Character recognition and information retrieval

Julia Ann Cooley Borsack
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds

Repository Citation
https://digitalscholarship.unlv.edu/rtds/300

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6” x 9” black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road. Ann Arbor, MI 48106-1346 USA
313/761-4700  800/521-0600
Character recognition and information retrieval

Borsack, Julia Ann Cooley, M.S.
University of Nevada, Las Vegas, 1993
Character Recognition and Information Retrieval

by

Julia Ann Cooley Borsack

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

Department of Computer Science
University of Nevada, Las Vegas
August 1993
The thesis of Julia Ann Cooley Borsack for the degree of Master of Science in Computer Science is approved.

Kazem Taghva, Ph.D
Chairperson

Tom Nartker, Ph.D
Examining Committee Member

Laxmi P. Gewali, Ph.D
Examining Committee Member

Ashok K. Singh, Ph.D
Graduate Faculty Representative

Ronald W. Smith, Ph.D
Graduate Dean

University of Nevada, Las Vegas
June, 1993
Presented are two technologies, character recognition and information retrieval, that are used for text processing. Character recognition translates text image data to a computer-coded format; information retrieval stores these data and provides efficient access to the text. The necessity of their eventual coupling is obvious. Their sequential application though (with no manual intervention) has been considered impractical at best. Our experimentation exploits these two technologies in just this way. We identify problems with their combined use, as well as show that the technologies have come to a point where they can be applied in succession.
Contents

Abstract ................................................................. iii
List of Figures ........................................................... v
List of Tables ........................................................... vi
Acknowledgement ...................................................... vii

Chapter 1 Introduction .............................................. 1

Chapter 2 Document Recognition ............................. 3
2.1 Page Scanning ..................................................... 3
   2.1.1 Thresholding ............................................... 3
   2.1.2 Resolution .................................................. 4
2.2 Recognition Devices ........................................... 4
   2.2.1 Feature Extraction Techniques ....................... 7

Chapter 3 Information Systems ............................... 12
3.1 Information System Models ................................. 13
   3.1.1 Boolean Model ........................................... 13
   3.1.2 Vector Space Model ..................................... 16
   3.1.3 Probabilistic Model ..................................... 18
   3.1.4 Cluster-based Model .................................... 19
   3.1.5 IR Extensions ............................................. 20
3.2 Measure of Relevance ........................................... 23

Chapter 4 Combining OCR and IR ........................... 27
4.1 Overview of Experimental Purpose ...................... 27
4.2 Experimental Environment ................................. 28
   4.2.1 Scan and OCR environment ......................... 29
   4.2.2 Text Retrieval environment ......................... 30
4.3 Method of Evaluation ......................................... 32
4.4 Results: experiment 1 .......................................... 34
4.5 Results: experiment 2 .......................................... 35

Chapter 5 Conclusion ................................................ 37
Bibliography .......................................................... 38
List of Figures

2.1 Poorly scanned image ......................................................... 6
2.2 Template for the number 2 .................................................. 8
2.3 Crossing technique ............................................................ 9

3.1 Theoretical Information System .......................................... 12
3.2 Expanded Information System ............................................. 14
3.3 Vector representation of documents and query ...................... 17
3.4 Cluster-based organization ............................................... 19
3.5 Recall and precision graph for ranked output ...................... 25

4.1 Example test query translation ............................................ 32
# List of Tables

3.1 Example thesaurus entry .................................................. 21
3.2 Relevance contingency table ............................................. 24
3.3 Sample ranked documents ................................................. 25
3.4 Recall and precision values for sample documents ............ 25
3.5 Other measures of relevance ............................................. 26

4.1 Experiment 1 query results ................................................ 34
4.2 Experiment 1 source of errors .......................................... 35
4.3 Experiment 2 query results .............................................. 36
4.4 Experiment 2 source of errors .......................................... 36
Acknowledgements

For an undertaking of any size, no single person deserves the credit for its completion. This is true of my thesis. Without the support of my parents, my family, my colleagues and my friends, my work would not have been completed. I would like to give special thanks to my husband and my daughter, Kristín, for being there when I needed them.

I would like to note that the project itself was a group effort. A number of people had a hand in seeing it to its completion. I would like to thank Erva Srinivasulu, Bryan Bullard, and Allen Condit for their help in the experimentation.

I would also like to acknowledge Matt Kinkenon and again, Allen Condit, for their friendship and moral support.

Lastly, I would like to thank my advisor, Dr. Kazem Taghva, for his insight and guidance in this area of research.
Chapter 1
Introduction

Although optical character recognition (OCR) and information retrieval (IR) both manipulate text, their initial objectives were very different. In fact, these objectives began at opposite ends of the linguistic spectrum.

Optical character recognition devices, as their name implies, were designed to recognize characters and convert them to another format. One of the first examples of a machine that could read printed characters was the *Optophone* developed by Fournier D’Albe to aid the blind[18]. This device scanned the characters on a printed page and produced audible tones for each character. A number of other recognition devices were invented for different purposes but were based on the same character recognition principles.

The applications for OCR branched into a number of related domains. But here we focus on OCR devices that optically recognize scanned printed text. These devices have shown continued advancements in their effectiveness of the recognition process. Some of this improvement is due to the sophistication of related technologies[10]. But part of this improvement should be attributed to the more global view that is taken when a page is recognized. OCR devices no longer view a page as a set of unrelated, isolated, character patterns. In fact, it has been suggested that optical character recognition is a misnomer; the designation, *text recognition*, would better identify the
technology. This increased awareness of continuity of a text page not only discloses more information about the text being recognized, it also brings this technology a step closer to Information Retrieval which perceives a document as a single related unit of information.

The need to store and retrieve information predated computer technology. But the demand for information access escalated when rapid printing capabilities became available[22]. The earliest computer-based systems controlled a set of index terms that were chosen to represent the document's content. The documents themselves were stored on microfilm[15]. From its beginning, IR technology's purpose was to typify document content and be able to present this information upon request.

Unfortunately, semantically representing natural language is problematic, especially for a diverse document collection. Therefore, information retrieval systems, used for general application have resorted to simpler methods of text description. One of the more useful tools applied in IR is statistical analysis. This kind of decomposition of text breaks the document into smaller segments; sections, paragraphs, sentences, words, even character combinations can be used to delineate document content. Examination of document constituents has contributed to our understanding of language structure and the qualities of printed text. Further, the association between IR and OCR becomes clearer. From a progressive perspective, IR can be considered an extension of OCR, where the combined systems begin with an indivisible particle, the pixel, and produce an accessible collection of information.

It is the combined use of these two technologies that is under investigation in this paper. Presented are the methods applied in these technological domains together with experiments that report on their integration.
Chapter 2

Document Recognition

There are two major processing steps necessary to convert a paper document into its computer useable form:

1. page scanning: the conversion of the input page to its bit-mapped image.

2. page recognition: the translation of the bit-mapped image to its computer-coded format.

2.1 Page Scanning

Optical scanners sense variations in light intensity to determine patterns on the input page. These analog signals are digitized to represent the pattern viewed. Since most printed documents consist of black patterns on a white background, the digitized page can be represented by a matrix of 1’s (for black) and 0’s (for white). The digitized matrix is called a bit-mapped image. There are two important scanning parameters that influence the resulting bit-map: thresholding and resolution.

2.1.1 Thresholding

The light intensity generated from the sensing device is an analog signal and therefore, is not discrete. Thresholding refers to the value used to determine whether a particular point in the pattern should be classified as black or white.
The threshold can be a pre-established value or it can be adaptive. The thresholding method chosen can affect the accuracy of subsequent recognition. Fixed thresholding is satisfactory only when the image to be scanned has high contrast against its background\[2\]. Adaptive thresholding is less stringent in that the threshold value is determined using feature information from the image being scanned. Feature information can be analyzed at either a global (full-page image) or local (pixel) level.

2.1.2 Resolution

Resolution refers to the reduction of an image into its smallest, distinguishable components, more commonly known as dots per inch (dpi). The resolution settings should be dictated by the image to be scanned and the subsequent recognition method applied. Graphics recognition permits a lower resolution (about 80 dpi)\[2\], while text may require resolution as high as 400 dpi if the character size is small\[10\]. Interestingly enough, as pointed out in \[10\], “too high a resolution may actually degrade the performance [of the recognition phase] by digitizing unwanted noise.”

The recognition method should also be a factor in determining resolution. A simple template recognition approach (discussed later) will require a higher resolution than a more sophisticated feature recognition technique that is less sensitive to distortions\[2\].

There is a close association between optical scanning and the method of character recognition applied. Early recognition algorithms were tailored not only to its input but also to the optical sensing methods used\[24\].

2.2 Recognition Devices

Although the steps involved in optically recognizing a printed page goes beyond simple character recognition, the name, optical character recognition, endures. Interestingly enough, given a string of isolated characters, most commercial OCR devices fail
miserably[7]. The OCR device analyzes the page, the line, the word, and the character, indicating more to its algorithm than identifying a single unrelated pattern.

Presented with an image to be recognized, there are a number of steps performed prior to the actual character classification phase:

1. skew correction
2. zone segmentation
3. zone classification
4. line finding
5. character isolation
6. character normalization, smoothing and noise reduction
7. feature extraction
8. character classification

These functions may not be ordered as specified; some processes may be intermingled with others. The point is, text recognition is a complex process.

To cover each of these processes in detail is beyond the scope of this paper. So we will give a brief definition of each function and explain the most prominent function of OCR, feature extraction, in more detail.

**Skew correction** is the rotation of a printed image to its intended orientation.

*Deskewing* can either be accomplished at the page level or at the block level[10]. A commonly used algorithm for skew correction is Hough Transform[1]. Figure 2.1 is an example of a badly skewed page.
Zone segmentation and classification are used to decompose a page image into its logical parts (segmentation) and to differentiate text from non-text\(^1\) (classification). Logically these functions are distinct. But segmentation and classification are usually synthesized processes. Information used to segment a page into zones can also be applied to their classification.

Line finding refers to the location of individual lines of text. This process can be handled together with skew correction. Its purpose is to distinguish closely-spaced, adjacent lines.

Character isolation pertains to the separation of text into individual characters. Character isolation techniques, although easy to implement, can be impaired by touching and broken characters, proportional fonts and narrow vertical line

---

\(^1\)This statement is a generalization, since devices may classify non-text more precisely.
spacing. Nagy states, “Imperfect separation between adjacent lines accounts for a large number of misclassifications”[10].

Character normalization, smoothing and noise reduction render the page more “readable” to the OCR device. Normalization corrects character slant and reduces various fonts to a uniform standard. Smoothing fills unintended breaks and holes while thinning the character to produce a more distinguishable pattern. Noise reduction removes incidental variations in the image not part of the original document.

Feature extraction and character classification are generally thought of singularly as character recognition. Of the complete OCR process, feature extraction is the true paradigm of pattern recognition. A number of techniques are employed to extract shape features. Some are described in the section following. Character classification determines, from the set of features extracted, to which class the character belongs. The classification is usually based on similarities between the unknown input pattern and a priori information about character shape.

2.2.1 Feature Extraction Techniques

The following are general descriptions of the more common feature extraction algorithms. Actual implementations may fall into one or more of these descriptions.

Template Matching

Template matching, also referred to as prototype correlation, was the technique applied in the first working optical character reader[18]. In this technique, the unknown character is compared to a set of prototype patterns stored in the device. An example template is shown in Figure 2.2[6].
The distance between the unknown pattern and each template is calculated; the pattern is classified or rejected based on some closeness measure and threshold value.

In some sense, this technique is very restrictive. Any variation in the character, caused by differing fonts or noise will seriously affect recognition. Used in isolation, high accuracy rates should not be expected. But complementary techniques, such as character normalization, smoothing and noise reduction should improve performance[6]. Further, template matching has been implemented to contrive templates while a page is being processed[10]. In this way, templates can adapt to changing fonts.

**Point Distribution**

A number of recognition algorithms rely on the statistical distribution of points. Two algorithms in particular use the features derived from moments and crossings to determine shape[6].

Using central moments, the perpendicular distance of the points are calculated from a character’s center of reference or *centroid*. The moments are used as distinguishing features for the character pattern.
The crossing technique discerns character shape by tracking the intersections of lines with the character pattern. An example of this technique is illustrated in Figure 2.3[6].

This technique is commonly used in commercial OCR devices because it can be performed at high speed and is tolerant to moderate distortion[6].

Structural Analysis

Structural analysis typically relies on the geometrical properties of the pattern. The character is decomposed into a set of shapes, together with their placement within the character's frame. The most commonly exploited feature shapes are strokes and bays[6]. Other geometrical properties include line segments, loops and intersections. There are a number of shapes that can be extracted from characters. But as pointed out in [10], "good" features are those that occur commonly together in a single character class. Structural analysis has a high tolerance to image distortions but has not been fully exploited as a tool for optical character recognition.
Omnifont Analysis

Although omnifont analysis encompasses more than just feature extraction, a brief explanation is included here to point out that character classification for real-world documents may require more information that just single character attributes. As stated in [8] when discussing the construction of a prototype omnifont system, "we found it necessary to mix methodologies, to use context to different degrees at different stages, and to complement general algorithms with a few handcrafted rules for special cases."

The omnifont ideal should have the ability to recognize characters in any font and in any size. With this in mind, it seems obvious that the above algorithms are not discerning enough in isolation to correctly recognize different typefaces. The idea of omnifont analysis then is to use as much additional information as necessary prior to character classification.

The methods of analysis mentioned above assume that a single segmented character has been presented for recognition. Many times this is not the case. For example, some fonts may be tightly kerned or the serifs may be touching causing adjacent characters to join. Split characters can also be a problem because parts of a single character may be segmented separately. These kinds of problems are more readily addressed with omnifont analysis.

With omnifont, if a segmented character is not clearly recognizable, then a set of candidates are output, together with their confidence values. Further, if the algorithm believes the input shape may be a join or a split, the shape is resegmented prior to classification[8][1].

Omnifont analysis also takes advantage of previously recognized characters on the page. For example, if an “e” has already been recognized with certainty, then its features can be used to recognize other “e’s” on the same page[1]. Other contextual
methods\textsuperscript{2} such as document structure and linguistic rules are used to recognize characters. Examples of these contextual methods include line information, classification of adjacent text, n-gram analysis, spell checkers and heuristics[8][1].

One point to keep in mind is that the ideal omnifont classifier has not yet been attained. As pointed out in [1], “The best products [OCR devices], do a good job on clean documents, but they all degrade in performance as document quality (or scanner quality) degrades.”

\textsuperscript{2}these methods could also be classified as part of the postprocessing phase
Chapter 3

Information Systems

Automated document storage and retrieval was not originally a computer-based system. The first automated systems, developed in the 1950's, were designed to retrieve microfilm images using digitally-coded index information[15]. Although the implementations of these first systems were not computer-based, their purpose was the same as the information systems we have today: to provide a means of automatically locating information upon request.

Information can be presented in many forms but the concentration of this paper will be on information in the form of written text. Further we will assume that the database of information consists of discreet units called documents. With this in mind, a high-level information system can be illustrated as shown in Figure 3.1[17].

A set of information needs or requests is compared to a document collection to determine which documents satisfy the requests. The methodologies described in

Figure 3.1: Theoretical Information System
the next section model the three pieces of an information system: its requests, its documents and its similarity measure.

3.1 Information System Models

Figure 3.1 is only a theoretical depiction of an information system. A user request must be formalized and a parallel representation of the documents in the collection must be built prior to the comparison. So before the similarity measure can be applied, some resolution between the requests and the documents must be done. This resolution is defined through information retrieval models. There are four well known models that currently influence information retrieval: Boolean model, vector space model, probabilistic model and cluster-based model[4]. The most convenient way of perceiving the database is as a set of documents. But in practice, the most common structure for document storage is an inverted index. An inverted index transposes the document-term relationship to a term-document relationship. For each term in the collection, the documents in which that term occurs is assigned to that term. This implementation allows for immediate response to user requests. Although not explicitly stated in the description of the models below, this is the document database representation employed.

Unfortunately, anytime a structured representation is forced, characteristics of the original notion may be lost. The ability to truly represent a user's information needs and the meaning of a document's content is a difficult problem in information retrieval. A less idealistic representation of an IR system, shown in Figure 3.2[16], illustrates these changes.

3.1.1 Boolean Model

The Boolean model is named primarily for its method for formulating user requests. The Boolean model is a formal retrieval model since it has a clear theoretical basis[4],
Boolean algebra. In a pure Boolean model, each document is denoted as a binary vector representing a set of concepts (i.e. index terms) assigned to that document. The request or query is represented as a set of terms joined by the logical operators or, and, and not.

Let $A$ represent a request and $B$ represent a document vector, then:

or is the disjunction of $A$ and $B$ and is true if either $A$ or $B$ is true otherwise it returns false.

and is the conjunction of $A$ and $B$ and is true only if both $A$ and $B$ are true otherwise it returns false.

not is the negation of $A$ and is true whenever $A$ is false and false otherwise.

These operations use set union, intersection and difference respectively. Using the Boolean logic definitions described, the similarity measure becomes the evaluation of a Boolean query against the document collection. The documents retrieved represent the satisfiability of a propositional logic expression. If a document satisfies the expression, then a true value results and the document is considered relevant; a false value indicates non-relevance.
Two problems in particular are associated with the Boolean model:

1. The complexity of query formulation and interpretation.
2. The lack of ranked document output.

The syntactic structure of a Boolean query language is quite simplistic. With unambiguous parsing rules and a set of axioms, the evaluation of a query is clear. But the more simplistic the language, the more tedious it becomes for the user to express complex relationships. To further confuse the issue, the order in which operations are executed may change the query’s results. If the parsing rules are not fully understood by the user, it may not be clear why a certain set of documents was returned and its complement was excluded. For example, for the following inverted index and query,

<table>
<thead>
<tr>
<th>Terms</th>
<th>Document Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>orange</td>
<td>2 3 4 5 6</td>
</tr>
<tr>
<td>banana</td>
<td>4 6 8</td>
</tr>
<tr>
<td>grape</td>
<td>3 7 9 11</td>
</tr>
</tbody>
</table>

Query: APPLE AND ORANGE OR BANANA

if the parsing starts at the left and moves right, the documents retrieved will be: 3 4 5 6 8. If the parsing is done from right to left the results will be: 3 5[17]. Salton states, “In general, formulating Boolean queries is an art that is not accessible to uninitiated users”[16].

A Boolean query returns a result set by partitioning the document collection into two parts, the retrieved part and the non-retrieved part. Even if we assume that all the documents retrieved are relevant, it is still left to the user to determine which documents are most relevant. In a large document collection, this filtering may not
be feasible. Document ranking addresses this problem but is not easily incorporated into the Boolean model[16].

3.1.2 Vector Space Model

The vector space model is similar to the Boolean model in that theoretically, each document in the collection is represented as a term vector. The vector space model though extends this vector representation to its queries. The presence of term \( K \) in a document or query is represented by true (or 1) and its absence by false (or 0). An alternative approach would be to weight the value of term \( K \) in its vector based on its importance to the document or query. There are a number of similarity measures that can be used in the vector space model but the most common function is the cosine of the angle between two vectors. Document-query similarity can be computed as

\[
D_r \cdot Q_s = \sum_{i,j=1}^{t} a_{ri}q_{sj}T_i \cdot T_j
\]

(3.1)

For this formula to be used, term correlations, \( T_i \) and \( T_j \), must be known. Since these values are not easily generated in practice, document terms are assumed to be uncorrelated and the formula is reduced to the simple sum of products form:

\[
sim(D_r, Q_s) = \sum_{i,j=1}^{t} a_{ri}q_{sj}
\]

(3.2)

A pictorial representation of the cosine similarity measure can be seen in Figure 3.3[17]. Since the terms are uncorrelated, the term vectors are orthogonal and therefore linearly independent. Other similarity measures are described in detail in [16].

The most commonly employed weighting algorithm is the tf.idf weight which uses the frequency of a term in a single document (tf or term frequency) balanced by its
Doc1 = (term2, term5, term 7)

Query = (term2, term5, term6)

Doc2 = (term1, term3, term6)

Figure 3.3: Vector representation of documents and query

frequency in the entire collection (idf or inverse document frequency):

\[ w_{ij} = t_{f_{ij}} \cdot \log \frac{N}{d_{f_{ij}}} \] (3.3)

Other term weighting algorithms include the signal-noise ratio and the term discrimination value[17].

The vector space model resolves some of the problems of the Boolean model,

1. **The query is easier for the user to formulate since it consists of a set of relevant terms. No logical operators need be considered. Further,**

2. **Since ranking can easily be introduced into the retrieval system through term weighting, the user has more information about the probability of relevance in the retrieved set.**
Some of its disadvantages are its assumed term independence and its arbitrary selection of a weighting technique and similarity function to determine relevance[4].

### 3.1.3 Probabilistic Model

The probabilistic model rests on the premise that document to query relevance is a matter of degree[17]. If the probability of relevance is above some threshold, then the document is considered sufficiently similar and should be retrieved in response to the given query.

To maximize the possibility of retrieving a relevant document, cost parameters are introduced. Cost parameter $C_1$, associates a cost with retrieving a non-relevant document and cost parameter $C_2$, associates a cost with not retrieving a relevant document. The total cost is minimized by retrieving a document when

$$Prob(R) \cdot C_2 \geq [1 - Prob(R)] \cdot C_1$$  \hspace{1cm} (3.4)

However, before this equation can be satisfied, the probability of relevance, $Prob(R)$, for a document must be found. Document relevance relies on term relevance. To determine term relevance values, not only must the occurrence characteristics for each term be calculated, the correlation between terms must be considered. Individual term occurrences can be characterized by applying a probability distribution, such as Poisson. Another method generalizes distributions found in similar document collections to characterize term frequency[17].

Term correlation probabilities cannot feasibly be calculated for all term subsets in a document collection of any size. Therefore, the probabilistic model simplifies these calculations by considering only some of the more important pairwise term relationships. Reduced term dependency though may result in the possible exclusion of important term correlations. Further, if each term is considered independently, the probabilistic model becomes a form of vector space[16].
The most notable shortcoming of the probabilistic model is its difficulty in calculating representative values for term occurrences. Current research using the probabilistic model continues under the direction of Dr. Bruce Croft and others to attempt to rectify this deficiency[4].

3.1.4 Cluster-based Model

Cluster-based retrieval is different from the other models in that the similarity measure is first applied to the collection to form clusters of related documents. Figure 3.4[17] shows a possible document collection clustering. A single document vector, called the centroid is constructed from documents that were clustered together. Rather than individual documents, these centroids are compared to incoming queries. The clusters denoted by the most similar centroids to the query are retrieved. The cluster-based retrieval model rests on the Cluster Hypothesis which states: “closely associated documents tend to be relevant to the same requests”[23]. Implicitly, this model incorporates term association through document association.
A cluster-based retrieval system should embody two qualities,

**stability:** minor changes to the database should cause only minor alterations to the clusters.

**clear definition:** clusters should be well-defined; they should represent a single concept or a few compatible concepts[17].

A number of variables are introduced in this retrieval model. Not only must the similarity measure and document and query representations be selected, but the centroid clustering method and thresholds must be specified. Further, changes in these values alter the documents considered relevant by the system.

### 3.1.5 IR Extensions

The models discussed provide a framework for the three elements of an IR system. The differences between a structured record database and a free-text IR system should be obvious. But at some level, string matching must occur to locate the documents which match a query. This is an inherent difficulty in information retrieval. To illustrate, if one were interested in finding documents concerning childhood illnesses, several phrases could be used: child illness, children's afflictions, infant diseases, adolescent disorders, etc. There are no field specifications that restrict the IR user or the contents of the database. Even if the information is available, because of these variants, the user may not be able to locate it. Following is a description of some techniques applied in IR that attempt to overcome these kinds of problems and improve an IR system's effectiveness.

**Word stemming** is the removal of suffixes (and in some systems prefixes) to form root words. This normalization reduces many forms of the same word to a single common word stem. It is easy to see how document recall can be improved by
applying word stemming to a document collection. The same algorithm must be applied to query terms as well. There are several algorithms that exist to remove suffixes. One in particular introduced by Paice, locates the longest suffix for removal by consulting a list of word endings together with a set of associated rules[12]. When using these kinds of algorithms, for English or any other natural language, there are a number of exceptions that should be considered.

**Truncation** is similar in spirit to word stemming except that it is applied by the user at query time. Truncation gives the user the ability to search on word fragments. For example, the words epileptic, epilepsy and epilepsies can all be searched using the truncated form epilep*. This is an example of the most common form of truncation, right truncation. Other forms, left truncation and infix truncation, allow the system to complete the initial and the interior portion of the string respectively. The method of term storage will dictate the kind of truncation that can be used[9].

**Thesauri** are employed to control the vocabulary in an IR system. Essentially, thesauri consist of a list of lead terms that should be used for indexing and searching. Associated with each lead term is a list of related words. An entry found in the LSS thesaurus is shown in table 3.1.

**Table 3.1: Example thesaurus entry**

<table>
<thead>
<tr>
<th>accounting</th>
<th>bookkeeping</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF (use for term)</td>
<td>management</td>
</tr>
<tr>
<td>BT (broader term)</td>
<td>accounting systems</td>
</tr>
<tr>
<td>NT (narrower term)</td>
<td>cost accounting</td>
</tr>
<tr>
<td>RT (related term)</td>
<td>procurement</td>
</tr>
<tr>
<td></td>
<td>audits</td>
</tr>
</tbody>
</table>
In this example, accounting is the lead term. The associated words include terms that are hierarchically above the lead term, management, terms that are hierarchically beneath the lead term, accounting systems and cost accounting and terms that are related, procurement and audits. Another commonly used relation is the “use for” relation. This tells the system (or an indexer) that whenever the word bookkeeping is used it should be replaced with the lead term, accounting. The IR system user usually has viewing access to the thesaurus to give him a better idea of what words to use when searching for information.

Most thesauri of this nature are constructed through human effort. But automatic thesauri construction has been implemented through concordance of terms in document collections [16]. Another method of automatic thesaurus construction can be done by monitoring user queries over a period of time and analyzing the term relationships they use [9].

**Relevance feedback** is another method that can be used by an IR system to improve system effectiveness. Relevance feedback uses a priori relevance information gained from previous system users to reformulate current requests. The assumption is that the new system queries will retrieve more relevant documents and exclude more nonrelevant ones. Terms that occurred in documents previously identified as relevant are added to the original query, or if the term already occurs, weights for that term are adjusted accordingly [17].

The techniques described tend to improve document recall by generalizing the vocabulary. But by increasing recall, precision may be adversely affected. “Noise” can be created by falsely stemming a term to an unrelated root or by creating false term relationships in an automatically generated thesaurus. The detriments as well as the advantages of using these techniques should be considered and evaluated prior to their application in an IR system.
3.2 Measure of Relevance

An information retrieval system can be evaluated using various criteria. If we direct our attention to the concerns of the users of the system, Salton and Van Rijsbergen both point to six criteria in particular which are considered critical in an IR evaluation:

1. **Coverage** defines the extent to which the system includes relevant documents.
2. **Time lag** is the average time it takes to produce an answer to a search request.
3. **Presentation** designates the quality of the output.
4. **Effort** determines the energies put forth by the user to obtain the information he seeks.
5. **Recall** is the proportion of relevant material received from a query.
6. **Precision** is the proportion of retrieved documents that are actually relevant[23][17].

*Effort, time, and presentation are easily evaluated[17]. Coverage deals with the breadth of the collection and is not directly related to system performance. The last two criteria, recall and precision, measure the system’s effectiveness. How well can the system find documents that are relevant to a user’s request?*

*Relevancy is difficult to quantify because of its subjectivity. If the same query is run by different searchers, their judgement of document relevancy will differ. In experimental situations, relevancy assessments are made by experts. A set of queries are defined for which the correct responses are known. In this way, a system’s effectiveness can be established. The assumption is that if a system fares well under experimental conditions, the same performance can be expected in an operational situation. Different relevancy judgements have been noted among users and experts,*
but in general, the differences are small and therefore, do not invalidate experimental testing[23].

To quantify relevance then, two measures, recall and precision, have endured since their introduction by Kent in 1955[9]. Put simply, recall is the ratio of the number of relevant documents retrieved to the total number of relevant documents in the collection. Precision is the ratio of the number of relevant documents that have been retrieved to the total number of retrieved documents in the query result[17]. Table 3.2 illustrates the partitioning of the document collection based on binary relevance.

Note the reference to binary relevance. In this interpretation, recall and precision do not address ordered document retrieval. The quantitative values for measuring effectiveness have been motivated by the form of the retrieval results[23].

Adjustments can be made to recall and precision to accommodate ranked output by calculating pairs of values for the first $n$ documents in the result set. Tables reftable:rank, 3.4 and Figure 3.5[22] show how these results can be interpreted.

This is not to say however, that recall and precision are the only measures of relevance. Table 3.5[9] lists other criteria upon which to base relevance. All these ratios can be calculated from the contingency table in 3.2. Advantages of computing recall and precision instead of these other ratios is that recall and precision are generally accepted and the values produced are well understood.
Table 3.3: Sample ranked documents

<table>
<thead>
<tr>
<th>ranked output</th>
<th>relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 3.4: Recall and precision values for sample documents

<table>
<thead>
<tr>
<th>no. retrieved $(a + b)$</th>
<th>no. relevant $(a)$</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.2</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.4</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.4</td>
<td>0.66</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0.6</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Figure 3.5: Recall and precision graph for ranked output
Table 3.5: Other measures of relevance

<table>
<thead>
<tr>
<th>formula</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{c}{a+c}$</td>
<td>Complement of recall. Probability of a miss.</td>
</tr>
<tr>
<td>$\frac{b}{a+b}$</td>
<td>Complement of precision. Noise factor.</td>
</tr>
<tr>
<td>$\frac{b}{b+d}$</td>
<td>Conditional probability of false drop. Fallout ratio.</td>
</tr>
<tr>
<td>$\frac{d}{b+d}$</td>
<td>Complement of fallout. Correct rejection or specificity.</td>
</tr>
</tbody>
</table>
Chapter 4

Combining OCR and IR

In the practical, long term sense, the ultimate objective of character recognition is some form of subsequent retrieval. Otherwise, recognizing and storing data of any sort would be analogous to a blackhole. OCR has several applications, but the form of retrieval that complements document recognition is of course full-text retrieval. We feel, at least in this setting, the definitive test of the OCR’s goodness can be based on the information extracted from optically recognized documents. With this in mind, we describe an overview of our experiment’s purpose.

4.1 Overview of Experimental Purpose

A number of full-text applications use both OCR and text retrieval to capture and retrieve data respectively. But there is an interim step that is common for most of these applications: manual correction. One full-text application in particular that applies these methods is the Licensing Support System (LSS). The LSS is a planned system that will capture and track documents pertaining to the site licensing proceedings of the Nuclear Regulatory Commission. Eventually, this system will need to provide access to millions of documents. A prototype of the LSS was designed to determine the methodology of setting up such a large and diverse collection. The bottleneck and most costly step in the methodology was found to be the correction of OCR
errors[3]. A 99.8% accuracy rate was to be attained for the corrected document set. But whether this level of accuracy is necessary is one of the questions we attempt to answer through our testing. Our purpose is to determine the effect errors in optically recognized documents will have on retrieval.

We use two document sets that are identical except in one respect: one set is a 99.8% correct version, the other is an automatically recognized version. These two sets are queried to determine the effects the OCR’d version has on the information retrieval system.

After making observations about the characteristics of OCR data, we designed an automatic post-processing system that improves document recall on this kind of input. This system was applied to the OCR document set and the same set of queries were rerun against it. A description of the environment within which our testing was done, our method of evaluation, and our experiment results follow.

4.2 Experimental Environment

Our experimental environment is unique in the sense that we were given a set of documents by the Department of Energy (DOE) that had been manually corrected together with their corresponding images. These documents were part of the LSS prototype system previously described. We use both the corrected ASCII and the images generated by the LSS in our testing environment. Although we do not use the complete LSS prototype database, our document set was selected without bias. The set consists of 204 documents, for which we have images, corrected ASCII text on line, and hard copy.

Our collection is heterogeneous. There are numerous fonts, differing qualities of hard copy, and there is a diversity of content. The documents are scientific in

---

1 Actual accuracy rates are unknown
2 to a level of 99.8% character accuracy[11]
nature. They contain formulas, graphs, photos, and maps. All sixteen subject areas (concepts) contained in the complete LSS are covered by our 204 documents. We use the full document text, with documents ranging from a single page to 679 pages and an average length of thirty-eight pages. For a more complete description of these documents, consult [11].

4.2.1 Scan and OCR environment

The scanning of the images was not controlled in this experiment. The images produced by the contractors of the LSS are the same images we use for our testing. The use of these gives more credibility to our experimentation in the sense that they can be considered real-world samples. According to our records, the images were produced with either a Ricoh or Fujitsu scanner at 300 dpi[11]. We have no information on thresholding.

The scanned images were converted into a format usable by ISRI's vendor-independent interface[11] prior to the OCR process. Each image was then recognized using ExpertVision RTK (beta version 1), a software-based OCR system for PC-DOS. For a complete accuracy assessment of this device and other OCR devices please see [13]. Eighty-one of these page images could not be recognized using this beta version, so we completed the collection using the Calera RS 9000.

We use automatic zoning for two reasons:

1. Manual zoning of 9,300 pages would have been labor intensive and time consuming.

2. The correct text had been manually zoned by the DOE contractors using a complex set of rules. There was no guarantee that the zones we selected would have matched their set exactly.

The lack of manual zoning may have had some adverse effect on the accuracy of
the corresponding output. Sciences Applications International Corporation (SAIC)\(^3\) claims "[manual] zoning... results in higher output accuracy which, in turn, reduces required OCR editing"[3]. Also as stated in [2], the presence of non-text data and noise increases the difficulty of character classification and recognition.

Another side effect of automatic zoning is the generation of graphic text. Since graphics are not always recognized as such by OCR devices, non-text data, such as maps, photos, and graphs are translated to ASCII. This erroneous translation produces lines of unreadable ASCII characters.

The process described above was performed on each of the 9,300 pages. The ASCII pages generated were concatenated into complete documents for loading into the text database.

4.2.2 Text Retrieval environment

BASISplus is the text retrieval system we use for our experimentation. This system is based on the traditional boolean logic positional inverted file methodology presented earlier. BASISplus incorporates a relational database for querying structured fields on top of its original full text retrieval system (BASIS). The inverted file model was chosen for our experimentation because it is the most widely used technology[16].

Document environment

The correct text and raw OCR document sets were loaded as continuous text structures using the default options such as stop word lists and break characters (e.g. blank . , :). Since the OCR text was not formatted neatly like the correct document set, a number of load parameters had to be adjusted before BASISplus would accept this OCR text properly. In particular, the index sort parameters needed to be adjusted. The number of "terms" to be indexed was 150,000—three times the size of the cor-

\(^3\)SAIC was one of the LSS contractors.
responding correct set. Each time a character is incorrectly translated by the OCR device, a new word is formed and in turn, indexed by the text retrieval system.

**Query environment**

BASISplus provides a query language called *Fundamental Query and Manipulation* or FQM. FQM is a command language based on boolean logic that supports wildcarding and proximity searching. These features are used infrequently in our queries. One of our queries uses wildcarding and only phrase proximity searching is employed. Although thesaurus facilities are available with FQM, none were used.

The queries we use for our testing are a subset of the LSS prototype test questions. These queries were artificially constructed to evaluate how well users were able to retrieve needed information from the database—a very different intention than ours—and therefore should reflect no bias in our testing. Many of the queries were written to retrieve information from the structured fields of the records, not the actual text. Some of these structured fields are: author name, title, descriptor field, and document type. Because of this difference, some of the original queries were excluded from our test set; many others were reworded so as to reference only the text of the document. The translation of the original English queries to their FQM representation was done by a geologist, two computer scientists, and two research assistants to ensure correctness. The interpretation of the original queries was not lost and they represent an unbiased set of seventy-one queries. Figure 4.1 is an example of an original test query, its text-only interpretation, and its FQM translation.

There are 205 unique search terms for the seventy-one queries. The average number of terms for the queries is five. The queries were quite relevant to the subset of 204 documents used in our testing since there was an average of eight hits per query. The same set of queries was automatically run on each database—no interactive searching
Test Query INJD-T3-Q1

LSS Prototype Test Question: Your office is trying to trace the evolution of NRC's position on repository sealing concepts (e.g., shaft and borehole seals). You need to produce a listing of all documents (including meeting material) discussing seals.

Text only translation: Find documents discussing repository sealing concepts (shaft and borehole seals).

FQM translation: find document where text include phrase like 'repository' & 'seal' or text include phrase like 'shaft' & 'seal' or text include phrase like 'borehole' and 'seal' order by docid

Figure 4.1: Example test query translation

was done.

4.3 Method of Evaluation

The purpose of our experimentation is to determine the effect of a single independent variable, the input data, on the performance of a boolean logic inverted file text retrieval system. The dependent variable under assessment is the retrieved documents from the queries. Keeping all other variables constant, we would like to measure differences using the number of hits returned in the correct database as a benchmark. As discussed in [21] we would like to ensure the validity, reliability, and efficiency of our experiment and its results.

First, we would like to point out that we are not trying to evaluate each individual technology separately—we are evaluating the results of their synthesis. This unification introduces a number of possible variations for these experiments: different scanners, different settings, different OCR devices, and different text retrieval systems will give different results. But relative to the environment we have used for our experiments, we believe our testing is valid. The independent variable, the OCR
input data, is a good indicator of the concept under investigation[21].

The reliability and efficiency of our testing stems from the diversity and size of the collection we use. Although the number of documents may seem small in comparison to other text retrieval experiments, the number of pages (9,300) and the number of index terms (150,000 in the OCR database), is quite sufficient for the kind of testing we do.

The technologies we use represent a reasonable sample of the those currently available. The OCR and IR systems, the input data, and the queries were not selected or designed with this kind of testing in mind. Their selections were not only independent of this experiment, they were independent of each other. Further, since no human influence is introduced in our retrieval testing, many of the considerations for evaluating experimental results[21] are eliminated.

The only factor that could possibly alter our results to some degree would be a modification in the definition of correct text. We state the correct text has a 99.8% character accuracy. We assume this measure to be correct; however, we only performed a cursory scan of the text. Further, a complex set of rules was used to determine the formatting, inclusion, and exclusion of text. If these were changed, it may affect the outcome.

Although precision and recall are the standards for evaluating performance, we do not use these criteria for our current measure of evaluation. Instead, we report on the comparison of the result sets for each query run on both collections. This evaluation method, although simplistic, will indicate the effects optically recognized text will have on an IR system. Since it turns out that, in general, these result sets are identical, we do not expect a significantly different conclusion if precision and recall are used. We would eventually like to consider precision and recall, and also ranking[14][17] as a means of evaluation on a larger test set.
Table 4.1: Experiment 1 query results

| Total number of documents retrieved for correct data | 632 |
| Total number of documents retrieved for OCR data     | 617 |
| Percentage returned                                  | 97.6% |
| Number of queries for which result sets are identical| 63  |
| Number of queries for which result sets are different| 8   |

4.4 Results: experiment 1

Experiment 1 includes the loading, querying, and comparing of the correct document set with the raw OCR set. The results of the seventy-one queries that were run on the 204 documents appear in Table 4.1. Of the seventy-one queries that were run, sixty-three of the OCR database result sets were identical to the correct database result sets. For these 71 queries, there were a total of 632 documents returned in the correct database and 617 in the OCR database. Fifteen documents were missing from the OCR result sets. The source of errors for these fifteen missing documents can be found in Table 4.2. Since the images were not generated by us, we do not correct errors caused by poor scanning or bad hard copy. But by massaging these OCR documents, removing end-of-line hyphenations, and making some spelling corrections, the other missing documents should be retrieved. This kind of automatic document processing is described in [20]. Cleaning up this OCR text leads us to the second version of our experiment: re-examining the query results after automatic correction of the OCR text with the post-processing system.
4.5 Results: experiment 2

Experiment 2 is experiment 1 with an additional processing step. Before loading the OCR documents into the text database, they are filtered through an end-of-line hyphenation remover and the post-processing system. No manual correction was made to these documents; only two automatic processes were applied. For this set, the break character list was adjusted to aid in the location of misspellings. For example, if the OCR device cannot make a decision on what a character should be, it puts a “~” in its place. Since the tilde is a default break character for BASISplus, this substitution caused incorrect word breaks and therefore, partial words were indexed. Although these adjustments helped the post-processing system locate errors, it may have had an adverse effect on other properly indexed terms. Evaluation of this effect was not considered.

The results of the query retrieval are documented in Table 4.3. Since no attempt was made to improve the images by rescanning, the errors due to poor images are not corrected. Of the remaining ten, the automatic post-processing corrected seven. Only three documents were not recalled. It is difficult to say whether two of the remaining three errors can actually be attributed to OCR error. The only necessary condition for both these documents to be retrieved was the inclusion of the string SCP. Both documents in the correct database had had only a single occurrence of this string. After examining the hard copies, we found the string was not part of the original
Table 4.3: Experiment 2 query results

<table>
<thead>
<tr>
<th>Total number of documents retrieved for correct data</th>
<th>632</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of documents retrieved for OCR data</td>
<td>624</td>
</tr>
<tr>
<td>Percentage returned</td>
<td>98.7%</td>
</tr>
<tr>
<td>Number of queries for which result sets are identical</td>
<td>65</td>
</tr>
<tr>
<td>Number of queries for which result sets are different</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.4: Experiment 2 source of errors

| Poor original images or hard copy                  | 5   |
| Missing SCP string                                | 2   |
| Incorrect OCR translation not corrected by the post-processing system | 1   |
| Total                                              | 8   |

document text and therefore was not relevant to the query. In any case, these are counted as errors in Table 4.3.

Since there is little room for improvement from experiment 1 to experiment 2, the impact of the post-processing system is not obvious. But 1100 misspellings of the 205 distinct query terms were actually corrected in the OCR’d text.
Chapter 5

Conclusion

Optical character recognition together with information retrieval encompass the task of producing accessible information. Instead of analyzing each technology in separate domains, if the application permits, we believe the two can and should be assessed as a singular system. This is what we demonstrate in our testing.

The results of our preliminary experiments intimate that, at least in an environment where 100% accuracy is not imperative, optical character recognition and Information Retrieval can be applied in succession with little human intervention. This is a consequential result since realizing 100% accuracy, even with manual correction, is nearly impossible. The prototype simulation efforts of the Licensing Support System proved how difficult this task could be [3]. Some compromise then must be made on accuracy. The amount of compromise should depend on the given application.

We would like to note however, that we have addressed only a single issue in our experimentation: retrieval results. Through our testing, other issues became apparent. For example, we know the index is artificially enlarged due to misspellings and "graphic text" strings. How will this added overhead affect an IR system’s performance? Further, how will user confidence in an IR system be affected by noisy data? Certainly, if a user is presented with OCR text, they may lose trust in the system, even if it has been shown that retrieval results are equivalent to a
corrected document collection. Therefore, redesign of the IR interface would be in order. Moreover, the documents we use are full-text documents. Some IR systems store and retrieve on titles and abstracts alone. The important terms in these kinds of databases may occur only once, and if misspelled, will more drastically affect retrieval.

Still, we feel the two technologies have progressed to a point where they can be used in combination to build an information system. Continuing research in this area should help close the gap between them.
Bibliography


