Machine learning in a multimedia document retrieval framework

by M. P. Perrone
G. F. Russell
A. Ziq

The Pen Technologies group at IBM Research has recently been investigating methods for retrieving handwritten documents based on user queries. This paper investigates the use of typed and handwritten queries to retrieve relevant handwritten documents. The IBM handwriting recognition engine was used to generate *N*-best lists for the words in each of 108 short documents. These *N*-best lists are concise statistical representations of the handwritten words. These statistical representations enable the retrieval methods to be robust when there are machine transcription errors, allowing retrieval of documents that would be missed by a traditional transcription-based retrieval system. Our experimental results demonstrate that significant improvements in retrieval performance can be achieved compared to standard keyword text searching of machine-transcribed documents. We have developed a software architecture for a multimedia document retrieval framework into which machine learning algorithms for feature extraction and matching may be easily integrated. The framework provides a "plug-and-play" mechanism for the integration of new media types, new feature extraction methods, and new document types.

One of the most powerful benefits of electronic documents is the ability to retrieve information automatically from a database on the basis of some search criteria. The last few years have seen accelerating progress in methods for multimedia retrieval, including methods based on text meta-data attached to nontext media (e.g., text annotations of speech and video) and methods based on automatic extraction of nontext characteristics of multimedia that can be used for nontext queries such as images. Some of these are based on human-generated text descriptions and indexing of these documents, some on automatic generation of text descriptions (e.g., face recognition), and some on abstract query-by-example methods (e.g., locating images with color histograms similar to a sample image).[1,2] Progress in the last is clearly illustrated in the recent Multimedia Content Description Interface, MPEG-7, work of the Moving Picture Experts Group to standardize the description and representation of multimedia features for retrieval purposes.[3]

Speech, scanned text, and handwritten documents have been made more accessible for retrieval by using machine learning algorithms. These algorithms generate text transcriptions and then use conventional text search technology to retrieve the corresponding nontext document. Matches from the text search are then used to retrieve the original scanned or speech documents. If precise transcripts of these documents exist, information retrieval (IR) techniques can be applied; however, such transcripts are typically too costly to generate by hand, and machine learning methods for automating the process of tran-

0018-8670/02/$5.00 © 2002 IBM

script generation are far from perfect.[4] Thus, such transcripts are usually incomplete or corrupted by incorrect transcriptions.

It has been observed[5] that IR is not significantly degraded when the documents to be retrieved are machine-printed documents that have been transcribed using machine optical character recognition (OCR) methods. Apparently, OCR of machine-printed documents is sufficiently accurate. When transcription is inaccurate, word redundancy in the target documents may compensate;[6] however, in general, sufficient word redundancy cannot be assumed, especially for short documents.

The problem of transcription errors on retrieval in the context of speech has been addressed. One approach[7] relies on query expansion, a second approach[8] employs a variety of string distance methods, and a third approach[9] uses global information about probable phoneme confusions in the form of an average confusion matrix for all data observed but does not handle confusions at the individual word instance level.

A class of successful approaches uses template matching between handwritten queries and handwritten documents;[10–13] however, this method can be very slow if the number of documents to be searched is large and the match method is very complex; also, this method does not allow for text queries. Another approach[14] successfully used pieces of handwritten words to handle inaccuracies in machine transcription. This approach attempts to reduce the complexity of the transcription process at the expense of allowing certain words to become ambiguous. As one might expect, this approach was found to work well in domains in which words were long and easily distinguishable but less well in domains with many similar words.

Current search engines are fragmented in that each engine handles a single media type, or a limited set of media types, and these engines are not easily interoperable. Each time such a system is constructed, similar design issues are revisited again and again, repeating existing work and leading to stand-alone systems that can utilize one another's search capabilities only after a considerable effort at integration. As a way to avoid these problems, we are prototyping a flexible and extensible Multimedia Document Retrieval (MDR) System. This MDR System will help achieve four goals: Provide a uniform search facility upon which client applications can be developed without regard to media-specific issues; provide a mechanism to streamline the implementation, testing, and distribution of new multimedia search algorithms; provide a framework for leveraging existing search algorithms; and provide a means for naturally extending to "cross-media" searches, i.e., searches between two or more different media types.

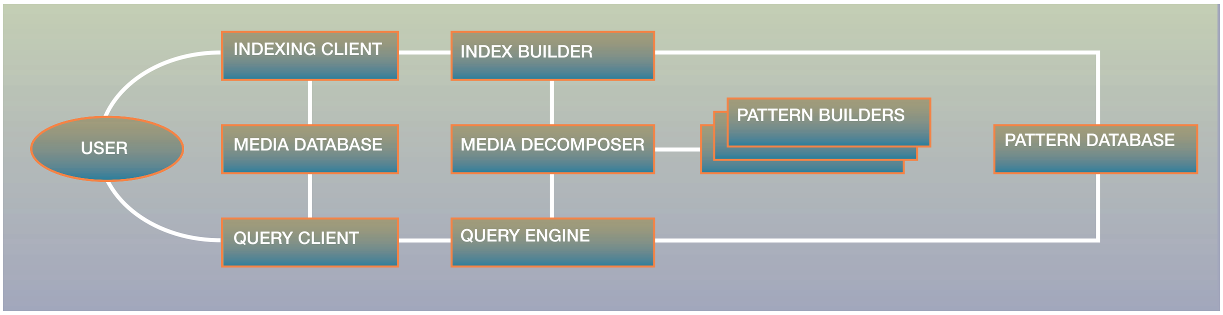
In the next section, we address the issues facing multimedia document retrieval by describing a flexible, extensible, "plug-and-play" multimedia document retrieval framework. In the third section, we describe specific details of the MDR approach when applied to the task of handwritten document retrieval. In the fourth section, we describe handwritten document retrieval experiments based on using the IBM handwriting recognition engine to construct pattern objects. In the fifth section, we present the results of these handwritten document retrieval experiments. The last section summarizes our findings.

The MDR System

Our prototype, the Multimedia Document Retrieval (MDR) System, shown in Figure 1, performs two basic functions: the indexing of multimedia documents and the retrieval-by-query of indexed multimedia documents. We begin by giving a high-level overview of indexing and retrieval and then present component details.

Indexing. Before retrieval is possible, an index must be built. The index is built in the following manner. A user interacts with an indexing client and requests that multimedia documents be indexed. The indexing client is responsible for verifying that the corresponding media types are supported, locating and retrieving the documents from the media database, and passing them to the index builder along with any user-specified indexing preferences. The index builder is responsible for converting the documents it receives into statistical representations that it then stores in the pattern database. Documents are processed by passing them to a media decomposer that is responsible for decomposing a multimedia document into its constituent "primitive" media elements (e.g., an MPEG file might be decomposed into audio and video). Once a document is decomposed into primitive media elements, the media elements are then passed to their corresponding pattern builders that are responsible for generating statistical representations of the media. These representations are called pattern objects. The pattern objects are passed

Figure 1 The MDR System



back to the index builder, which then adds them to the pattern database.

Retrieval-by-query. Once an index has been built, a user may search the corresponding documents by submitting a multimedia query to the query client. The query client is responsible for constructing queries, validating that the media types in the query are supported, and submitting the query to the query engine. The query engine uses the media decomposer and pattern builders to convert the query into pattern objects, which are then compared to the pattern database entries using a pattern similarity metric. The comparison process results in a relevance score for each indexed document. These scores are then passed to the query client, which retrieves documents from the media database ranked by their relevance to the query.

Component details. The media database may be any repository or set of repositories where media documents are stored. The documents themselves are identified using URLs (uniform resource locators) and may consequently be stored in a local or remote file system, database, FTP (File Transfer Protocol) site, or Web server.

The most general goal of the MDR System is to enable retrieval of documents of any media type using queries of any media type. This goal requires the ability to compare content in various media types. What is needed are compact media representations for which a measure of approximate match is easy to calculate. In the MDR System these representations are the pattern objects. A pattern object has four attributes: a URL, an extent, pattern data, and a pattern similarity metric. The URL points to the document from which the pattern object is derived. The extent describes which subset of the document corresponds to the pattern object. The pattern data is some representation of the media within the extent. The pattern similarity metric measures the similarity of two pattern objects. The pattern objects are the core of the MDR System and therefore must be chosen with care. For nontext media such as handwriting, speech, and video, machine learning algorithms are needed to assist in the construction of pattern objects.

A pattern builder is the component of the MDR System that converts specific, primitive, media elements into sets of pattern objects. Machine learning algorithms can be used to convert the media into statistical representations of the media. Different pattern builders may be designed to extract the same pattern object type from different media types, thus enabling cross-media search and retrieval.

The pattern objects are stored in a pattern object database and are retrieved using database queries that are themselves converted into pattern objects, which search the database for pattern objects similar to themselves.

The retrieval process determines the relevance value of each document to a user-specified query. The documents indexed by the pattern object database can then be sorted and retrieved based on these values. In general, queries are in disjunctive normal form and may also include additional constraints on metadata such as creation time or authorship. The MDR System represents these complex queries as trees in which the leaf nodes contain the media elements and the parent nodes contain the relation information.

The query engine contains the functionality for obtaining query results, using the query tree and the pattern database. Complex queries are processed by passing relevance scores from leaf nodes up the tree structure to the root node, merging relevance estimates according to the relation information built into the tree. The pattern database indices are used to prune the number of pattern objects that must be examined with the similarity metric. In addition to this low-level search optimization, standard database query optimization techniques may be applied to the query tree.

Plug-and-play. A major goal of the MDR System is to create a "media agnostic" framework into which media-specific components can be easily and seamlessly incorporated. The MDR System allows researchers and developers to focus on the media-specific aspects of their work, while taking advantage of the media-independent services, i.e., query trees and merge methods. Additionally, the MDR framework provides abstract classes that a developer extends and implements as needed for specific media types.

Example: Handwritten media

In the context of the MDR System, we have investigated the document retrieval performance of several pattern objects for handwriting. In this case, the media elements are handwritten documents.

Pattern object: The *N*-best list. This paper uses a statistical classifier to convert each word of a handwritten document into a set of word or score pairs, one for each of the most likely text translations of the handwritten word.[15] This approach is robust when there are transcription errors (which is of particular importance for low-frequency words) and has the ability to retrieve words that are not in the lexicon of the machine transcription system.

This set of scores is termed an "*N*-best list." In practice, each handwritten word in each document is converted into an *N*-best list. This step need only be done once. Each word of a handwritten query is likewise converted into an *N*-best list. For a text query, each word is converted into a trivial *N*-best list by giving a maximum score to the query word and a minimum score to all other *N*-best list entries. (That is, we assume no noise in recording a text query, though this assumption could easily be relaxed.)

Let $\mathcal{W}$ be the set of all possible words and let $\mathcal{I}$ be a given handwritten occurrence of $w \in \mathcal{W}$. We define the *N*-best list associated with $\mathcal{I}$ as the vector $\vec{S}(\mathcal{I}) = (S_1(\mathcal{I}), S_2(\mathcal{I}), \ldots)$, where $S_i(\mathcal{I})$ is the score of $\mathcal{I}$ given w_i, the i th word of $\mathcal{W}$, according to some machine transcription system. In this paper we used an HMM (Hidden Markov Model)[16] trained on an unconstrained, writer-independent data set to calculate $S_i(\mathcal{I})$ as a measure of the HMM's probability of $\mathcal{I}$ given w_i. In practice, we set a threshold for $S_i(\mathcal{I})$ to disregard low scores, which results in *N*-best lists averaging approximately 16 nonzero entries. For the rest of the paper, we drop explicit reference to $\mathcal{I}$.

In standard handwriting recognition systems, the *N*-best list is the final result of the recognition process and is used to indicate the correct transcription of the handwritten word. In some systems, if the first word in the *N*-best list is incorrect, a user can optionally select another word from the *N*-best list.

Motivation for *N*-best-list patterns. *N*-best list retrieval compensates for transcription noise by allowing the search for words to go beyond the best match from the transcription process, i.e., the top word in the *N*-best list. For example, suppose a typed query of "cat" is used and the corresponding retrieval handwriting had the following *N*-best list (sorted by score):

Word	cut	cot	cat	lot	let	···
Score	100	95	94	10	5	···

If only the top-scoring word ("cut") is used, the correct handwriting will not be retrieved; however, if the *N*-best list is used, a well-designed method could use the additional information to detect that the corresponding handwriting is more closely related to the query than the simple transcription would suggest.

N-best lists can also correct for transcription noise caused by words that are unknown to the transcription model (e.g., proper names, symbols from foreign languages, or even nontext handwriting such as arrows, circles, doodles, etc.). In many transcription systems, such words cannot appear in an *N*-best list. However, if a writer writes consistently, the statistical structure of the *N*-best lists of various handwriting instances of the same thing should be similar, and that similarity should yield a good match of *N*-best lists.

One might think that the addition of so many words through the use of *N*-best lists would add significant amounts of noise to the retrieval process; however, the likelihood of retrieving the wrong document is not significantly increased by the use of *N*-best lists. This can be understood by considering the following: Typically an individual writer's set of query words is much smaller than the set of all possible words; therefore, the likelihood is low that a transcription error will lead to a high score for a query word in an *N*-best list that does not correspond to that query word. Thus, incorrect document retrieval, i.e., a high score for the right word in the wrong *N*-best list, is not significantly increased as a result of the use of an *N*-best list. Of course, if queries have very similar handwriting representations (e.g., "puppy" and "puppet"), cross-confusion may be a problem.

False positives occur when the transcription engine gives a high word score to handwriting that is incorrectly transcribed. Incorrectly transcribed words other than query words rarely generate false positives, since with a large vocabulary of known words, the engine will more often select words other than the query word for the *N*-best list.

In summary, the *N*-best list, combined with knowledge of the behavior of the HMM, provides a more comprehensive description of the document, while still facilitating effective search techniques.

***N*-best list similarity metrics.** Our *N*-best list retrieval methods work by defining a metric between the *N*-best list from a query and the *N*-best lists from a database. The metric scores for each *N*-best list in a given document are combined to generate a relevance score for the whole document. Documents are then ranked by their relevance scores and retrieved in rank order. The relative retrieval performance using various metrics can then be examined. We now define some metrics for which we have done experiments.

Text metric. Handwritten documents can be retrieved using conventional text searches on machine transcribed text, with links back to the original handwritten documents. We used this method as the baseline for our experiments. The text transcription for each document is simply the text assembled by taking the highest-scoring word from each *N*-best list. The search terms are ASCII strings, taken from the hand-generated ground truth. The metric score is one or zero, depending on whether the query word matches any document word. The document score is the sum of the metric scores.

Ranked text metric. The traditional text search can be enhanced by including other words from the *N*-best list generated by the transcription model. As a simple trial, we took the top three words from the *N*-best list and weighted them solely by rank: 1.0 for the top word, α for the second word, and β for the third word, where α and β were greater than zero and optimized on an independent data set. We then searched through this expanded document for the single ASCII search term. The metric score is the rank score of matching words. The document score is the sum of the metric scores. This metric score will always be equal to or greater than the text metric, since the top word still has a weight of 1.0. However, the contributions of the other words cause additional documents to have nonzero scores and can change the rank ordering of the documents. This metric is very convenient and powerful because it requires very little information from the transcription model. Only the first three candidates need be stored and indexed, and no score information is needed.

Scored text metric. In this metric, the expanded document includes up to 20 words from the *N*-best list. Each word is weighted proportionally to the score assigned it by the transcription model, normalized so that the sum of the scores for all the alternate words for one piece of handwriting sums to 1.0. The metric score is the matching word score, and the document score is the sum of the metric scores.

Dot-product metric. The dot-product metric between a query *N*-best list, $\vec{q}$, and a document *N*-best list, $\vec{d}$, is given by

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}||\vec{d}|} \tag{1}$$

which is always between 0 and 1 since the *N*-best list scores are nonnegative.

From the *N*-best list perspective, the dot product is the sum of the products of the normalized scores of words that appear in both the query *N*-best list and a document word *N*-best list. The score for a document is the sum of the dot products of all the *N*-best lists in the document with the *N*-best list for the query handwriting.

Experiments

The data used in these experiments were collected from 108 writers, each of whom was asked to select from one of three topics and hand-write a one-page document. The categories were "thank you note," "room description," and "defective product letter."

Within a category, the writers were free to write whatever they wanted. The documents were then transcribed by hand. The result was a database of 108 documents with a total of 10985 words. These documents were collected using the CrossPad** notebook files and the IBM InkManager* software.

The data set size in this experiment is much smaller than typical databases used in text retrieval research, though it is typical of available on-line handwriting recognition databases. However, from a speed and resources point of view, we expect our system to scale almost identically to keyword retrieval of text documents because standard indexing methods can be used equally well with the N-best lists. From a retrieval accuracy point of view, we do not know whether the performance gains will increase or decrease as the data set size scales, but we believe it will continue to outperform keyword search because of the error-tolerant nature of the algorithms.

Each document was processed by a heuristic clusterer[17] to group handwriting data. Handwriting clusters were then normalized and had features extracted for processing by a multistate, Bakis-topology, lexeme-based HMM.[16] The HMM returned an N-best list for each word in each cluster of the top scoring words from a 30000-word lexicon. The HMM was trained in a writer-independent fashion on data from over 200 writers. Thus, each document in our database was converted into a set of word N-best lists that summarize the scores of the HMM for the most likely words for each cluster of handwriting.

The total number of words in all N-best lists after this process was approximately 176000, of which approximately 28000 were unique out of a total possible 30000-word lexicon.

Query generation procedure. The handwritten data used for this paper were not originally collected for the purpose of testing IR techniques, and no attempt was made to collect separate handwritten query handwriting or even to identify what queries would be appropriate. Thus we were faced with two tasks: identifying plausible query words and obtaining the corresponding handwriting.

Identifying query words. Query words were selected based on word frequencies and availability of other renderings of the same word. These queries were taken from documents from other writers. Ideally, we would have used query words written by the writer of each letter; however, those data were not collected since this data set was originally collected for another purpose and, after the fact, the writers were not available to write additional words as queries. Thus, the results presented here are lower than one would expect if the same writer had written both the queries and the documents to be retrieved.

We take a standard approach[18,19] to query selection. We define tf(t, d), the term frequency, as the number of occurrences of a term t within a given document d. We define idf(t), the inverse document frequency, as the inverse of the number of documents in the database that contain a term t. We selected queries for each document by choosing the terms in the document that had the highest tf$(t, d)*$idf(t) product, subject to the constraint that the query words exist in the ground truth for at least one other document, to ensure that handwriting is available outside the target document for use as a query. We chose the five words from each document having the highest tf$*$idf product subject to this constraint.

The various document scores were: the word count for text metric, the sum of the rank weights for ranked text metric, the sum of the metrics for scored text metric, and the sum of the dot products for dot-product metric. We did not use any weighting of query words (e.g., Okapi[20,21]) since the query words were chosen based on tf$*$idf, so the differences in tf$*$idf would be modest. Because of the small size of the documents, we do not expect high query word redundancy in any of the documents.

Obtaining handwritten queries. Since our document database did not include word-level ground truth, we relied on the output of our handwriting recognizer to identify query handwriting from documents other than the one for which the query word was selected. From all of the documents that included a query word in their ground truth, we selected the N-best list with the highest score for the query word. Because of errors in machine transcription, it is possible that some query handwriting corresponded to a word other than the desired query word. Thus, the transcription-based algorithms may retrieve documents using the wrong query handwriting, making the results more pessimistic than they would be in practice.

Since the query word is drawn from one of the documents in the database, single-word queries include as part of their "truth set," i.e., the set of documents that are correct to retrieve, the document from which the word was drawn. Consequently, the reported

Figure 2 Single word query performance over all 540 one-word queries

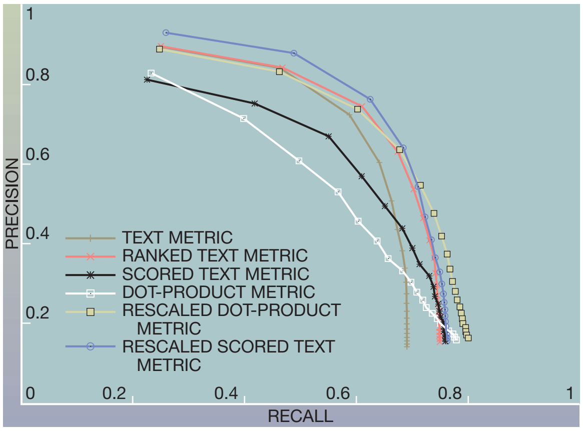
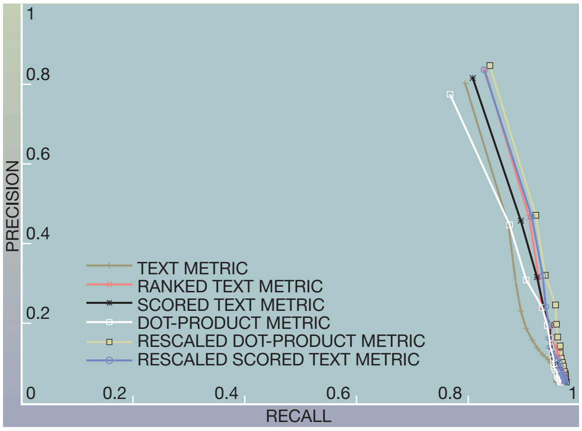


Figure 3 Two-word query performance averaged over all 1080 two-word queries



single-word retrieval performance may be optimistic. For the multiword queries, the documents that the query words are drawn from generally drop out of the truth set. Thus we expect that the results for the multiword queries may actually be slightly pessimistic because the document from which the word was drawn will have an elevated score due to an exact match to one of the query words. This artificially elevated score triggers a false positive.

Document scoring for multiword queries. A previous paper[15] reported the behavior of N-best list retrieval of single words. Here, we focus on multiword handwritten queries of multiword handwritten documents. For multiword queries, the score for a document was determined by multiplying the single term scores for the document. This process corresponds to an "AND" operation over all query words. In order to prevent documents from dropping out if a single query word was missing, a small positive number was added to all word scores. This addition introduced a very large penalty for missing query words, but the document would still stay in the result list. Primarily affected is the high recall region of the precision-recall curves.

Precision and recall are used to measure the retrieval performance of the system. Precision is the percentage of the documents retrieved by a query that are correct, and recall is the percentage of all correct documents that are retrieved by a query. Below, we define these terms more precisely.

For each document d, we calculate a relevance score to query q. Let the truth set of q be the set of documents whose ground truth text contains the ground truth of q. Let $n(q)$ be the number of documents that are in the truth set of query q. Let $nr(q, \theta)$ be the number of documents with a relevance score to q above a threshold, θ. Let $nc(q, \theta)$ be the number of documents with a relevance score to q above θ which are also in the truth set of q.

Using these definitions, we define precision and recall as follows:

$$\text{Recall}(\theta) = \sum_q \frac{nc(q, \theta)}{n(q)} \tag{2}$$

$$\text{Precision}(\theta) = \sum_q \frac{nc(q, \theta)}{nr(q, \theta)} \tag{3}$$

The precision-recall curves implicitly parameterized by θ are shown in Figures 2, 3, 4, 5, and 6. Only results of queries using the same number of words were averaged. One-word queries had a truth set size of 3.9, on average, and a variance of 2.3. On average, two-word queries had about 1.2 correct results per query. Most had one correct result, a significant number had two correct results, eight queries had three correct results, and one query had four correct results.

Averaging query results. Five words were selected from the actual text of each document for use as query words. For each retrieval method, a query was performed with every possible combination of the five query words from each document. This generated five single-word queries, ten two-word queries, ten three-word queries, five four-word queries, and one five-word query from each document in the database.

The ground truth for each query was determined simply by finding whether all the query words used for a given query appeared in the actual text of each of the documents. Thus the document truth set varies depending on which subset of the five query words is used. Each query had at least one correct result document, since each group of five query words was chosen from the actual text of one document.

For a fixed number of query words, retrieval results were averaged over all possible combinations of subsets of that size from the five query words. This averaging helps to reduce retrieval performance variability that may occur because of the inherent variability of the handwriting representations, and to account for the fact that an individual may attempt to retrieve a document using one of a variety of different queries.

Metric optimization. We explored optimizing the scored text metric and the dot-product metric by replacing them with simple functions of the original scores, based on *N*-best list rank. We optimized candidate functions using an independent data set of simple single word queries of words in a small database of approximately 1100 handwritten word samples from 78 writers. The area enclosed by the precision-recall curve (see earlier discussion on document scoring in this section) obtained by dot-product queries in the 1100 word database was used as the optimization criterion, a reasonable overall measure of retrieval performance.

We optimized by substituting the score at each rank with a linear function of the score:

$$s_i' = \alpha_i s_i + \beta_i$$

where s_i is the original ith rank score, s_i' is the new score, and α_i and β_i are the global parameters that were optimized. For this case, we ran a Monte Carlo optimization of a few thousand trials, concentrating the variation in the parameters for the higher ranks. We then looked at the sets of parameters that generated the best results, averaged them, and rounded them off.

Figure 4 Three-word query performance averaged over all 1080 three-word queries

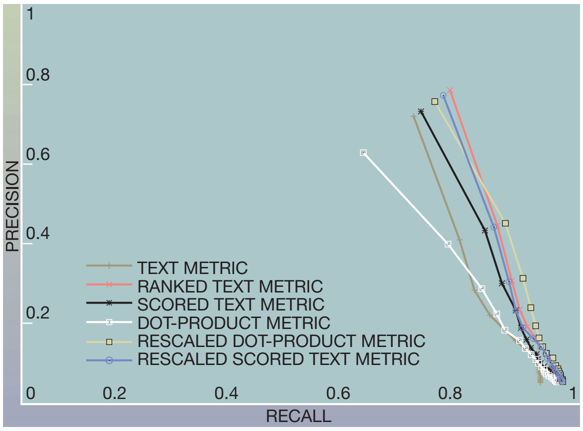
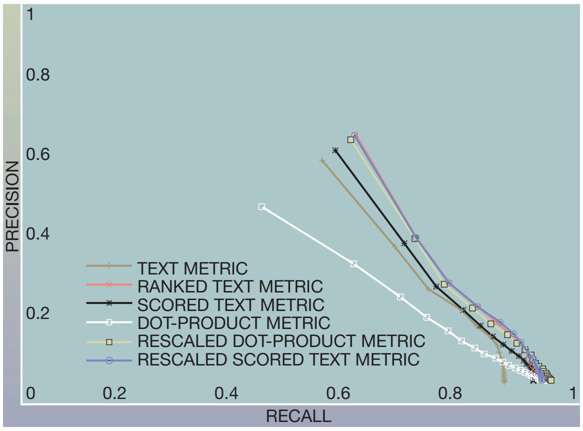


Figure 5 Four-word query performance averaged over all 540 four-word queries

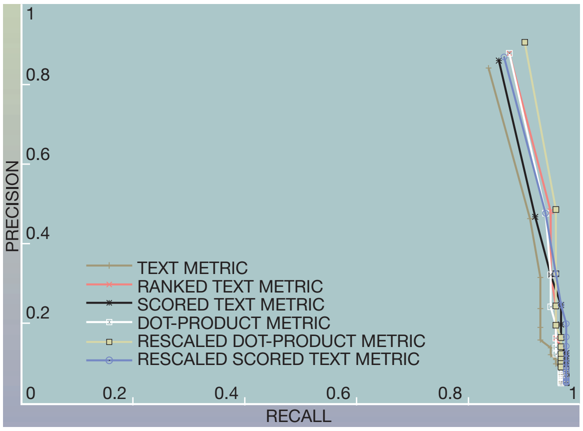


The *N*-best list dot-product metric with rescaled scores appears in the results as “Rescaled Dot-Product Metric,” and the scored text metric with rescaled scores appears as “Rescaled Scored Text Metric.”

Results

Figures 2, 3, 4, 5, and 6 show the average precision-recall curves for one-, two-, three-, four-, and five-word queries. Each graph contains one curve for each of the six retrieval methods (as discussed previously in the third section). The text metric in each graph

Figure 6 Five-word query performance averaged over all 108 five-word queries



may be considered the baseline performance against which the other metrics should be compared.

Single word queries. Inspection of the precision-recall curves shows that the different algorithms produce curves of substantially different shape, so the choice of best algorithm depends on the regime. In the low-recall/high-precision regime for single-term queries (Figure 2), the rescaled scored text metric search actually had the best average performance. This performance persists up to about 70 percent recall, at which point the rescaled dot-product search has the best performance. Note that this regime dependence diminishes as query word count increases.

With the exception of the rescaled score text metric method, the text metric is actually on par with or better than all other search strategies in the regime up to 50 percent recall. All other strategies are superior at recall levels above 70 percent.

Multiword queries. In general, the performance improved dramatically as more query terms were added. Improvement is partly a result of the reduction in the truth set size, which is the denominator of recall. It can also be seen from the graphs in Figures 2–6 that as the number of query words increases, the regime dependence observed in the single-word queries diminishes, so much so that for five word queries, there are no crossovers between the baseline method and the other methods. Also note that as the number of query words increases, the baseline method (text metric) gradually falls in relative performance until it is the worst performing method.

Conclusions

We have described the MDR System and have shown how its "media agnostic" approach leads to a flexible, extensible, plug-and-play system for multimedia and cross-media information retrieval.

In the context of the MDR System, we have developed IR algorithms for handwritten documents that are suitable for both typed and handwritten queries. We have demonstrated that document expansion using *N*-best lists can provide improvements in precision and recall compared to simple text metric, even using the most lightweight method (ranked text metric), and that these improvements persist for multiword queries. Additional improvements were illustrated using more complex metrics. The improved searchability that we have demonstrated has the potential to make handwritten databases much more valuable to users.

The methods described in this paper have the additional benefit that they are text-based. Unlike template matching methods, these methods can leverage much of the existing text IR technology and enable one-time preprocessing and indexing of an *N*-best list. Furthermore, the approaches presented here do not rely on word redundancy to overcome transcription errors. This is borne out by the positive results we obtained on the very short documents that comprised the database used. And finally, dot-product methods have the potential to retrieve words or symbols that are not in the transcription lexicon.

We are currently working to improve and extend both the MDR System and our handwriting-specific components.

Acknowledgments

The authors would like to thank Alain Biem, Tom Kwok, Millie Miladinov, John Pitrelli, Gene Ratzlaff, and Jay Subrahmonia for data collection and useful discussions.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of A. T. Cross Company.

Cited references

1. D. Gibbon, A. Basso, R. Civanlar, Q. Huang, E. Levin, and R. Pieraccini, "Browsing and Retrieval of Full Broadcast-

Quality Video," *Packet Video 99*, New York (April 26–27, 1999).

2. A. Mojsilovic and B. Rogowitz, "Capturing Image Semantics with Low-Level Descriptors," *Proceedings of the International Conference on Image Processing* (October 2001).
3. See the Moving Picture Experts Group (MPEG) home page at http://mpeg.telecomitalialab.com/.
4. R. Plamondon and S. N. Srihari, "Online and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, No. 1, 63–84 (January 2000).
5. K. Taghva, J. Borsack, and A. Condit, "Results of Applying Probabilistic IR to OCR Text," *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* (July 1994), pp. 202–211.
6. J. Nielsen, V. L. Phillips, and S. T. Dumais, "Information Retrieval of Imperfectly Recognized Handwriting," at http://www.useit.com/papers/handwriting_retrieval.html (1993).
7. P. Jourlin, S. Johnson, K. Spark-Jones, and P. Woodland, "Improving Retrieval on Imperfect Speech Transcription," *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (August 1999), pp. 283–284.
8. J. Zobel and P. Dart, "Phonetic String Matching: Lessons from Information Retrieval," *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (August 1996), pp. 166–172.
9. S. Srinivasan and D. Petkovic, "Phonetic Confusion Matrix Based Spoken Document Retrieval," *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (July 2000), pp. 81–87.
10. W. Aref, D. Barbara, and P. Vallabhaneni, "The Handwritten Trie: Indexing Electronic Ink," *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data* (May 1995), pp. 151–162.
11. A. El-Nasan and G. Nagy, "Ink-Link," *Proceedings of the 15th International Conference on Pattern Recognition*, Vol. 2 (September 2000), pp. 573–576.
12. D. Lopresti and A. Tompkins, "On the Searchability of Electronic Ink," *Proceedings of the 6th International Workshop on the Frontiers of Handwriting Recognition* (August 1998).
13. D. Lopresti and G. Wilfong, "Crossdomain Searching Using Handwritten Queries," *Proceedings of the 7th International Workshop on the Frontiers of Handwriting Recognition* (September 2000).
14. D. Cooper, "How To Read Less and Know More: Approximate OCR for Thai," *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (July 1997), pp. 216–225.
15. T. Kwok, M. Perrone, and G. Russell, "Ink Retrieval from Handwritten Documents," *Proceedings of the 2nd Annual International Conference on Intelligent Data Engineering and Automated Learning*, Chinese University of Hong Kong (December 2000).
16. J. Subrahmonia, K. Nathan, and M. Perrone, "Writer Dependent Recognition of On-Line Unconstrained Handwriting," *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing* (May 1996), pp. 3478–3481.
17. E. Ratzlaff, "Inter-Line Distance Estimation and Text Line Extraction for Unconstrained Online Handwriting," *Proceedings of the 7th International Workshop on the Frontiers of Handwriting Recognition* (September 2000).
18. K. Spark-Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," *Journal of Documentation* **28**, 11–21 (1972).
19. S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," *Proceedings of the 3rd Text Retrieval Conference* (1995), pp. 109–126.
20. J. Ponte and W. Croft, "A Language Model Approach to Information Retrieval," *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (August 1998), pp. 275–281.
21. H. Turtle and W. Croft, "Efficient Probabilistic Inference for Text Retrieval," *Proceedings of the 3rd RIAO Conference Computer-Assisted Information Searching on Internet* (1991).

Accepted for publication April 2, 2002.

Michael P. Perrone *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: mpp@us.ibm.com).* Dr. Perrone is a research staff member in the Pen Technologies group. He joined IBM in 1994 and has worked on statistical machine learning algorithms for machine transcription of handwritten documents, as well as statistical methods for information retrieval. His interests include all aspects of statistical optimization and econometric modeling. He has received numerous invention and technical achievement awards and has contributed to award-winning IBM products. Dr. Perrone received a B.S. degree in both mathematics and physics from the Worcester Polytechnic Institute. He received Sc.M. and Ph.D. degrees in physics from Brown University.

Gregory F. Russell *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: gfr@us.ibm.com).* Dr. Russell is a research staff member at the T. J. Watson Research Center. He joined IBM in 1987, and has worked on pen-based computing systems, digitizer technology for handwriting capture, IR communications, optical subsystems, MPEG-2, and other I/O subsystems for ThinkPad®, as well as volumetric visualization and other graphics techniques. He participated in standards committees for IrDA-2 and most recently in developing a draft standard for "Ink-XML." He has received numerous invention and technical achievement awards, and he has contributed to award-winning products and concept prototypes. Dr. Russell received his bachelor's degree in engineering from Swarthmore College, and master's and Ph.D. degrees in mechanical and aerospace engineering from Princeton University.

Aiman Ziq *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (electronic mail: ziq@us.ibm.com).* Mr. Ziq holds a B.Sc. degree from the PSUCT university, Jordan. Before he joined IBM he participated in developing the architecture and wireless applications for cell phones and handheld devices. He joined IBM Research in 2001 and is currently working with the Pen Technologies group as a software engineer. His interests include autonomic computing and multimedia indexing and retrieval.