldn’t all be contained together. The same design as for MyDatabase was used here.

The resulting file contained disc information which had entries as below

DISCID=a50b390c
DTITLE=Various Artists / Put your Tongue to the Rail (Disc 1)
DYEAR=1987
DGENRE=Data
TTITLE0=Nothing Is True - Mae Pang
TTITLE1=It’s too Late - Psyclone Rangers
TTITLE2=Three Sisters - Thorazine
TTITLE3=People Who Died - Bootleg Thorazine Nightmare
TTITLE4=Catholic Boy - Bottom
TTITLE5=City Drops into the Night - Stomaboy w/ Marrow
TTITLE6=I Want the Angel - Jen Hess
TTITLE7=Day and Night - Iota
TTITLE8=Differing Touch - Marah
TTITLE9=Crow - Del Pess
TTITLE10=Plain Division - Mia Johnson
TTITLE11=Wicked Gravity - Sensitive Pricks

Finally, the file was boiled down just to track titles so as to be able to build the language model. The resulting file now contained all song titles from the database.

Nothing Is True - Mae Pang
5 IMPLEMENTATION

Another problem with the freedb was that because it wasn’t updated and maintained on a regular basis that as a result it contained errors and indeed a lot duplicate information. These errors included spelling errors and a non-standard way for entering CD entries. Due to the immense size of the database it was not feasible to develop an automatic process of correcting the database. However, as this is a statistical language model, the frequencies for the misspellings should filter to the bottom of the language model and not affect the overall probability. The section describing the language model belows looks at this further.

5.10 Language Model

Statistical language models are probability distributions defined on sequences of words. Language modelling has been used in many NLP applications such as part-of-speech tagging, parsing, speech recognition, machine translation and information retrieval.

In order to statistically guess if a word is part of a track title, a model of all track titles in the freedb database was constructed. Language models are particularly useful as they hold a key to the language, information contained within each model gives each language its own unique characteristics.

5.11 Bigrams

Bigrams are the character pairs in a text file. They are central to the process of constructing this language model. A word such as “database” has 7 bigrams, they are “da”, “at”, “ta”, “ab”, “as” and “se”.

By counting all the bigrams in a corpus (a corpus of track titles in this case), but also taking a note of the number of like-bigrams as well, the relative frequency for each bigram can be calculated. For example, if the bigram “an”
1,542 times in a corpus where the total number of bigrams was 958,993, then
the frequency of “an” relative to the total is \(\frac{1,542}{958,993}\) = 0.00160793668557466.
This is an estimate of the relative frequency of a bigram.

This model was again built by writing a Perl module. The Perl subroutine that creates the language model makes use of the `open()` function which takes two arguments, a file handle and a string representing the file. The file handle here is a type of buffer which reads in the file segment by segment and not in an entire chunk. Care must be taken when using a line-by-line procedure as the problem of the return line may arise. To avoid this, a function called `chomp()` is provided by Perl to remove this end of line character. The file containing the track titles converts each title to lowercase, strips out punctuation and puts each word of the string (i.e. title) into an array. Each word in this array is split into its respective bigrams, keeping track of its frequency and indeed the overall bigram frequency in the file. The relative frequencies are then calculated and finally the bigrams with their corresponding relative frequencies are stored in a hash table.

```perl
chomp $line; #remove end of line character
$line =~ lc($line); #convert to lowercase
$line =~ s/\s//g; #remove the white spaces
$line =~ s/[-'(),;:!\?\.]{1,}//g; #remove the punctuation

#wordline contains each word of the string
@wordline = $line;

foreach $word (@wordline)
{
    #letterline contains just a letter
    @letterline = split //, $word;

    for($i=0; $i <= $#letterline; $i++)
    {
        if($i < $#letterline )
        {
            $bigram = $letterline[$i] . $letterline[$i+1]; #bigram is the two characters from i to i+1
            $bigramfrequency{$bigram}++; #increment count of the bigram and put it into a hash
            $totalbigram++; #increment the overall total number of bigrams
        } #End if
    } #End for i
} #End for each word
```

The snapshot of the resulting language model can be seen in figure 25.
5 IMPLEMENTATION

5.12 Smoothing the Language Model

One major problem with standard N-gram models is that they must be trained from some corpus and because any corpus is finite, some n-grams can be missing from language model. So, it is possible that some bigram cases could have a ‘zero probability’. In multiplicative combinations with larger words, these zero out the whole word.

However, there are some techniques which can be used to assign a non-zero probability to these “zero probability bigrams”. This task of re-evaluating some of the zero-probability and low-probability, and assigning them non-zero values, is called smoothing.

In this section, a review of some popular smoothing algorithms for N-grams is looked at as well as the actual smoothing technique we implemented ourselves.

5.12.1 Add-One Smoothing

The simplest smoothing technique is the add-one method. This approach first adds one to all of the counts before normalizing into probabilities. So, the normal n-gram probability becomes a count of the bigrams plus one over the total number of bigrams plus the size of the vocabulary.
The advantage of using the add-one smoothing technique is that it is very easy to implement. However, using this method is not accurate and shifts too much probability mass towards rare n-grams. Also, the probability of frequent n-grams is underestimated and conversely the probability of rare n-grams is overestimated. Furthermore, all unseen n-grams are smoothed in the same way.

### 5.12.2 Witten-Bell Discounting

This is a better smoothing algorithm which is closely related to the add-one technique. Witten-Bell discounting is based on keeping track of zero-frequency events. If one thinks of a zero frequency n-gram as one that just hasn’t happened yet but that when it does happen it will be the first time the new N-gram has been seen. So the probability of seeing a zero-frequency N-gram can be modelled by the probability of seeing an N-gram for the first time. Essentially, based on the count of things seen once, use this to help estimate the count of things that have never been seen.

This method of discounting provides rather good estimates the problem is that if an n-gram occurrence did not appear in the training data the smoothed probability is still zero.

### 5.12.3 Good-Turing Discounting

The main idea of Good-Turing smoothing is to re-estimate the amount of probability weighting to assign to zero or low counts by looking at the number of N-grams with higher counts. So, let $N_c$ be the number of N-grams that occur $c$ times. It applies the idea to smoothing the joint probability of bigrams, $N_0$ is the number of bigrams $b$ of count 0, $N_1$ is the number of bigrams $b$ with count 1, etc. The revised count is:

$$C^* = (C + 1) \frac{N_{c+1}}{N_c}$$

The Good-Turing discounting works very well in practice. Usually, the discounted estimate, $c^*$, is used only for unreliable counts (e.g. $\leq 5$). As with other discounting methods, it is the norm to treat Ngrams with low counts (e.g. counts of 1) as if the count was 0.

### 5.12.4 Our Smoothing Technique

The previously discussed smoothing techniques are all very interesting but it was not possible to apply them to our language model as they were built...
around a different model. Their techniques were based on the probability of some event occurring given another event whereas we wanted the probability that in a training set one of those events didn’t exist.

The approach we took was first to have a value for a very rare event. This initial value was easy to get just by sorting the current language model by relative frequency and taking the value with the lowest relative frequency as in figure 26.

When this lowest frequency value was determined it was also decided to be the value that would be assigned to rare occurring bigram i.e. one not currently in the model. As a result, it was necessary to distribute this value amongst all the other values in the model to maintain consistency. Finally, it is also taken into account is a rare bigram occurs more than once in a word by spreading the probability on that word depending on the number of occurrences where \(1 \ldots n\) are known and \(n + 1 \ldots m\) are unknown. This is different to the add-one smoothing with respect to the fact that our probability space sums up to one.
So, the probability formula to calculate this would be:

\[
P(W_j) = \begin{cases} 
  j \leq n, n < j \leq m & \frac{\sum_j C(w_n)}{\min \left( \frac{\sum_j C(w_n)}{\sum_j C(w_j)} \right)} \\
  j \leq m 
\end{cases}
\]

### 5.13 Calculating the Probability of the OCR Results

Once the language model had been built the next stage was to process the results from the OCR to determine the probability of a word been in a song title. This was done in a number of steps. First, the smoothed language model and a file containing the OCR results were passed in as command line arguments. The first few stages of the processing are quite similar to that of building the language model, i.e. the strings are converted to lowercase characters, all punctuation is removed, each string is split into words and the respective bigrams are constructed. Their relative frequency is not however required.

The next routine looks up the relative frequency for each bigram from the language model created. However, if there is an occurrence of a bigram that is not in the language model then the relative frequency value assigned to this is a very small value which had been calculated earlier to factor in the rarity of an occurrence of this bigram. Otherwise, a value of zero would have been assigned which under multiplication laws would have the resulting product set to zero. There is also a count kept of the frequency of this rare value so if it occurs more than once the probability is evenly distributed throughout the probability of a word.

To check if a word can be legitimately considered as being a word from a song title the bigram probability must first be calculated. So, for example, a word like “database” is determined by splitting the word up into its bigrams and then finding the corresponding frequencies of these from the language model as in the table below.

<table>
<thead>
<tr>
<th>CD Title Language Model</th>
<th>Bigram Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>da</td>
<td>0.005284533</td>
</tr>
<tr>
<td>at</td>
<td>0.008272473</td>
</tr>
<tr>
<td>ta</td>
<td>0.006576437</td>
</tr>
<tr>
<td>ab</td>
<td>0.00203819</td>
</tr>
<tr>
<td>ba</td>
<td>0.00422132</td>
</tr>
<tr>
<td>as</td>
<td>0.00571648</td>
</tr>
<tr>
<td>se</td>
<td>0.006002179</td>
</tr>
</tbody>
</table>
In order to calculate a probability estimate for “database” using the
derived language model, it is necessary to multiply the probability of each of
the bigrams as in figure 27.

\[ P(\text{database}) = P(da) \times P(at) \times P(ta) \times P(ab) \times P(ba) \times P(as) \times P(se) \]

Figure 27: Probability Formula

There is of course a problem of calculating the probability in this man-
ner. As the words get longer, the estimate is lowered. Because this unfairly
dismisses long words, it is necessary to calculate the \( n^{th} \) root of the result
where \( n \) is the number of bigrams in the word as in figure 28.

\[ \sqrt[n]{P(da) \times P(at) \times P(ta) \times P(ab) \times P(ba) \times P(as) \times P(se)} \]

Figure 28: \( n^{th} \) Root Probability

As can be seen in the code below, the subroutine `getProbability` imple-
ments this as below. Again, this loops through an array of bigram frequencies
for a word taking into account any rare occurrences of bigrams and then gets
the \( n^{th} \) root of the running total.

```perl
sub getProbability
{
  my $run_total = 1;
  my $nth_root = 1;

  foreach (@bifrequency)
  {
    $run_total *= $_;
  }
  $run_total = ($run_total / $rarefreq);
  $nth_root = scalar(@bifrequency);
  my $prob = $run_total ** (1/$nth_root);
  return $prob;
}
```

The output of this subroutine will basically be a list of words with a
corresponding probability as in figure 29.
<table>
<thead>
<tr>
<th>Word</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>sky</td>
<td>0.000617</td>
</tr>
<tr>
<td>hen</td>
<td>0.002349</td>
</tr>
<tr>
<td>ron</td>
<td>1.383825e-009</td>
</tr>
<tr>
<td>little</td>
<td>0.004562</td>
</tr>
<tr>
<td>nudge</td>
<td>0.002682</td>
</tr>
<tr>
<td>shop</td>
<td>1.383825e-009</td>
</tr>
<tr>
<td>buck</td>
<td>0.004967</td>
</tr>
<tr>
<td>irthe</td>
<td>0.004967</td>
</tr>
<tr>
<td>corn</td>
<td>0.004681</td>
</tr>
<tr>
<td>pany</td>
<td>0.003556</td>
</tr>
<tr>
<td>express</td>
<td>0.002221</td>
</tr>
<tr>
<td>on</td>
<td>1.383825e-009</td>
</tr>
<tr>
<td>id</td>
<td>0.002291</td>
</tr>
<tr>
<td>jee</td>
<td>0.001232</td>
</tr>
<tr>
<td>clark</td>
<td>0.003260</td>
</tr>
<tr>
<td>06</td>
<td>1.383825e-009</td>
</tr>
<tr>
<td>15</td>
<td>1.383825e-009</td>
</tr>
<tr>
<td>right</td>
<td>0.002973</td>
</tr>
<tr>
<td>more</td>
<td>0.008656</td>
</tr>
<tr>
<td>miles</td>
<td>0.007671</td>
</tr>
<tr>
<td>to</td>
<td>0.008042</td>
</tr>
<tr>
<td>louisville</td>
<td>0.003662</td>
</tr>
<tr>
<td>03</td>
<td>1.383825e-009</td>
</tr>
<tr>
<td>arewell</td>
<td>0.005308</td>
</tr>
<tr>
<td>blues</td>
<td>0.003943</td>
</tr>
<tr>
<td>18</td>
<td>0.007953</td>
</tr>
<tr>
<td>riding</td>
<td>0.007274</td>
</tr>
<tr>
<td>the</td>
<td>0.016121</td>
</tr>
<tr>
<td>waves</td>
<td>0.005259</td>
</tr>
<tr>
<td>fire</td>
<td>0.004272</td>
</tr>
<tr>
<td>on</td>
<td>0.016178</td>
</tr>
<tr>
<td>the</td>
<td>0.016121</td>
</tr>
</tbody>
</table>

Figure 29: Probability Word Results
5.14 Guess the Title

The next stage based on the probability values for each word was to guess the song titles. A module was written which took a file containing each word and its corresponding probabilities. If a word was equal to or above a certain threshold then it was put into an array and deemed to be part of a song title. This was done until a word was reached which was below the frequency then the title was printed. The array is emptied and the process starts again.

5.14.1 Defining the Threshold

When defining the threshold, it had to be strict enough so not to allow misspelled or incorrect words but also lax enough not to deem words which are correct to be misspelled or not part of the title. After examining the first few runs of data, it was decided that a threshold value of 0.002 would be reasonable.

The only exception to this rule was if the bigram was ‘a’ or ‘i’ as this had been assigned a value of zero. It could also be possible to introduce some form of an exception list to remove other nuisance words such as ‘copyright’ or ‘production’.

An example of the output from this stage can be seen in figure 30.

```
dueling banjos
i
maggie
ukin
pony express
ld
clark
ight more miles to louisville
farewell bit es
```

Figure 30: Probability Word Results

It must be reminded again at this stage that the goal of this project was not to correct the OCR errors but to correctly parse the track titles got from a CD cover using OCR.
5.15 Conclusion

This chapter has provided a comprehensive account of the implementation stage of the project. This ranged from the OCR output in the previous stage right through to the implementation of the various modules for parsing, building the language model, looking up bigrams, calculating the probability and finally guessing the title. An account of Perl and MySQL and the motivations for using them has been provided.

Finally, a lot of new areas have been explored and developed here including learning the Perl language and MySQL. This also provided me with an opportunity to use database design and implementation techniques learned in the fourth year Information Systems course. The building of language models was an area of Computational Linguistics I had a theoretical knowledge of but never a practical experience with until this project. Also, statistical and smoothing techniques were areas of Computational Linguistics which I had read about but never explored further.
6 Results and Evaluation

6.1 Introduction

The goal of this chapter is to conduct an evaluation of the system. It was decided to carry out three sets of tests on the system. The first two sets are of performance subject to degraded inputs from OCR while the third one highlights the ‘as good as it gets’ results. Firstly, using a sample of fifteen randomly selected CD covers (as in appendix B) which were scanned using a flat bed scanner, then pre-processed using the various techniques as discussed in chapter 2.6. The results of the OCR processing were then put through the proposed system.

Next, a sample of five CD covers were scanned using the mobile phone device, these were then pre-processed applying the techniques discussed in chapter F and then put through the proposed system.

Finally, 10 samples were taken from the freedb database as examples of CD covers. These were also put through the proposed system and metrics were calculated. This was to have a benchmark for the other results and to ensure the language model was accurate.

6.2 Analysis of Results

Two metrics were used to calculate the quality of the system output:

6.2.1 Recall

Recall is the ratio of the number of relevant track titles retrieved by the system to the total number of relevant track titles in the OCR data. It is usually expressed as a percentage. First, a measure of the total relevant track names in the OCR data was taken i.e. based on looking at the OCR data how many track titles could be identified. A measure of all the relevant track names identified by the proposed system was also taken. These two values were then compared and expressed as a percentage, thus, giving the metric of recall for the number of CD tracks correctly identified by the system.

\[
\text{Recall} = \frac{\text{Relevant Track Titles Retrieved By Proposed System}}{\text{Total Number of Relevant Track Titles from OCR}} * 100
\]

Figure 31: Recall for Proposed System
6.2.2 Precision

Precision is the ratio of the number of relevant tracks accurately identified to the total number of irrelevant and relevant tracks identified. Thus, this takes into account the number of strings incorrectly identified as being title tracks.

\[
\text{Precision} = \frac{\text{Track Titles Identified by Proposed System}}{\text{Track Titles Identified by Proposed System} + \text{incorrectly identified tracks}} \times 100
\]

Figure 32: Precision for Proposed System (%)

6.3 Sample Input/Output

Figure 33 is a sample of some OCR data that was sent into the system. Highlighted is what would be considered to be valid track titles.

```
LET IT BE, NAKED
GET BACK, 2 DIG A PONY
FOR YOU BLUE
THE LONG AND WINDING ROAD
TWO OF US
IVE GOT A FEELING
ONE AFTER 9098, DONT LET ME DOWN
LIVE MINUTE 10, ACROSS THE UNIVERSE
LLET IT BE
```

Figure 33: Sample OCR Data for Input

The resulting output is as in figure 34. Highlighted are the tracks which the system correctly parsed.
6.4 Results of Scanned CD Covers

The table in figure 35 shows the results which were calculated on a random sample of 15 CD covers. The CD cover images are in appendix B, the OCR data used as input is in appendix D and the resulting output can be seen in appendix E.
### Results and Evaluation

#### Test Number | % Precision | % Recall | Titles Identified | Correct Titles | Incorrectly Identified
---|---|---|---|---|---
Test 1 | 33.33% | 33.33% | 7 | 21 | 14
Test 2 | 68.75% | 84.62% | 11 | 13 | 5
Test 3 | 26.67% | 25.00% | 4 | 16 | 11
Test 4 | 63.64% | 70.00% | 7 | 10 | 4
Test 5 | 32.56% | 77.78% | 14 | 18 | 29
Test 6 | 37.50% | 81.82% | 9 | 11 | 15
Test 7 | 10.17% | 85.71% | 6 | 7 | 53
Test 8 | 43.75% | 63.64% | 7 | 11 | 9
Test 9 | 41.67% | 55.56% | 5 | 9 | 7
Test 10 | 48.00% | 66.67% | 12 | 18 | 13
Test 11 | 35.71% | 83.33% | 10 | 12 | 18
Test 12 | 60.71% | 72.34% | 34 | 47 | 22
Test 13 | 47.06% | 80.00% | 16 | 20 | 18
Test 14 | 22.73% | 83.33% | 5 | 6 | 17
Test 15 | 46.15% | 78.57% | 12 | 15 | 14
**Average** | **41.23%** | **69.45%** | | | |

Figure 35: Results from Scanned CD Covers

#### 6.5 Results of Mobile Phone CD Covers

The table in figure 36 shows the results which were calculated on a random sample of 5 CD covers which were scanned using a mobile phone as discussed in chapter 4.2.2. The CD covers are in appendix F, the OCR data used as input is in appendix G and the resulting output can be seen in appendix H.
### 6.6 Results of freedb Data

The table in figure 37 shows the results which were calculated on a random sample of 10 CD covers from the freedb music resource. The data used as input is in appendix I and the resulting output can be seen in appendix J.

<table>
<thead>
<tr>
<th>Test</th>
<th>% Precision</th>
<th>% Recall</th>
<th>Titles Identified</th>
<th>Correct Titles</th>
<th>Incorrectly Identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>56.25%</td>
<td>75.00%</td>
<td>9</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>Test 2</td>
<td>61.11%</td>
<td>73.33%</td>
<td>11</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>Test 3</td>
<td>76.92%</td>
<td>76.92%</td>
<td>10</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Test 4</td>
<td>82.35%</td>
<td>87.50%</td>
<td>14</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>Test 5</td>
<td>56.52%</td>
<td>72.22%</td>
<td>13</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>Test 6</td>
<td>53.85%</td>
<td>73.68%</td>
<td>14</td>
<td>19</td>
<td>12</td>
</tr>
<tr>
<td>Test 7</td>
<td>55.56%</td>
<td>71.43%</td>
<td>15</td>
<td>21</td>
<td>12</td>
</tr>
<tr>
<td>Test 8</td>
<td>56.25%</td>
<td>69.23%</td>
<td>9</td>
<td>13</td>
<td>7</td>
</tr>
<tr>
<td>Test 9</td>
<td>100.00%</td>
<td>100.00%</td>
<td>9</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>Test 10</td>
<td>37.93%</td>
<td>55.00%</td>
<td>11</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>63.67%</strong></td>
<td><strong>75.43%</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 37: Results of Processing freedb CD Covers

### 6.7 Discussion of Results

Overall, the results were largely as expected.

The results for the scanned CD covers are generally good with an average score of 69.45% for recall. However, precision is significantly lower with an
average of 41.23%. As well as this, if the results are examined closely it can be seen that they vary greatly across the tests carried out with precision ranging from a low of 10.17% up to 68.75%. Similarly, recall varies from a low of 25% up to a high of 84.62%. It seems fair to say that, in general, where recall is high that precision is contrastively low. This is due to the fact that while a high number of titles are correctly identified there are also a number of strings incorrectly identified as titles resulting in a lower precision rate. This was particularly the case in test 11 where 10 of the 12 track were correctly parsed but a lot of additional data was also identified to be track titles which resulted in low precision. This data was mainly artist and copyright information. While it was possible to limit some of this information by setting an upper bound value for the length of a track title, some information still got through the system as it had a high probability of been part of a track title given the language model used to guess what counts as a token in a track title. The worst results were for test 3 with a precision of 26.67% and recall of 25%. This seems to be because of a difficulty of identifying the track title markers due to no bulleting, numbering or other such formatting techniques in the OCR data which aid in separating the tracks. Also, as the words were correctly spelt common words, it was difficult to separate the titles.

The results for the mobile phone are similar to those for the scanner as in that when recall is high, the precision is lower. This is again similar to the reasons outlined above. The results are also worse because of the poor quality of the OCR data used as input.

The results for the freedb tracks are the highest across the three experiments and could be used as a benchmark when considering the previously discussed results. The best results are for test 9 where 100% was achieved for both metrics. The worst results were for test 10 which had scores of 37.93% and 55% for precision and recall respectively. This seems to be because there were misspelled (e.g. foolosophy) or rare words (e.g. Baxtor in the name Sophie Ellis Baxtor) that would have had a low probability result thus been below the threshold previously defined and then cutting the track title too soon.

6.8 Conclusion

Through the implementation of an n-gram based system as discussed in chapter 5, track titles from OCR results can now be parsed. While the results for the scanned items aren’t perfect, they are quite good, in particular, if the results are compared to that which were got from running sample freedb data through the system and using this as a benchmark to reach. However, if the freedb is considered the ‘best case’ performance of the system, then it
It seems clear that there is considerable room to make significant improvements on my method or to use simply better alternatives, whatever they might be. In section 7.4, I make some educated guesses about directions that might be worth following up on. Finally, this chapter has provided a comprehensive evaluation of the system using the different input methods i.e. scanner and mobile phone. As a result, problem areas which the system may encounter have been identified to be considered going forward.
Conclusion and Future Work

7.1 Summary

This project has successfully designed and implemented a system in Perl for parsing track titles from CD covers using statistical N-Gram techniques. The system can take OCR data as input and will then output what it believes to be the title tracks. An appropriate database has also been designed and implemented in MySQL to hold a music collection. A broad understanding of OCR software and possible pre and post processing techniques has been achieved as well as that of image manipulation techniques in preparation for the OCR processing.

Chapter 2 has provided a review of OCR research and development in the past 50 years. This covered different pre and post processing techniques as well as an explanation of which techniques best suited the goals of this project.

Chapter 3 gave a high-level design of the proposed system briefly describing each stage. The rationale of choosing the N-gram approach was also looked at.

Chapter 4 looks at the image manipulation software used and different pre-processing techniques employed to obtain the best quality image to use in the OCR processing of the project.

Chapter 5 provides a detailed look at the project implementation. This included a description of the use of Perl and MySQL in the project. The parsing stages are described in detail and also an evaluation of two potential music resources was provided. The building of the language model is looked at and an explanation of bigram and probability calculations was given. Finally, a number of smoothing techniques were looked at and the actual guessing of the track title was carried out.

Finally, Chapter 6 carried out the testing of the system from three different perspectives. Three tests are described using different sets of data - one for scanned images using the flatbed scanner, one using the camera on a mobile phone and one using random samples from the freedb resource.
7.2 Achievements

As mentioned in section 1.2, the aim of the project was to build a system to parse track titles from CD covers. I feel that this has been successfully implemented. The system can calculate the probability of a word being in a track title and thus break the titles on this. It is also successful, to an extent, in filtering out unwanted formatting characters, track lengths and long strings that couldn’t possibly be a title.

While the system is in no means perfect, I feel that, taking into consideration the time constraints on this project, I have achieved a lot and certainly done some good initial work for anyone that might want to develop the project further. I have advanced my knowledge immensely in the areas of optical character recognition, image analysis and the application of statistical techniques in natural language processing. I have also acquired a good knowledge of the Perl programming language and the use of wireless technologies for transferring data from a mobile phone to a PC.

The project has been rewarding and has shown me that determination and good research can make such a project happen. It has also allowed me an opportunity to use the knowledge gained throughout my undergraduate degree.

7.3 Problems Encountered

One of the main problems encountered while undertaking this project was working with the freedb music resource. The enormous size of the database (1.8Gb and over 2 million files) to download resulted in a number of failed attempts, on my behalf, to download and unzip the database. Eventually, with help from the technicians the database of files were successfully downloaded and unzipped. The file system which the database was using was different to that on the college network so the easiest thing to do was to concatenate the files together providing something more manageable to work with.

Initially, I also found that when starting to work with Perl that simple programming tasks were taking longer due to my lack of knowledge of the language.

With regards to the approach taken to the project, the main disadvantage is that a number of incorrect strings are being let through and deemed as track titles. Because these are such common words and could likely be part of a
song title, this will continue to be a problem. Also, after the parsing stage, all
titles that have been identified are without punctuation and capitalisation.
It is also unfortunate that track times and whatever other details available
cannot be extracted and used in a useful way.

7.4 Future Work

An automatic parser, such as the one described in this thesis, can always be
improved upon. As a result, the following section describes some possibilities
for future work.

7.4.1 Improving the Parser Quality

As mentioned in section 7.3, the system still has problems filtering out inap-
propriate data such as copyright and production information. One possible
way to solve this could be to use the extra information from the freedb data-
base to build another language model which could check the probability of
this “extra” information not being part of a track title and possibly part of
this extra production or copyright information. This could involve developing
some calculus approach to identify a song versus the probabilistic technique
used in this project.

7.4.2 Full Integration with a Mobile Phone Server

Unfortunately, due to time and money constraints, full integration of the
mobile phone server as described in section 4.4.3 was not possible. The
work around for this was to still implement the system with a mobile phone
interface but transferring the images was done via infrared as discussed in
section 4.4.1. I think this would be an extension which would reflect a real
world scenario and wouldn’t be too challenging to implement.

7.4.3 Automation and User Interface

It could be a possibility to bring together the modules I developed and have
an easy to use user interface. It may also be an idea to automate the whole
process more and integrate it with the OCR software.

7.4.4 Training the OCR Software

Another approach to improve the OCR results would be to develop some way
of training the OCR software using the information from the freedb resource.
As a result OCR recognition rates could be much higher. However, there will
still be the challenge that images from CD covers pose as described in section 2.5.

7.4.5 Partial Matching

It may be possible to introduce some partial matching approach as used in Scully (2005) which would correct noise introduced by the OCR process. This would take the parsed track and partial match it to its correct equivalent in the freedb resource.
References


A Code

See source code on accompanying CD.
B  CDs Scanned from Scanner

See images on accompanying CD.
C Pre-Processed CD Images from Scanner

See images on accompanying CD.
a big
love my wish game true
how and then there
a foot such as
need your love tonight
don
i beg of you
santa bring my baby back
santa
is back in town party paralyzed one night i got stung king creole wear
time mean woman blues
playing for keeps
hard headed woman
gotaloto
todo
ll
peace in the valley
atoiaa
i
ni
usa
FTEB MIDSIGHT
3. TRY AGAIN
4. TWO CIGARETTES A ASHTRAY
5. TH 5
7. COME ON IN (4 MAKE YOURSELFATjg"
8.1 DoR1 WANNA
9. YOUR CHV i ME AGAIN
11. THE MAN UPSTAIRS
12. STOP THE WORLD (& LET ME OFF)
13. d|
ME IN CARE OF TH ! BLUES
15. NO WHEELS ON THIS SHIP
16. HUNGRY FOR LOVE
17. WAlI

Figure 39: OCR Data from Test 2

FRAGILE,
A THOUSAND YEARS
PERFECT LOVE...GONE WRONG
ALL THIS TIME
THE HOUNDS OF WINTER
MAD ABOUT You
DON’T STAND So CLOSE To ME
WHEN WE DANCE
DlENDA ROXANNE
i,H” (IF You LOVE SOMEONE) SET THEM FREE
BRAND NEW DAY
FIELDS OF GOLD
MOON OVER BOURBON STREET
SHAPE OF MY HEART
IF r EVER LOSE MY FAITH IN You
EVERY BREATH You JAKE

Figure 40: OCR Data from Test 3
CD 32308
LU MAGGIE’S FARM
O ONE TOO MANY MORNINGS
[M STUCK INSIDE OF MOBILE WITH THE MEMPHIS BLUES AGAIN
GQ OH, SISTER
m LAY, LADY, LAY
H] SHELTER FROM THE STORM
YOU’RE A BIG GIRL NOW
(H I THREW IT ALL AWAY
GO IDIOT WIND
COLUMBIA
C 1976 Sony Music Entertainment Inc./'Cotumba" is the exclusive
trademark of Sony Music Entertainment Int/Distributed Sony Music

Figure 41: OCR Data from Test 4
DUELING BANjOS* 3:16  
LITTLE MAGGIE 1:12  
SHUCKINTHE CORN 2:12  
PONY EXPRESS 2:06  
OLD JOE CLARK 1:15  
EIGHT MORE MILES TO LOUISVILLE 2:03  
FAREWELL BLUES 2:00  
EARL'S BREAKDOWN 1:52  
END OF A DREAM* 1:49 10 BUFFALO GALS 2:18  
:  
REUBEN’S TRAIN 2:58  
RIDING THE WAVES 1:35  
FIRE ON THE MOUNTAIN 2:18  
EIGHTH OF JANUARY 1:07  
BUGLE CALL RAG 1:32  
HARD AINT IT HARD 1:50  
MOUNTAIN DEW 1:26  
RAWHIDE 2:05  
*Traditional, arranged by Eric Weissberg  
Performed by Eric Weissberg and Steve Mandell  
Eric Weissberg and Marshall Brickman performed all other selections  
which were previously released on the album New Dimensions in Banjo and Bluegrass (Elektra EKS 7238)  
7599-27268-2  
France WE 835  
*v  
"x  

iLC)0392  
*  
The music on this Compact Digital Disc was originally recorded on analog  
equipment We have attempted to preserve, as closely as possible, the  
sound of the original recording. Because of its high resolution, however,  
the Compact Disc can reveal limitations of the source tape.  
Bros. Records Inc. for the U.S. & WEA International Inc. for the world outside  
the U.S. 1973 Warner Bros. Records Inc. for the VS. Sc WEA International Inc.  
for the world outside the U.S. All Rights Reserved Mm m.m u red i n, G er m m *.  
Presse m \ \\cm mm - R econl Service G mfcft,- A Ssdor f. Q A ' Warner. Com m y  
v\cm ions Com pmy./ U n bis i hoiked Cop:mg, Hinngv Lending.'Public Performance  
Brmdcamng QtTfm Record Prohibited-!"'.

Figure 42: OCR Data from Test 5
THE BOY IN THE BUBBLE
GRACELAND
I KNOW WHAT I KNOW
GUMBOOTS
DIAMONDS ON THE SOLES OF HER SHOES
YOU CAN CALL ME AL
UNDER AFRICAN SKIES
HOMELESS
CRAZY LOVE. VOL II
THAT WAS YOUR MOTHER
ALL AROUND THE WORLD

OR
THE MYTH OF FINGERPRINTS
PRODUCED BY PAUL SIMON ENGINEER ROY HALEE

The music on this Compact Digital Disc was originally recorded on analog equipment. We have attempted to preserve, as closely as possible, the sound of the original recording. Because of its high resolution, however, the Compact Disc can reveal limitations of the source tape.

Warner Bros. Records Inc., a Warner Communications Company ©© 1986 Warner Bros Records Inc for the U.S & WEA International Inc for the world outside of the US 1986 Paul Simon All Rights Reserved. Unauthorized duplication is a violation of applicable laws. Manufactured in Germany by Record Service GmbH Alsdorf © Presse en Allemagne by Record Service GmbH. Alsdorf © a Warner Communications Company (LC)03ft2)

Figure 43: OCR Data from Test 6
WHO?

WHAT? ..

. GRIS-GRIS GUMBO YA YA
(By f> John Oeau* Time 5 34"

. OANSE KALINDA BA DOOM
(By Dr. John Craaux 4 Haroid Battiste T.me 3 44)

. MAMA ROUX
(By Craaux & Hill Time 2 55t

. DANSE FAMBEAUX
(By Or. John Creaux. Time. 4 63)

. CROKER COURTBULLION
(By Harold Battibte. Time 5 571

. JUMP STURDY
(By Or, John Creaux. Time 2191

. I WALK ON GUILDED SPLINTERS
(By Dr. John Creaux. Time 7 $7)

All three songs in this album are published by Masque - Jonarv. BMI

Cover & backliner photos: Raphael
Album design: Marvin Israel

ARRANGED & PRODUCED BY HARROLD BATTISTE

Figure 44: OCR Data from Test 7
IKOIKO (4:08)
- BLOW WIND BLOW (3:17)
- BIG CHIEF (3:25)
- SOMEBODY CHANGED THE LOCK (2:42)
- MESS AROUND (3:09)
- LET THE GOOD TIMES ROLL (3:56)
- JUNKO PARTNER (4:27)
- STACK-A-LEE (3:28)
- TIPITINA (2:04)
- THOSE LONELY LONELY NIGHTS (2:30)

a. HIGH BLOOD PRESSURE
b. DON’T YOU JUST KNOW IT
c. WELL I’LL BE JOHN BROWN

12. LITTLE LIZA JANE (2:59)
75678039829

Figure 45: OCR Data from Test 8
CD 32308

LU MAGGIE'S FARM

O ONE TOO MANY MORNINGS

[M STUCK INSIDE OF MOBILE WITH THE MEMPHIS BLUES AGAIN

GQ OH, SISTER

m LAY, LADY, LAY

H] SHELTER FROM THE STORM

YOU’RE A BIG GIRL NOW

(H I THREW IT ALL AWAY

GO IDIOT WIND

COLUMBIA

C 1976 Sony Music Entertainment Inc. ’'Cotumba" is the exclusive trademark of Sony Music Entertainment Inc./Distrittion Sony Music

Figure 46: OCR Data from Test 9
THIS RECORD MUST BE PLAYED LOUD
"I’M SO GOOD
THAT
I DON’T
HAVE
TO
Brag!"
SHEL SILVERSTEIN
SINGS HIS SONGS
Off The Record
. MODERN TALK
. I ONCE KNEW A WOMAN
. EVER LOVIN’ MACHINE*
. LOOKIN’ FOR MYSELF
. THE UGLIEST MAN IN TOWN*
. I’M SO GOOD THAT I DON’T HAVE TO BRAG
. THE MERMAID
. I CAN’T TOUCH THE SUN*
. PLASTIC*
. BETTER NOT ASK ME
. YOWSAH!
. LEMMEBESOMETHIN’
. TESTING THE BOMB
PERSONNEL:
SHEL SILVERSTEIN - GUITAR & VOCALS
MALCOLM HALE - GUITAR
ROBERT MATTHEWS - BASS
CHARLES WATSON - DRUMS
ROBERT SLAWSON - HARMONICA
LITTLE WALTER* - HARMONICA
ISSUED UNDER LICENCE FROM
UNIVERSAL MUSIC (UK) LTD.
1965 UNIVERSAL MUSIC.
2001 DEMON MUSIC GROUP LTD.
MANUFACTURED IN ENGLAND.
MARKETED BY DEMON / WESTSIDEj*
LONDON, NW6 1TB.
DEMON / WESTSIDE IS A DIVISION OF'
DEMON MUSIC GROUP LTD.
7.40155E+11

Figure 47: OCR Data from Test 10
LET IT BE... NAKED
I. GET BACK 2. DIG A PONY
. FOB YOU BLUE
. THE LONG AND WINDING MAD
. TWO OF US
. I'VE GOT A FEELING
. ONE AFTER 909 8. DON'T LET ME DOWN
. I ME MINE 10. ACROSS THE UNIVERSE
II. LET IT BE

* Additional Disc: FLY ON THE WALL
A unlgui insight into the Beatles a! work In rehearsi ind
in the studio durtng January 1969;
4
www.thebeatles.com
The copyright in this sound recordingis owned by Apple Corps Ltd./EMI
Records Ltd. This label copy information is the subject of copyright
protection. All rights reserved. # 2003 Apple Corps Ltd./EM1 Records Ltd.
Artwork 2003 Annie Coras Ltd. Printed in the EU. 07243 595713 24
7.2436E+11

Figure 48: OCR Data from Test 11
Invitation (:40)
  . Eighteen Flavors (-.45)
  . Melinda Mae (-.36)
  . Sick (1-.22)
  . Ickle Me, Pickle Me, Tickle Me Too (1:49)
  . Enter This Deserted House (1:08)
  . Jimmy Jet and His TV Set (1:08)
  . For Sale (:45)
  . Warning (-.38)
  . The Yipiyuk (1:40)
  . Crocodile’s Toothache (1:18)
  . Stone Telling (:29)
  . Ridiculous Rose Oil)
  . Boa Constrictor (1:12)
  . Peanut-Butter Sandwich (3:31)
  . Listen to the Mustn’ts (-.24)
  . Hug o’ War (-.23)
  . Smart (1:09)
  . Forgotten Language f.-56;
  . The Farmer and the queen (1-.06)
  . The One Who Stayed (1:17)
  . No Difference (-.46)
  . Wild Boar (-.31)
  . Thumbs (-.30)
  . Sarah Cynthia Sylvia Stout Would Not Take the Garbage Out (2:34)
  . My Hobby fl.-oo;
  . Early Bird (-.24)
  . Me-Stew (:49)
  . Captain Hook (:39)
  . With His Mouth Full of Food (1:37)
  . The Flying Festoon (-.49)
  . The Silver Fish (1:02)
  . The Generals (1:49)
ISBN 0-VaAT-DODI-?
7464-66079-2 3
  . The Worst (:31)
  . Dreadful (1:06)
  . My Beard (-.27) Previously unreleased bonus tracks:
  . The Little Blue Engine (1:03)
  . If I Had a Brontosaurus (-.12)
hel Silverstein is the universally loved, best-selling author and illustrator of A LIGHT IN THE ATTIC, THE GIVING TREE, FALLING UP and many other books of poems and stories. This newly remastered edition of the GRAMMY winning album based on the all-time favorite WHERE THE SIDEWALK ENDS features 11 previously unreleased tracks, complete lyrics, and new liner notes. A delight for all ages, it showcases the enduring legacy of Silverstein’s magical writing and vocal artistry, both in spoken word and song.

Figure 49: OCR Data from Test 12
DISC 1
SONGS FOR SWINGING CELIBATES
Sodom Tonight
Very Naughty Party
Galway Bay
Black Contact Lenses
Goodbye to Berlin
Love Theme From "Yeats: The Movie"
Still Life with Girl
Bouncing Off The Boulders
The Best Things in Life Are Mine

DISC 2
CHARM & ARROGANCE
Drown the Browns
Lost and Found
LSD (Isn’t What It Used to Be)
Stay Tonight
You Make Girls Unhappy
Abandon the Galleries
Here Comes the New Year
There Goes Everything
Go to Sleep
Charm & Arrogance
Some Drugs
You Can Always Go Home

All songs written by Collins / Farrell / Gough
Both albums digitally remastered by Neil Farrell
All photographs by Aengus McMahon (yes, even the ones he’s in) except where otherwise credited
Cover painting by Gareth "Napoleon" Allen, 2005
Layout and artwork by A&D
All songs and Toasted Heretic 2005 Copyright Control
A Bananafish recording licensed to our friends at Big Yes Music
Copy this for your poorer friends, and make the rich ones buy it.
Catalogue Number BYM002
www.toastedheretic.com
5.06008E+12

Figure 50: OCR Data from Test 13
LOOP GAROO mm: 4A2) <;"  
WHAT GOES AROUND COMES AROUND  
(Time: 2:56J)  
WASH, MAMA, WASH (Time: 3:35)  
CHIPPY, CHIPPY (Time: 3:30)  
MARD* GRAS DAY iTwne: 808)  
ANGOLA ANTHEM (Time 17:331  
All the selections were written by Mac Rebannack and are published by Eltoi  
ad -Cauldron ASCAP.  
Photography: Steve LaVere  
Alburn design: Stanley Moss/Art by the Moss  
ARRANGEMENTS: MAC REBEMKACK  
PRODUCED BY TOM OOWDi OR. JOHN $ CHARLES GREECE  
This is a stereo recording For best results ob-{l) serve the R.I.A.A.  
high frequency roll-off charac: ^*r* tenstic with a 500 cycle crossover.  
? 1970 Atlantic-Recording Corporation Printed in Japan  
ATCO RECORDS  
1841 BROADWAY, NEW YORK, N.Y, 10023  
DIVISION OF ATLANTIC RECORDING CORPORATION
SOMEONE'S LOOKING AT YOU
  DIAMOND SMILES
  WIND CHILL FACTOR (MINUS ZERO)
  HAVING MY PICTURE TAKEN
SLEEP (FINGERS’ LULLABY)
I DON'T LIKE MONDAYS
NOTHING HAPPENED TODAY
KEEP IT UP
NICE N NEAT
WHEN THE NIGHT COMES

BONUS TRACKS
  EPISODE #3
  REAL DIFFERENT - B-SIDE
  HOW DO YOU DO ?-B-SIDE
  LATE LAST NIGHT - B-SIDE
  NOTHING HAPPENED TODAY - LIVE IN CARDIFF 982 677-5

Figure 52: OCR Data from Test 15
a big
love my wish game true
how and then there
a foot such as
need your love tonight
don
i beg of you
santa bring my baby back
santa
is back in town party paralyzed one night i got stung king creole wear
time mean woman blues
playing for keeps
hard headed woman
gotaloto
todo
ll
peace in the valley
atoiaa
i
ni
usa

Figure 53: System Output from Test 1
mindsight
try again
cigarettes a ashtray
th
come on in
make
wanna
your
i me again
the man upstairs
stop the world
let me
me in care of th blues
no wheels on this ship
hungry for love
wali

Figure 54: System Output from Test 2
fragile a thousand years perfect love gone wrong all this time the hounds of stand so close to me when we dance dlenda roxanne i you love
set them free brand new day fields of gold moon over bourbon street shape ever lose my faith in you
every breath you jake
records ammrecords is a division of inc
a
records is a division rhcordings inc thl cop
in this sound
a im records and uchnsed exclusively i ydor ll
i

Figure 55: System Output from Test 3
lu maggie farm
one too many mornings
stuck inside of mobile with the memphis blues again
sister
lay lady lay
shelter from the storm
you're a big girl now
i threw it all away go idiot wind columbia
sony music entertainment inc cotumba is
tratonafk of sony muse
int oistrithjiion sony music

Figure 56: System Output from Test 4
dueling banjos
little maggie
shuckinthe corn
pony express
old
clark
eight more miles to louisville
farewell blues
earl
breakdown
end of a dream
gals
reuben
train
riding the waves
fire on the mountain
eighth of january
bugle call rag
hard aint it hard
mountain dew
rawhide
traditional arranged
eric weissberg tperformed
eric weissberg and steve mandell eric weissberg and marshall brickman performed
all other selections
new dimensions in banjo and bluegrass
france we
the music on this compact digital disc was originally recorded onanalog
we have attempted to preserve as closely as possible the sound of the original
warner bros records inc for the
wea international inc for the world outside the
warner bros records inc for the
sc wea international inc for the world outside the
all rights reserved
red i
er
presse
econl service
a
a warner com
ions com
bis i hoiked cop
lending public performance nd
record prohibited

Figure 57: System Output from Test 5
france we
i in i
the
in the
graceland
i know what i know
gumboots
diamonds on the soles of her shoes
you can call me al
under african skies
homeless
love vol
that was your mother
all around the world or the myth of fingerprints
paul simon engineer roy halee the music on this compact digital disc was
originally we have attempted to preserve as closely as possible the sound
of the original warner bros records inc for the
wea international inc for the world outside of the us
paul simon all rights reserved unauthorized duplication is a violation
of applicable manufactured in germany
record service
presse en allemagne par record service
a warner communications company

Figure 58: System Output from Test 6
who my group consists of
poo
doo of destine tambourine and
ditmus of con ga
boudreaux of
skins and
battiste of scorpio in bass clef
mclean of mandolin comp school
mann of bottleneck learning
of the immortal flute fleet the baron of ronyards
china goncy
leary shir ley marie laveaux
durden govenor plas johnson senator
west bowing croak er jean
sister stephanie and st theresa
cecilia la favorite karl a le jean who were all dredged
from the rigolets
the
of the second line under the eight visions of professor longhair reincanted the
my charts the rites or coco robicheaux who invisible to all
me will act as a second guardian angel until you over work him all
to us from the antilles to the bayou st
and aunt francis who told me the epic of
sturdy and apricot glow mimi who in silence says the
to mamma roux in chipacka the chopatoulis choctaws without teepees on magnolia st
parade and the golden blade the sun
to sun down second liners who dig fat tuesday more than
and thats plenty i have also
the old danse kalinda to remind you we have not chopped out the old
delphia we did the
a la gris gris calimbo to frame our thing into the medium of down
of deaux deaux the rattlesnake whose forked tongue hisses pig latin in silk and
ig day may the gilded splinters of auntie andre
forth in your path to light and guide your way through the bayous of
on your pirougue of heartaches and good times push and the shove that you
gris gris
eoau time
oanse kalinda ba doom
haroid battiste
me
mama roux
hill time
danse fambeaux
or
creaux time
croker courtbullion
harold battibte time
jump sturdy

or
creaux time
i walk on guilded splinters
creaux time
ail trie songs in this
are
masque jonarv
cover backliner photos raphael
design marvin
arranged produced
harrold battiste

Figure 59: System Output from Test 7
blow wind blow
big chief
somebody changed the lock
mess around
let the good times roll
partner
stack a
those lonely lonely nights
smith
a high blood pressure
don
you just know it
well i ll be
brown
little
jane

Figure 60: System Output from Test 8
lu maggie
farm
one too many mornings
stuck inside of mobile with the memphis blues again
sister
lay lady lay
shelter from the storm
you’re a big girl now
i threw it all away go idiot wind columbia
sony music entertainment inc cotumba is
tratonafk of sony muse
int oistrithjiion sony music

Figure 61: System Output from Test 9
this record must be played loud
i so good that i don have to brag
shel silverstein sings his songs
off the record
modern talk
i once
a woman
ever lovin machine
lookin for myself
the ugliest man in town
i so good that i don have to brag
the mermaid
i can
touch the sun
plastic
better not ask me
lemmebesomethin
testing the bomb recorded at mother blues chicago on october
personnel shel silverstein guitar vocals malcolm hale guitar
robert matthews bass charles watson
robert slawson harmonica little walter harmonica issued under
licence from universal music
universal music
demon music group
manufactured in england marketed
demon westsidej london
demon westside is a division of demon music group

Figure 62: System Output from Test 10
let it be naked
get back
dig a pony
you blue
the long and winding mad
two of us
i've got a feeling
one after
don't let me down
i me mine
across the universe
let it be additional disc
on the wall a
insight into the beatles a work in rehearsal in the studio during january
thebeatles.com
the
in this sound recording is owned
apple
emi records
this label
information is the
of
protection all rights reserved
apple
records
artwork
annie coras
printed in the
invitation
eighteen flavors
melinda mae
sick
ickle me pickle me tickle me too
enter this deserted house
jet and his tv set
for sale
warning
the
crocodile
toothache
stone telling
ridiculous rose
constrictor
peanut butter sandwich
listen to the
war
smart
forgotten language
the farmer and the queen
the one who stayed
no difference
wild boar
sarah cynthia
stout would not take the garbage out
my
oo
early bird
me stew
captain hook
with his mouth full of food
the flying festoon
the silver fish
the generals
vaat doid
the worst
my beard
previously unreleased bonus tracks
the little blue engine
i had a brontosaurus
one inch tall
long haired
rain
ture story
hungry mungry

standing
the world was
hector the collector
spaghetti
producer and director ron
sony music entertainment inc
sony music entertainment inc manufactured
columbia records
madison avenue new york
columbia legacy sony wonder and
reg
pat
marca registrada warning all rights reserved unauthorized duplication is a violation of applicable
sony
hel silverstein is the universally loved best selling author and illustrator of a light
and many other
of
and stories this
remastered edition of the grammy winning
based on the all time favorite where the sidewalk ends features
previously unreleased tracks complete
and new liner notes a delight for all ages it showcases the enduring legacy
magical writing and vocal artistry both in spoken word and song

Figure 64: System Output from Test 12
disc
songs for swinging celibates
sodom tonight
very naughty party
black contact lenses
goodbye to berlin
love theme prom yeats the movie
still life with girl
bouncing
the boulders
the best things in life are mine
disc
charm arrogance
drown the browns
lost and found
what it used to
stay tonight
you make girls unhappy
abandon the galleries
here comes the new year
there goes everything
go to sleep
charm arrogance
some
you can
go home
all songs written
collins farrell gough both
digitally remastered
neil farrell all photographs
aengus
even the ones he
where otherwise credited cover painting
gareth napoleon allen
layout and artwork
a
all songs and toasted heretic
control a bananafish recording licensed to our friends at big yes music
this for your poorer friends and make the rich ones
it catalogue number
toastedheretic com

Figure 65: System Output from Test 13
loop garoo mme
what goes around comes around
wash mama wash
mard gras day itwne
angola anthem
all the selections were written
mac rebannack and are published
eltoi ad cauldron ascap photography steve lavere alburn design stanley moss
art the moss arrangements mac
produced
tom
or
charles greece
this is a stereo recording for best results
serve the
i a a high
roll
charac
tenstic with a
crossover
atlantic recording corporation printed in japan atco records
new york
division of atlantic recording corporation

Figure 66: System Output from Test 14
someone
looking at you
diamond smiles
wind chill factor
having my picture taken
sleep
i don
like mondays
nothing happened today
keep it up
nice
neat
when the night comes
bonus tracks
episode
real different
side
how do you do
side
late last night
side
nothing happened today live in cardiff
biem sabam
mercury records
mercury records
the
in this sound recording is owned
mercury records
a universal music company all rights reserved unauthorised copying niring
lending public performance and

Figure 67: System Output from Test 15
CDs Scanned from Mobile Phone

See images on accompanying CD.
iQrections
xPlyiny Saucer Tour
3. I Please Do Not Disturd
4tGay m he Military
^Smoking
6tGreat Tines On Drugs
7. sex On Trial
iWhat ic/oraography?
*B*vm Willie
i0. Confession Time(COPS
) 11. Step on the g (L.A. Riots)
U.Folidcs In America
uT e Elite aTime To Evolve itOdd Beliefs
In The Bible
>>Easter iftGideon
a. Your Children
a The Sanctity Of Life

Figure 68: OCR Data from Test 1
UCant I
Lait Pofct Or t< I^glB
J. Don Be Sty
id Be I>>ns
eic her. he W
? The Ha
9. Arbeit liacht Frel
3. Campaign Ox Bate
,,MX Katie Did
U. Tom>>lan*
< The Saga
I5. Pead To P . . n
H. What Becaae Di fn

Figure 69: OCR Data from Test 2
i co
together where than the sun
i
myself bund burning wheel kowalski
long
swastika eves kill ail hippies accelerator shoot
kill
miss lucifer deep htt of
sun
some velvet morning

Figure 70: OCR Data from Test 3
INNUENDO  ’  RADIO GAGA  I WANT TO BREAK FREE UNDER PRESSURE  
’  A KIND OF MAGIC  
OGRE RATTLE SEVEN SEAS OF RHYE  DONT STOP ME  
""'*'.?*. . . . . . .  
-fli’.jSii’.V-  
..^.Vfe  
:TWAiiiiiiSri-.---............  
ONE VISION  WHO WANTS TO .^T  
....  ANOTHER ONE BITES THE DUST THESE ARE THE DAYS OF OUR LIVES  
WE WILL ROCK YOU  
tr*  
ili^H1illlAY:THE GAME  GUT YOU [Only The Good Die Young]  
Fitslfili:  
HAMMER TO FALL  
WE ARE THE CHAMPIONS  WE WIL1 ROCK1Y0liiitpiiion ]  
BOHEMIAN RHAPSODY  

Figure 71: OCR Data from Test 4
510 160-2
i
calling via

d
U 7*.
a*
*
&
SVA

"M

on every street
S."
i 'l
LC) 16

...*r

**
*<*
.~-*
*4*
"v
when it comes to yCi*
#:U7,>
/Jr

SJV.
.,'r-
V->*

-:*-X^.
^-.
rt
>
*&
--"i
System Output: CD Images from Mobile Phone

saucer tour
i please do not disturdisturd
he military
smoking
times on
sex on trial
ic
willie
confession
step on the
a
folidics in america ut
elite atime to evolve itodd beliefs in the bible easter
a your children
the sanctity of life

Figure 73: System Output from Test 1
ucan
i
laiit
or
i
don be sty
id be i ns
eic her he
the ha
arbeit liacht frel
campaign
bate
katie did
tom lan
the saga
pead to
what

Figure 74: System Output from Test 2
Figure 75: System Output from Test 3
innuendo
radio gaga i want to break free under pressure
a kind of magic
dont stop me
fli
one vision who wants to
another one bites the dust these are the days of our lives we will
you only the good die young fitslfiili hammer to fall we are the champions
rocklyoliiipiion
bohemian rhapsody

Figure 76: System Output from Test 4
Figure 77: System Output from Test 5
I Freedb Sample Tracks

0. Nothing Is True - Mae Pang
1. It’s too Late - Psyclone Rangers
2. Three Sisters - Thorazine
3. People Who Died - Bootleg Thorazine Nightmare
4. Catholic Boy - Bottom
5. City Drops into the Night - Stomaboy Marrow
6. I Want the Angel - Jen Hess
7. Day and Night - Iota
8. Differing Touch - Marah
9. Crow - Del Pess
10. Plain Division - Mia Johnson
11. Wicked Gravity - Sensitive Pricks

Figure 78: Test 1
0. Sinead O’Connor / Nothing Compares 2 U
1. Lou Reed / Perfect Day
2. The Scene / Het werk van God
3. Eels / Beatiful Freak
4. John Hiatt / Have A Little Faith In Me
5. Morphine / Potion
6. U2 / Love Rescue Me
7. Nick Cave / Into My Arms
8. Noordkaap / Ik Hou Van U
9. Joan Osborne / What If God Was One Of Us
10. Flowers For Breakfast / One Man Show
11. dEUS / Lettle Arithmetics
12. Moondog Jr / Jintro & The Great Luna
13. The Scene / Horizon
14. Radiohead / Fade Out
15. Nirvana / Where Did You Sleep Last Night

Figure 79: Test 2
0. Gsieg Gott
1. Der Eispickel
2. Die Haustechnik
3. Sponsorentum
4. Der Herr Wildmoser
5. Die Gehirnoperation
6. Die Deutsche Fernsehlanschaft
7. Vera am Mittag
8. Alfred Biolek
9. DSF und Onkel Erwin
10. Nackte Weiber
11. Das Sex-Symbol
12. Die Pause

Figure 80: Test 3
0. Hit The Road Jack
1. Speedy Gonzales
2. The Great Pretender
3. Under The Boardwalk
4. Ginny Come Lately
5. I’m Gonna Get Married
6. Rhythm Of The Rain
7. What Kind Of Fool Do You Think I Am
8. When
9. Mr Bassman
10. Sheila
11. The Ball Weevil Song
12. Heartbeat
13. Dream Lover
14. These Arms Of Mine
15. What A Wonderful World

Figure 81: Test 4
0. Outside of a Small Circle of Friends
1. Christine Keler
2. Cops of the World
3. Draft Dodger Rag
4. I ain’t Marching Anymore
5. In the Heat of the Summer
6. Joe Hill (Tape from California, 1968)
7. No Christmas in Kentucky
8. Outs. of a Small Circle of Friends (live)
9. The Power and the Glory
10. Spanish War Song
11. Freedom Riders
12. There But for Fortune
13. Too many Martyrs
14. When I’m Gone
15. I shoulda known Better
16. Changes
17. Hills of West Virginia

Figure 82: Test 5
0. Nuit De Folie
1. La Vie, La Nuit
2. Crocodiles Cauchemar
3. Cours D’anglais
4. Belles, Belles, Belles
5. Come On
6. Forêt D’amazonie
7. Si L’on Danse
8. Envie De Chanter
9. Entre Soleil Et Pluie
10. WeekEnd Dance
11. Début De Soir’Rock
12. Refrain D’enfer
13. Chance
14. Plus Jamais
15. De Revolution En Satisfaction
16. Fille Du Cha-Cha
17. Les Anges Du Futur
18. L’espoir Est Un Geant
19. Jardins D’enfants

Figure 83: Test 6
0. Riverdance - The celtic Orchestra
1. The Wild Rover - Patsy Watchorn
2. Song For Ireland - Barleycorn
3. The fields of Athenry - Paddy Reilly
4. The big Strong man - Patsy Watchorn
5. The rose of tralee - Sean Dunphy
6. Whiskey in the Jar - Patsy Watchorn
7. The Hag with the money - Brendan Mulhaire Cili Band
8. The Irish Rover - Patsy Watchorn
9. Boolavogue - Brendan Bowyer
10. The Craic was ninety in the ilse of man - Patsy Watchorn
11. Danny Boy - Sean Dunphy
12. Barndances - Arcady
13. Staten Island - Johnny McEvoy
14. A nation Once again - Paddy Reilly
15. Molly Malone - Dublin city ramblers
16. Lord of the dance - The celtic Orchestra
17. The Rare old times - Jim McCann
18. Seven Drunken nights - Patsy Watchorn
19. The molly maguire's - The Jolly Beggarmen
20. Pub Song Medley - Patsy Watchorn
21. The Irish National Anthem - The Band of an Garda Siochana
0. Roll Up And Shine
1. The Bartender And The Thief
2. Hurry Up And Wait
3. Pick A Part That’s New
4. Just Looking
5. Half The Lies You Tell Ain’t True
6. I Wouldn’t Believe Your Radio
7. T-Shirt Sun Tan
8. Is Yesterday, Tomorrow, Today?
9. A Minute Longer
10. She Takes Her Clothes Off
11. Plastic California
12. I Stopped To Fill My Car Up

Figure 85: Test 8
0. Murders In The Rue Morgue
1. Wrathchild
2. Run To The Hills
3. Children Of The Damned
4. The Number Of The Beast
5. Another Life
6. Killers
7. Acacia Avenue
8. Total Eclipse

Figure 86: Test 9
<table>
<thead>
<tr>
<th>Track</th>
<th>Artist / Album Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>Stardust / Music sounds better with you</td>
</tr>
<tr>
<td>1.</td>
<td>Sophie Ellis Baxtor / Murder on the dancefloor</td>
</tr>
<tr>
<td>2.</td>
<td>Phoenix / If I ever feel better</td>
</tr>
<tr>
<td>3.</td>
<td>Tonka / Never</td>
</tr>
<tr>
<td>4.</td>
<td>DB Boulevard / Point of view</td>
</tr>
<tr>
<td>5.</td>
<td>Jamiroquai / Love foolosophy</td>
</tr>
<tr>
<td>6.</td>
<td>Phats and Small / Feel good</td>
</tr>
<tr>
<td>7.</td>
<td>Alcazar / Sexual guarantee</td>
</tr>
<tr>
<td>8.</td>
<td>Galleon / So, I begin</td>
</tr>
<tr>
<td>9.</td>
<td>Benjamin Diamond / In your arms</td>
</tr>
<tr>
<td>10.</td>
<td>Stuart / Now I’m free</td>
</tr>
<tr>
<td>11.</td>
<td>Everything but the girl feat. Soulvision / Tracey in my room</td>
</tr>
<tr>
<td>12.</td>
<td>Milk &amp; Sugar / Lift me up</td>
</tr>
<tr>
<td>13.</td>
<td>Francesco Diaz / Say a little prayer</td>
</tr>
<tr>
<td>15.</td>
<td>Roger Sanchez feat. N’Dea Davenport / You can’t catch me</td>
</tr>
<tr>
<td>16.</td>
<td>TDN / Shame</td>
</tr>
<tr>
<td>17.</td>
<td>Jon Cutler feat. E-Man / It’s yours</td>
</tr>
<tr>
<td>18.</td>
<td>Live Element / Be free</td>
</tr>
<tr>
<td>19.</td>
<td>Various / The finest in house vol 4 cd 01</td>
</tr>
</tbody>
</table>

Figure 87: Test 10
nothing is true
pang
it
too late
rangers
three sisters thorazine
people who died bootleg thorazine nightmare
catholic
bottom
city drops into the night stomaboy marrow
i want the angel jen hess
day and night iota
differing touch marah
crow del pess
plain division mia johnson
wicked gravity sensitive pricks

Figure 88: Test 1
sinead
connor nothing compares
lou reed perfect day
the scene het werk van god
eels beatiful freak
hiatt have a little faith in me
morphine potion
love rescue me
nick cave into my arms
hou van
joan osborne what if god was one of us
flowers for breakfast one man show
deus lettle arithmetics
moondog
jintro the great luna
the scene horizon
radiohead fade out
nirvana where did you sleep last night

Figure 89: Test 2
sie gott
der eispickel
die haustechnik
-sponsored	

der herr wildmoser
die gehirnoperation
die deutsche fernsehlanschaft
vera am mittag
alfred biolek
und onkel erwin
nackte weiber
das sex
die pause

Figure 90: Test 3
hit the road jack
speedy gonzales
the great pretender
under the boardwalk
ginny come lately
i
gonna get married
rhythm of the rain
what kind of fool do you think i am
when
bassman
sheila
the ball weevil song
heartbeat
dream lover
these arms of mine
what a wonderful world

Figure 91: Test 4
outside of a small circle of friends
christine keler
cops of the world
draft dodger rag
i ain
mariching anymore
in the heat of the summer
hill
from california
no christmas in kentucky
outs of a small circle of friends
the power and the glory
spanish war song
freedom riders
there
for fortune
too many martyrs
when i
gone
i shoulda
better
changes
hills of west virginia

Figure 92: Test 5
nuit de folie
la vie la nuit
crocodiles cauchemar
cours
anglais
belles belles belles
come on
foret
amazonie
si
on danse
envie de chanter
entre soleil et pluie
weekend dance
debut de soir rock
refrain
enfer
chance
plus jamais
de revolution en satisfaction
fille
cha cha
les anges du futur
espoir est un geant
jardins
enfants

Figure 93: Test 6
riverdance the celtic orchestra
the wild rover patsy watchorn
song for ireland barleycorn
the fields of athenry
reilly
the big strong man patsy watchorn
the rose of tralee sean
whiskey in the jar patsy watchorn
the hag with the money brendan mulhaire band
the irish rover patsy watchorn
boolavogue brendan
do the craic was ninety in the ilse of man patsy watchorn
scruffy
sean
barndances arcady
staten island
a nation once again
reilly
molly malone dublin city ramblers
lord of the dance the celtic orchestra
the rare old times
seven drunken nights patsy watchorn
the molly maguire
the jolly beggarmen
song medley patsy watchorn
the irish national anthem the band of an garda siochana

Figure 94: Test 7
roll up and shine
the bartender and the thief
hurry up and wait
pick a part that
new
just looking
half the lies you tell ain
true
i
believe your radio
shirt sun tan
is yesterday tomorrow today
a minute longer
she takes her clothes
plastic california
i stopped to fill my car up

Figure 95: Test 8
murders in the rue morgue
wrathchild
run to the hills
children of the damned
the number of the beast
another life
killers
acacia avenue
total eclipse

Figure 96: Test 9
stardust music sounds better with you
sophie ellis
murder on the dancefloor
phoenix
i ever feel better
tonka never
boulevard point of view
love foolosophy
phats and small feel good
alcazar sexual guarantee
galleon so i begin
benjamin diamond in your arms
stuart now i
free
everything but the girl feat soulvision tracey in my room
milk sugar
me
francesco diaz say a little prayer
horny united pres lovesick feat mossee let
stay together
roger sanchez feat
dea davenport you can
catch me
shame
jon cutler feat
man it
yours
live element be free
various the finest in house vol

Figure 97: Test 10
K  Tools Utilised

Setting up MySQL on macneill
Email the Computer Science Department Helpdesk, help@cs.tcd.ie, to set up a MySQL account on macneill. They will provide you with a user name (your college login), password and database account. SSH into macneill.cs.tcd.ie, using your college login and college password. To access MySQL, type mysql -p nameOfDatabaseAccount. You will then be prompted for a password.