Document Retrieval with Unlimited Vocabulary

Viresh Ranjan[1] Gaurav Harit[2] C. V. Jawahar[1]
[1]CVIT, IIIT Hyderabad, India [2]IIT Jodhpur, India

Abstract

In this paper, we describe a classifier based retrieval scheme for efficiently and accurately retrieving relevant documents. We use SVM *classifiers for word retrieval, and argue that the classifier based solutions can be superior to the* OCR *based solutions in many practical situations. We overcome the practical limitations of the classifier based solution in terms of limited vocabulary support, and availability of training data. In order to overcome these limitations, we design a one-shot learning scheme for dynamically synthesizing classifiers. Given a set of* SVM *classifiers, we appropriately join them to create novel classifiers. This extends the classifier based retrieval paradigm to an unlimited number of classes (words) present in a language. We validate our method on multiple datasets, and compare it with popular alternatives like* OCR *and wordspotting. Even on a language like English, where* OCR*s have been fairly advanced, our method yields comparable or even superior results. Our results are significant since we do not use any language specific post-processing for obtaining this performance. For better accuracy of the retrieved list, we use query expansion. This also allows us to seamlessly adapt our solution to new fonts, styles and collections.*

1. Introduction

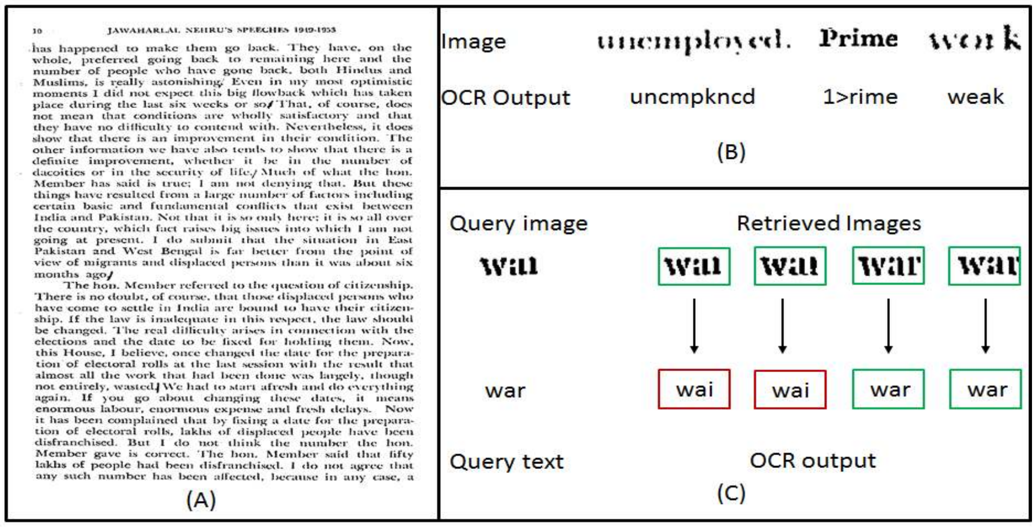
Retrieving relevant documents (pages, paragraphs or words) is a critical component in information retrieval solutions associated with digital libraries. Most of the present day digital libraries, use Optical Character Recognizers (OCRs) for the recognition of digitized documents, and thereafter employ a text-based solution for the information retrieval. Though OCRs have become the *de facto* preprocessing for the retrieval, they are realized as insufficient for degraded books [8], incompatible for older print styles [5], unavailable for specialized scripts [14] and very hard for handwritten documents [1]. Even for printed books, commercial OCRs may provide highly unacceptable results in practice. The best commercial OCRs can only give word accuracy of around 90% on printed books [18] in modern digital libraries. Recall of retrieval systems built on such text is thus limited.

In this paper, we hypothesize that word images, even if degraded, can be matched and retrieved effectively with a classifier based solution. A properly trained classifier can yield an accurate ranked list of words since the classifier looks at the word as a whole, and uses a larger context (say multiple examples) for the matching. We show, later in this paper, that a SVM based word retrieval can give mean average precision (mAP) as high as 1.0 even when the OCR based solution is limited to a mAP of 0.89. (See Figure 1 and Table 1.) Our results are significant since (i) We do not use any language specific post-processing for improving the accuracy. (ii) Even for a language like English, where OCRs are fairly advanced and engineering solutions were perfected, our simple classifier based solution is as good, if not superior to the best available commercial OCRs .

However, there are two fundamental challenges in using a classifier based solution for word retrieval (i)A classifier needs good amount of annotated training data (both positive and negative) for training. Obtaining annotated data for every word in every style is practically impossible. (ii) One could train a set of classifiers for a given set of frequent queries. However they are not applicable for rare queries.

In this paper we introduce a one-shot classifier learning scheme which enables direct design of a classifier for novel queries, without having any access to the annotated training data, i.e., classifiers are trained for a set of frequent queries, and seamlessly extended for the rare and arbitrary queries, as and when required. We refer to this as one-shot learning of query classifiers.

Recognition free retrieval was attempted in the past for printed as well as handwritten document collections [3, 5, 13]. Primary focus has been on designing appropriate features (eg. Profiles, SIFT-BoW), distance functions (eg. Euclidean, Earth Movers), matching schemes (eg. Dynamic Programing). Since most of these methods were designed for smaller collections (few handwritten documents as in [13]), computational time was not a major concern. Methods that extended this to larger collection [14, 15, 9]


10 JAWAHARLAL NEHRU'S SPEECHES 1949-1953

has happened to make them go back. They have, on the whole, preferred going back to remaining here and the number of people who have gone back, both Hindus and Muslims, is really astonishing/ Even in my most optimistic moments I did not expect this big flowback which has taken place during the last six weeks or so/ That, of course, does not mean that conditions are wholly satisfactory and that they have no difficulty to contend with. Nevertheless, it does show that there is an improvement in their condition. The other information we have also tends to show that there is a definite improvement, whether it be in the number of dacoities or in the security of life./ Much of what the hon. Member has said is true; I am not denying that. But these things have resulted from a large number of factors including certain basic and fundamental conflicts that exist between India and Pakistan. Not that it is so only here; it is so all over the country, which fact raises big issues into which I am not going at present. I do submit that the situation in East Pakistan and West Bengal is far better from the point of view of migrants and displaced persons than it was about six months ago/

The hon. Member referred to the question of citizenship. There is no doubt, of course, that those displaced persons who have come to settle in India are bound to have their citizenship. If the law is inadequate in this respect, the law should be changed. The real difficulty arises in connection with the elections and the date to be fixed for holding them. Now, this House, I believe, once changed the date for the preparation of electoral rolls at the last session with the result that almost all the work that had been done was largely, though not entirely, wasted/ We had to start afresh and do everything again. If you go about changing these dates, it means enormous labour, enormous expense and fresh delays. Now it has been complained that by fixing a date for the preparation of electoral rolls, lakhs of displaced people have been disfranchised. But I do not think the number the hon. Member gave is correct. The hon. Member said that fifty lakhs of people had been disfranchised. I do not agree that any such number has been affected, because in any case, a

Figure 1. (A) A typical page from a document image from our data set. (B) OCRs make many errors. Examples of word images and the errors from a commercial OCR in a page. (C) Examples of a classifier based retrieval compared with OCR retrieval. OCR did not recognize correctly the images whose OCR output are marked in red, and failed to recall. Note that the classifier based solution recalled all relevant images correctly, whereas the OCR recalled correctly 2 out of 4 images.

used mostly (approximate) nearest neighbor retrieval. For searching complex concepts in large databases, SVMs have emerged as the most popular and accurrate solution in the recent past [10]. For linear SVMs , both training and testing have become very fast with the introduction of efficient algorithms and excellent implementations.

Training has become efficient with methods like Pegasos [16] and whitening [7]. These methods make the offline training, and incremental/online training really fast. Training a classifier on the fly [4] is now considered as quite feasible for reasonably large image collections. (Indeed these are still not yet comparable to the indexing schemes popularly used in the text IR tasks, specially for huge data.)

The highlights of our current work are

1. We demonstrate that the SVM based word retrieval performance is superior to that of OCRs, and also the popular nearest neighbor based word spotting solutions.

2. We design a one-shot learning scheme, which allows to generate a novel classifier for rare/novel query words without any training data.

3. We demonstrate that with a simple retraining (with no extra supervision), the solution can adapt to a specific book or collection effectively.

4. We validate the performance on multiple books and demonstrate the qualitative and quantitative performance of the solution.

2. Accurate Classsifiers for Frequent and Rare Queries

Our word-level retrieval scheme is a direct application of the SVM classifier. We train a linear SVM classifier with few positive examples and a set of randomly sampled negative examples. During retrieval, this classifier is evaluated over the dataset images, and a ranked list of word images is predicted.

For representing word images, we prefer a fixed length sequence representation of the visual content, i.e., each word image is represented as a fixed length sequence of vertical strips (of varying width according to the aspect ratio of the word image). A set of features $\mathbf{f}_1, \ldots, \mathbf{f}_L$ are extracted, where $\mathbf{f}_i \in R^M$ is the feature representation of the i^{th} vertical strip and L is the number of vertical strips. For a SVM classifier, this can be considered as a single feature vector $\mathbf{F} \in R^d$ of size $d = LM$. However, we exploit the sequential nature of the feature representation for an on the fly synthesis of the novel classifiers in Section 2.2.1.

2.1. Efficient Classifier based Retrieval

Our classifier is basically a margin maximizing SVM classifier trained in a 1 vs rest setting. SVM gives maximum margin hyperplane separating the positive and negative instances. For a query word x_q, a SVM classifier $\mathbf{w}_q$ ($\mathbf{w}_q$ is the normal vector to the maximum margin hyperplane) is learned during the training, and for retrieval, database images are sorted based on the score $\mathbf{w}_q^T\mathbf{F}_i$. This evaluation is very efficient, since it requires only d multiplications and $d - 1$ additions. For frequent queries, this can in-fact be

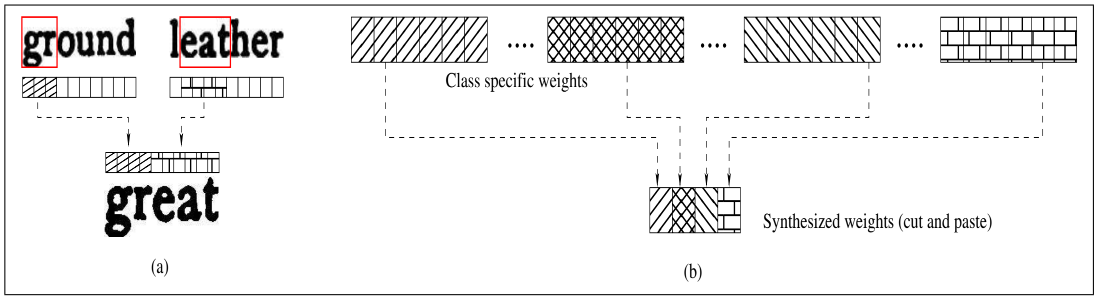


Figure 2. Synthesis of classifier for rare queries. (a) portions of the classifiers corresponding to "**gr**ound" and "l**eat**her" are joined to form a classifier for **great**. Note that the appropriate segments are automatically found. (b) In a general setting, a novel classifier gets formed from multiple constituent classifiers.

computed offline. However, traditional SVM implementations require many positive and negative examples to learn the weight vector $\mathbf{w}_q$.

In [10], Malisiewicz *et al.* proposed the idea of exemplar SVM (ESVM) where a separate SVM is learnt for each example. Almazan *et al.* [2] use ESVMs for retrieving word images. ESVMs are inherently highly tuned to its corresponding example. Given a query, it can retrieve highly similar word images. This constrains the recall, unless one has large variations of the query word available. Another demerit of ESVM is the large overall training time since a separate SVM needs to be trained for each exemplar. One approach to reduce training time is to make the negative example mining step offline and selecting a common set of negative examples [17]. Gharbi *et al.* [6] provide another alternative for fast training of exemplar SVM. As discussed in [6], a SVM between a single positive point and a set of negative points can be seen as finding the tangent to the manifold of images at the positive point. Assuming a Gaussian distribution over feature space, the authors give closed form expression for the normal vector to the Gaussian at query point x_q. The normal is given as $\Sigma^{-1}(x_q - \mu_0)$, where Σ is the global covariance matrix, μ_0 is global mean vector. This expression can also be interpreted in terms of linear discriminant classifiers(LDA) as done by Hariharan *et al.* in [7]. We also assume a Gaussian distribution over the feature space and hence, use this normal vector as an approximation of the SVM weight and use this weight vector for retrieval. Generalized expression for LDA weights are given as $w = \Sigma^{-1}(\mu_+ - \mu_-)$ where μ_+ and μ_- are the means of the positive and negative examples respectively. This simple computation makes the training extremely efficient. It requires only few d^2 multiplications for the design of specific query classifiers. Also, the same method can be used independent of whether we have one example query or multiple examples from query class.

2.2. Classifier design for rare queries

The number of possible words in a language can be infinite. It is not practical to build classifiers for all the possible words. However, on a closer look, we realize that all these words are composed from a very small number of characters, and a reasonably small set of *ngrams*. In many practical applications related to text processing, a finite set of *ngrams* were used to cover the vocabulary, and the small vocabulary solutions were extended to unlimited vocabulary settings. In this work, we show that the SVM classifiers corresponding to the *ngrams* can be effectively composed to generate novel classifiers on the fly. Fig 2 shows the overview of our classifier synthesis strategy. Such synthesized classifiers, by simply concatenating *ngram* classifiers, could be inferior to the directly built classifiers. (i) Due to the nature of scripts and writing styles, the joining will not be ideal. One should prefer larger grams for better synthesis. (ii) Classifiers for smaller *ngrams* could be noisy. Since the classifiers need to be built for all the words, the overall performance could thus be poor. We address these limitations as follows. It is well known that the queries in any search engine follow an exponentially decaying structure (Zipf's law) like the frequency of occurrence of words in a language. We build SVM classifiers (Section 3) for the most frequent queries and use classifier synthesis only for rare queries. This improves the overall performance. When the synthesized classifiers are not as good as the originl one, we further use query expansion (QE) for the refinement of the query classifier (See Section 3). Our method does not build artificial *ngram* classifiers; we use complete word classifiers and dynamically decide what portions from what words to be cut and pasted to create the novel classifier. We refer to this solution as a Direct Query Classifier (DQC) design scheme. In Section 2.2.1, we describe DP DQC, a dynamic programming based approach for DQC synthesis.

2.2.1 DP DQC: DQC **Design using Dynamic Programming**

Given a set of linear classifiers $\mathcal{W}_w = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$ for most frequent N queries and a query feature vector $\mathbf{x}_q$, we would like to synthesize a novel classifier $\mathbf{w}_q$ as a piecewise fusion of the parts from the available classifiers from $\mathcal{W}_w$ (See Figure 2). Let us assume that there are p portions that we need to select to form a novel classifier. Intervals (portions) are characterized by the sequence of indices a_1, ... a_{p+1} where $a_1 = 1$ and $a_{p+1} = L$. We formulate the classifier synthesis problem using a set of already available linear classifiers, as that of finding the optimal solution to

$$\max_{\{a_i\},\{c_i\}} \sum_{i=1}^{p} \sum_{k=a_i}^{a_{i+1}} {w_{c_i}^k}' x_q^k \tag{1}$$

where w_{c_i} corresponds to the c_i^{th} classifier that we choose and the inner summation applies the index range (a_i, a_{i+1}) to use a portion from the classifier c_i. The index i in the outer summation is over the portions, and p is the total number of portions we need to consider. The solution to the problem requires picking up the optimal set $\{c_i\}$ and the set of segment indices $\{a_i\}$ such that the $\{a_i\}$ form a monotonically increasing sequence of indices.

We use LDA as the linear classifier in our DQC solution. LDA weight $\mathbf{w}_q$ is given as

$$\mathbf{w}_q = \Sigma^{-1}(\mu_q - \mu_0) \tag{2}$$

where Σ and $\boldsymbol{\mu_0}$ are covariance and mean computed over the entire dataset of word images. Since Σ and μ_0 are common for all classes, synthesizing w_q requires finding mean vector (μ_q) for unknown query class. We consider the problem of finding μ_q for the (unknown) query class as the classifier synthesis problem outlined above. Let the set of mean vectors of frequent words be defined as $\mathcal{W}_\mu = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_N\}$. We divide query vector x_q into p fixed length portions and match each cut portion x_q^k to the most similar feature portion (say $\boldsymbol{\mu}_q^k$) of equal length from the set $\mathcal{W}_\mu$. We solve the problem of selecting the optimal set $\{c_i\}$ by solving for the optimal alignment of the query feature portion x_q^k with the best matching portion $\mu_{c_i}^k$ of the mean vector μ_{c_i} picked from $\mathcal{W}_\mu$. We then combine each of the obtained $\boldsymbol{\mu}_q^k$ to compose the mean vector $\boldsymbol{\mu}_q$ for the query. We ensure monotonicity of the sequence $\{a_i\}$ by using a fixed sequence for $\{a_i\}$, thus avoiding optimization over the set of indices $\{a_i\}$.

The alignment of feature portions is done using subsequence Dynamic Time Warping(DTW) [12], which is a dynamic programming (DP) algorithm. The DTW takes care of the variability in the different instances (word images) of the same class. The time complexity of subsequence DTW is $O(l_1\, l_2)$, where l_1 and l_2 are the length of the two sequences given as input. In our case, the l_1 is the length of each small segment and $l_2 = N \cdot d$ is the length of the concatenated sequence of all the mean vectors in the set $\mathcal{W}_\mu$. If we use all the known mean vectors in the set $\mathcal{W}_\mu$, synthesizing the classifier for a single query could take long time. To reduce the time complexity, we compute the normalized dot product between the query vector and all the mean vectors of the known classes. We select the 10 most similar mean vectors and use them in the subsequence DTW.

3. Efficient and Accurate Retrieval

When a direct classifier is used for the frequent words, retrieval is efficient since this requires only the evaluation of the classifiers. In practice, these are also pre-computed. For the rare words, we use the DQC classifier which requires a DP based selection from multiple composite classifiers. We call this DP based classifier synthesis strategy as DP DQC . This DP based strategy affects the efficiency and accuracy of the solution to some extent. We now discuss two refinements to the solution which can improve the efficiency and accuracy of the retrieval, NN DQC, which is an approximate nearest neighbor based implementation of DQC , and use of query expansion for adapting DQC to a previously unseen word collection without using any new training data.

NN DQC: DQC with Nearest Neighbor indexing DP DQC synthesis is slow because it uses DTW based alignment of the query vector portions with the mean vectors of the known classes. We can obtain a speed-up by using approximate nearest neighbor search instead of using DTW based alignment . This, in principle, compromises the optimality of the synthesis, however, in practice, it does not affect the quality of the classifier. We consider fixed portions (length R) of the mean vectors of all the known word classes and build an index over all such portions using FLANN [11].

The query $\mathbf{x}_q$ is also divided into portions of fixed length R and the approximate nearest neighbor match of each of these portions with the indexed portions is found using FLANN. The so obtained nearest neighbors are concatenated to give the mean vector $\boldsymbol{\mu}_q$, which is then used as in Equation (2), to compute the LDA weight $\mathbf{w}_q$. FLANN has a time complexity $O(RBD)$ when using hierarchical k-means for indexing, where B is the branching factor and D is the depth of the tree. This time complexity $O(RBD)$ is typically much smaller than $O(NRd)$ of subsequence DTW. Hence DQC using NN is much faster than using subsequence DTW.

Query expansion for DQC Classifiers which are trained on one dataset need not perform well on another dataset due to the print and style variations. For adapting the query to a new collection, query expansion (QE) is used. We imple-

Dataset				Retrieval						
Dataset	Source	Type	#images	#queries	OCR	NN	LDA	SVM	ESVM	NN DTW
D1	1 Book	Clean	26555	100	0.97	0.95	0.98	1.0	0.96	0.83
D2	2 Books	Clean	35730	100	0.95	0.72	0.92	0.98	0.72	0.62
D3	1 Book	Noisy	4373	100	0.89	0.73	0.98	1.0	0.86	0.71

Table 1. Table gives details of the datasets. It also shows comparison of various word retrieval schemes such as nearest neighbor (NN),LDA, SVM, EXEMPLAR SVM, DTW based NN with that of OCR based retrieval. Classifier based methods (specially SVM based methods are much superior to the OCR based solution. Performance is reported as mAP.

ment QE very efficiently. Query expansion is a concept used in information retrieval where the seed or primary query is reformulated to improve the retrieval performance. We use QE to further improve the performance of DQC . An index is built over all the database vectors and those vectors similar to query vector $\mathbf{x}_q$ are identified by performing approximate nearest neighbor search over the index. The top 5 vectors closest to the query vector are averaged to give the reformulated query vector to be used in the DQC. The reformulated query better captures the variations of the query class. We use FLANN for getting the approximate nearest neighbors, thus incurring an additional cost of $O(MBD)$ where M is the number of vertical strips in the word image.

This is much smaller compared to the time complexity $O(NRd)$ for subsequence DTW matching. Hence, adding QE step before DQC does not cause significant increase in computation time. However, it improves the accuracy.

4. Experiments, Results and Discussions

In this section, we validate the DQC classifier synthesis method on multiple word image collections and also demonstrate its quantitative and qualitative performance.

Data Sets, Implementation and Evaluation Protocol Our datasets, detailed in Table 1, comprises scanned English books from a digital library collection. We manually created ground truth at word level for the quantitative evaluation of the methods. Note that the words are segmented using the ground truth information. The first collection (D1) of words is from a book which is reasonably clean. On this collection, commercial OCR (ABBYY Fine Reader 9.0) provides very high word accuracy.

We use this collection to demonstrate that our method provides satisfactory performance on collections where the OCR is satisfactory. Second dataset (D2) is larger in size and is used to demonstrate the performance in case of heterogeneous print styles. Third data set (D3) is a noisy book, and is used to demonstrate the utility of the performance of our method in degraded collections. For the experiments, we extract profile features [13] for each of the word images. Profile features comprise of the following: (i) Vertical projection profile, which counts the number of ink pixels in each column (ii) Upper and lower word profile, which encode the distance between the top (lower) boundary and the top-most (lower-most) ink pixels in each column. (iii) Background/Ink transition counts the number of background to ink transitions in each column. All these features are extracted for 100 vertical strips. This results in a 400 dimensional representation for every word image. The features are normalized to $[0, 1]$ so as to avoid dominance of any specific feature. Instead of the query log of a search engine, which lists the frequent queries and rare queries, we have considered the frequency of occurrence of the words in the collection. We report the mAP score for 100 frequent queries as the retrieval performance measure in Table 1. In Table 1, we compare the performance of various methods as mAP of the retrieval. The OCR scores correspond to average over all the queries in the corresponding dataset, and are obtained by using ABBYY Fine Reader 9.0. We observed that ABBYY 9.0 performs comparably with its more recent versions . Some of the salient observations out of this experiment are (i) OCR performance is inferior to the SVM based retrieval in all the cases. (ii) Faster approximation in the form of LDA with multiple positive examples is quite close to the performance of SVM . (iii) ESVM performs inferior due to the use of only one example (iv) Nearest neighbor methods (with DTW, Euclidean etc.) are inferior, in general, to SVM . (v) Nearest neighbour with DTW may be considered as equivalent to word spotting. Therefore, one could observe that this classifier based retrieval is superior to OCR and the DTW based retrieval.

Note that DTW cannot be directly used in the SVM classifier since the corresponding kernel will not be positive semidefinite, and moreover due to the computational complexity. Fig 1 depicts some of the qualitative examples of the retrieval. As can be seen that the classifier-based retrieved images are more relevant to the query. Classifiers are less sensitive to the degradations (eg. cuts, merges etc.) present in the word images. Figure 3 gives more qualitative examples of the retrieval. We show examples of classifier based retrieval, and DQC based retrieval. We also show that the quality of the retrieval improves with QE.

Performance of the DQC In the initial experiment, we built classifiers for most frequent 1000 word categories. For

Dataset	queries	Freq	Freq(QE)	Rare		Rare(QE)		Overall	
				DP	NN	DP	NN	DP	NN
D1	100	0.99	0.99	0.82	0.85	0.91	0.90	0.98	0.98
D2	100	0.98	0.98	0.84	0.82	0.87	0.87	0.97	0.97
D3	100	1	1	0.71	0.73	0.80	0.82	0.98	0.98

Table 2. Retrieval performance: mAP scores for the evaluation of frequent and rare queries. For rare queries, mAP scores are given for DP DQC as well as NN DQC. Notice that QE improves the performance significantly.

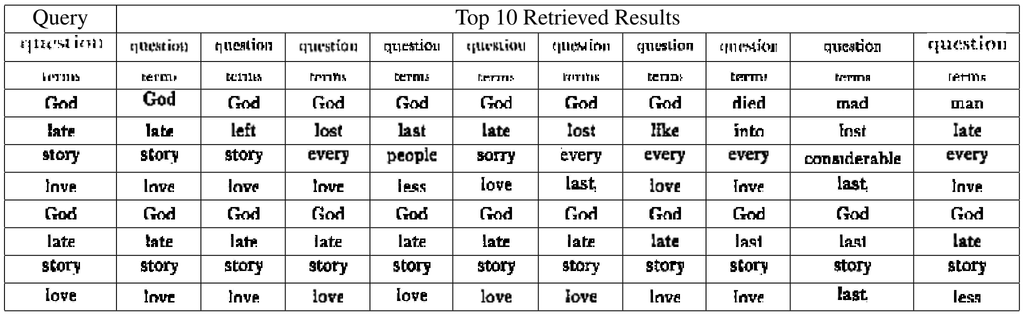

Query
Top 10 Retrieved Results

Figure 3. Figure shows few query words and corresponding top 10 retrieved results. First two rows show frequent query results. Row 3 - Row 6 show rare query results using DP DQC . Row 7 - Row 10 show rare query results using DP DQC with QE.

C	D1		D2		D3	
	mAP		mAP		mAP	
	DP	NN	DP	NN	DP	NN
1	0.80	0.79	0.77	0.75	0.61	0.61
10	0.82	0.82	0.77	0.78	0.62	0.65
20	0.82	0.84	0.81	0.79	0.61	0.66
30	0.81	0.82	0.75	0.78	0.60	0.65

Table 3. Table shows change in retrieval performance with change in cut length. mAP values are given for both DP DQC and NN DQC. C is the cut-length.

the rest of the words, we find that a DQC based solution is appropriate. We implement the DQC based solution as discussed in the previous section. During the DQC evaluation, we discard the trained classifiers and mean vectors for the chosen query word classes. DQC synthesizes the mean vector for the query, to be used to compute the LDA classifier. The DQC identifies the 10 most similar mean vectors (of the remaining 900 word classes) using the normalized dot product value. Subsequence DTW is used to find the best alignment of the cut-portions of the query feature vector with the concatenated mean vectors of the closest 10 word classes.

We observe that the DQC performs somewhat inferior to the true classifiers designed with multiple examples. This is partly due to the joining process associated. In our implementation, we simply concatenated the segments coming from multiple classifiers. We consider 100 frequent and 100 rare queries for the evaluation and show the results in Table 2. One can notice that the method is superior for frequent queries. In both the cases, one can see that the query expansion improves the performance. Improvement is significant in the case of rare queries. When the vocabulary covers 90% of the language, (which is relatively small for languages like English), the overall AP can be estimated as a weighed combination of these two mAPs. Please note that these are pure estimates and the actual measurement can be done only with a reasonably large query log. However, it can be seen that the performance of the system is quite competitive and also outperforms the OCR in many situations.

In Table 3, we report the effect of variation of cut length on retrieval performance. For smaller cut lengths, mAP is relatively less. We can also observe that performance degrades for larger cut-length of 30. This happens because occurrence of desired large portions(say bigram) might be less frequent than the smaller constituent portions(say unigram).

In Fig 4, we show few page level retrieval results for single word query and multiple words query. To observe how DQC performs in presence of noise, we introduce noisy versions of few query words into the dataset and perform DQC retrieval over these query words. We degrade the im-

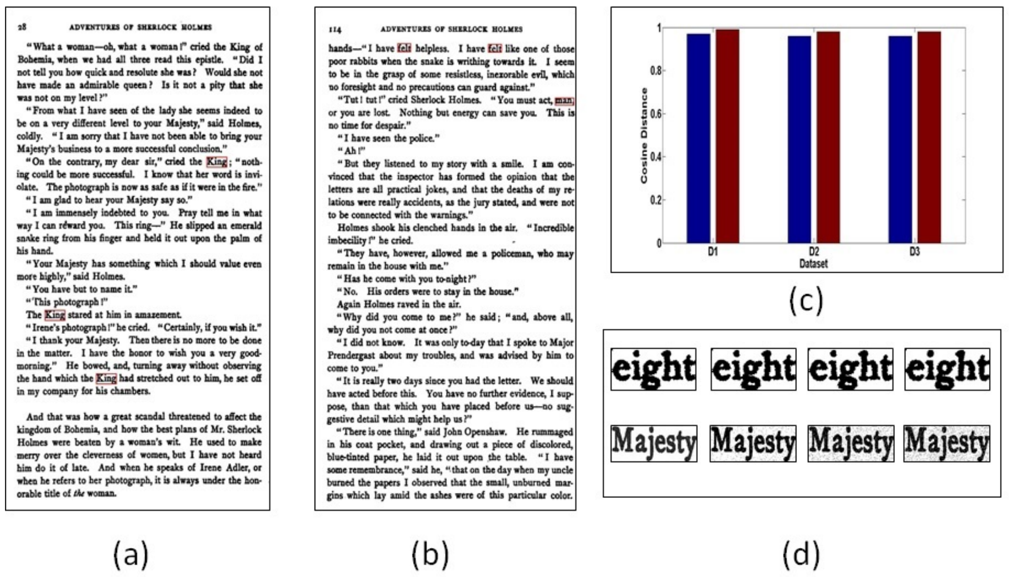


Figure 4. Fig(a) shows page retrieval for query "king". Fig(b) shows page retrieval with multiple query words. Here query words are "man" and "felt". Fig(c)shows effect of query expansion on weights synthesized for 100 queries each from the three datasets. Blue bars are for DQC weights and Red bars are for DQC+QE weights. Fig(d) shows few example images over which noise was added artificially and query was performed.

ages by adding Gaussian noise and by randomly performing flips for edge pixels of the words(flipping is done by randomly changing some of the foreground edge pixels to background pixels and vice-versa). We observe mAP of 0.81 for these noisy queries. Few sample images from the noisy set are shown in Fig4(d). To measure the goodness of the synthesized classifier, we compare it with the actually trained LDA classifier already available with us for the rare words we used. The effect of QE on the goodness of the synthesized classifier has been shown in Fig 4(c). Table 4 shows the averaged cosine distance and root mean square error(RMSE) between the (DQC and trained) classifier weight vectors over the 100 queries. The cosine distance between actual weight w_q and synthesized weight $\hat{w}_q$ can be given as $\cos\theta = \frac{w_q^\top \hat{w}_q}{\|w_q\|\|\hat{w}_q\|}$. RMSE is the averaged root means square error between the weight vectors. It can be calculated by the formula $RMS = \sqrt{\frac{\sum_{i=1}^{d}(w_i - \hat{w}_i)^2}{d}}$.

Dataset	# queries	Cosine Distance	RMSE
D1	100	0.97	0.11
D2	100	0.95	0.15
D3	100	0.96	0.11

Table 4. Table gives comparison between DQC weights and actual weights for 100 rare queries. Cosine distance and RMSE are used for comparison between the two.

We also compare effect of applying QE with DQC on all three datasets. We give corresponding results in Figure 5. We observe that in all three cases, QE further improves performance of DQC .

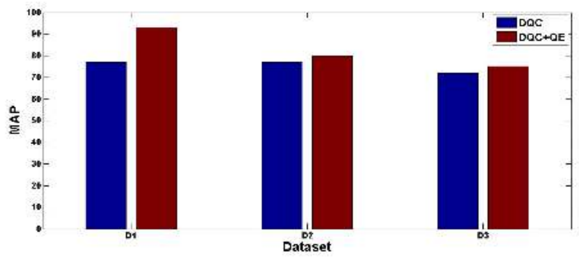


Figure 5. Barplot shows retrieval performance of DQC for rare queries. Improvement in mAP values with QE can be observed for all three datasets.

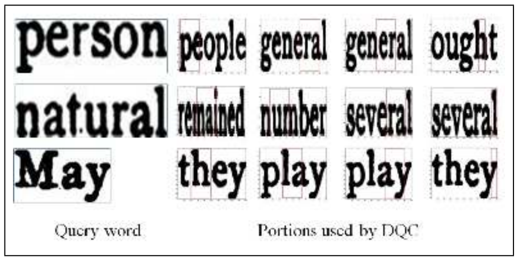


Figure 6. Figure shows few query images and corresponding portions used in generating DP DQC classifier.

In Fig 6, we show a few queries and the corresponding portions used in synthesizing DQC .It can be seen that visually similar portions corresponding to portions of the query word are utilized while synthesizing DQC . We also compare the average retrieval time for frequent and rare queries. For 100 frequent queries, average query time is $2.4ms$. For 100 rare queries, DP DQC takes an average of $140ms$ whereas NN DQC takes $26ms$.

Discussions Our system can perform page retrieval and support multiple keyword queries. Page retrieval is performed based on the score given by query weight to different word images present in the page. Location of word in page can be marked using ground truth information. To deal with multiple query keywords, weights corresponding to all the keywords are obtained and multiple resulting lists are combined based on the score of the word images. Images showing page retrieval and multiple keyword query are given in Figure 4. To facilitate faster retrieval for frequent queries, we evaluate SVM weights of frequent queries over all the word images and store the resulting index. Hence, classifiers need to be evaluated only for the rare queries.

5. Conclusion

In this paper, we have described a classifier based retrieval scheme for effectively retrieving word and document images from a collection. We argue that the classifier based method is superior to the OCR in practice. We introduce a novel classifier synthesis scheme which enable design of classifiers without any explicit training data. For this we exploit the fact that words in a language can be formed from fewer combination of character sequences. A major disadvantage of the classifier based scheme is the difficulty in indexing, which is important if the method needs to scale to millions of document images. One of our future work is to design efficient and scalable retrieval system which uses linear SVM classifiers in the back end.

References

[1] J. Almazán, D. Fernández, A. Fornés, J. Lladós, and E. Valveny. A coarse-to-fine approach for handwritten word spotting in large scale historical documents collection. In *Proc. ICFHR*, 2012.

[2] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Segmentation-free word spotting with exemplar svms. *Pattern Recognition*, 2014.

[3] H. Cao and V. Govindaraju. Vector Model Based Indexing and Retrieval of Handwritten Medical Forms. In *Proc. ICDAR*, 2007.

[4] K. Chatfield and A. Zisserman. Visor: Towards on-the-fly large-scale object category retrieval. In *Proc. ACCV*, 2012.

[5] B. Gatos, N. Stamatopoulos, and G. Louloudis. ICDAR2009 handwriting segmentation contest. *IJDAR*, 2011.

[6] M. Gharbi, T. Malisiewicz, S. Paris, and F. Durand. A Gaussian Approximation of Feature Space for Fast Image Similarity. Mit techinical report, 2012.

[7] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Computer Vision–ECCV 2012*, pages 459–472. Springer, 2012.

[8] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *IJDAR*, 2007.

[9] A. Kovalchuk, L. Wolf, and N. Dershowitz. A simple and fast word spotting method. *Frontiers in Handwriting Recognition (ICFHR)*, 2014.

[10] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *Proc. ICCV*, 2011.

[11] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. VISSAPP*, 2009.

[12] M. Müller. *Information retrieval for music and motion*. Springer, Heidelberg, 2007.

[13] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Proc. CVPR*, 2003.

[14] K. P. Sankar and C. V. Jawahar. Probabilistic reverse annotation for large scale image retrieval. In *Proc. CVPR*, 2007.

[15] K. P. Sankar, C. V. Jawahar, and R. Manmatha. Nearest neighbor based collection ocr. 2010.

[16] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proc. ICML*, 2007.

[17] M. Takami, P. Bell, and B. Ommer. Offline learning of prototypical negatives for efficient online exemplar svm. In *Applications of Computer Vision (WACV), IEEE Winter Conference on*. IEEE, 2014.

[18] I. Z. Yalniz and R. Manmatha. An efficient framework for searching text in noisy document images. In *Proc. DAS*. IEEE, 2012.