Automatically Identifying Join Candidates in the Cairo Genizah

Lior Wolf, Rotem Littman, Naama Mayer, Nachum Dershowitz
The Blavatnik School of Computer Science
Tel Aviv University

Roni Shweka, Yaacov Choueka
The Friedberg Genizah Project
Jerusalem, Israel

Abstract

A join is a set of manuscript-fragments that are known to originate from the same original work. The Cairo Genizah is a collection containing approximately 250,000 fragments of mainly Jewish texts discovered in the late 19th century. The fragments are today spread out in libraries and private collections worldwide, and there is an onging effort to document and catalogue all extant fragments.

The task of finding joins is currently conducted manually by experts, and presumably only a small fraction of the existing joins have been discovered. In this work, we study the problem of automatically finding candidate joins, so as to streamline the task. The proposed method is based on a combination of local descriptors and learning techniques.

To evaluate the performance of various join-finding methods, without relying on the availability of human experts, we construct a benchmark dataset that is modeled on the Labeled Faces in the Wild benchmark for face recognition. Using this benchmark, we evaluate several alternative image representations and learning techniques. Finally, a set of newly-discovered join-candidates have been identified using our method and validated by a human expert.

1. Introduction

Written text is one of the best sources for understanding historical life. The Cairo Genizah is a unique source of preserved middle-eastern texts, collected between the 11th and the 19th centuries. These texts are a mix of religious Jewish manuscripts with a smaller proportion of secular texts. To make the study of the Genizah more efficient, there is an acute demand to group the fragments and reconstruct the original manuscripts. Throughout the years, scholars have devoted a great deal of time to manually identify such groups, referred to as *joins*, often visiting numerous libraries.

Manual classification is currently the gold-standard for finding joins. However, it is not scalable and cannot be applied to the entire corpus. We suggest automatically identifying candidate joins to be verified by human experts. To this end, we employ modern image-recognition tools such as local descriptors, bag-of-features representations and discriminative metric learning techniques. These techniques are modified for the problem at hand by employing suitable preprocessing and by employing task-specific key-point selection techniques. Where appropriate, we use suitable generic methods.

(a) (b)

Figure 1. Example of a document from the Cairo Genizah. (a) The original fragment. (b) After the binarization process.

We validate our methods in two ways. The first is to construct a benchmark for the evaluation of algorithms that are able to compare the images of two leaves. Algorithms are evaluated based on their ability to determine whether two leaves are a join or not. In addition, we create a short list of newly discovered join-candidates that are the most likely, according to our algorithm's metric, and send it to a human expert for validation.

The main contributions of this work are as follows:

1. The design of an algorithmic framework for finding join-candidates. The algorithms are based on the application of local descriptors and machine learning techniques. The framework provides a high-throughput method for join finding in which human expertise is

utilized efficiently.

2. The study of suitable algorithmic details for obtaining high levels of performance for finding candidate joins. In particular, by carefully constructing our recognition method, we obtain an increase in recognition rate, at very low false-positive rates, of up to ten-fold.
3. Provide a benchmark for the evaluation of join-finding algorithms. Such a benchmark is important for evaluating such algorithms in the absence of accessible human experts.
4. The actual identification of new, unknown, joins in the Genizah corpus.

2. Related work

Genizah research Discovered in 1896 in the attic of a synagogue in the old quarter of Cairo, the Genizah is a large collection of discarded codices, scrolls, and documents, written mainly in the 10th to 15th centuries. The attic was emptied and its contents have found their way to over fifty libraries and collections around the world. The documents, with few exceptions, are of paper and vellum, and the texts are written mainly in Hebrew, Aramaic, and Judeo-Arabic (in Hebrew characters), but also in many other languages (including Arabic, Judeo-Spanish, Coptic, Ethiopic, and even one in Chinese). The finds included fragments of lost works (such as the Hebrew original of the apocryphal Book of Ecclesiasticus), fragments of hitherto unknown works (such as the Damascas Document, later found among the Qumran scrolls), and autographs by famous personages, including the Andalusians, Yehuda Halevi (1075–1141) and Maimonides (1138–1204).

Genizah documents have had an enormous impact on 20th century scholarship in a multitude of fields, including Bible, rabbinics, liturgy, history, and philology. Genizah research has, for example, transformed our understanding of medieval Mediterranean society and commerce, as evidenced by S. D. Goiten's monumental five-volume work, *A Mediterranean Society*. See [18] for the history of the Genizah and of Genizah research. Most of the material recovered from the Cairo Genizah has been microfilmed and catalogued in the intervening years, but the photographs are of mediocre quality and the data incomplete (thousands of fragments are still not listed in published catalogues).

The philanthropically-funded Friedberg Genizah Project is in the midst of a multi-year process of digitally photographing (in full color, at 600dpi) most—if not all—of the extant manuscripts. The entire collections of the Jewish Theological Seminary in New York (ENA), the Alliance Israélite Universelle in Paris (AIU), the recently rediscovered collection in Geneva, and many smaller collections have already been digitized, and comprise about 90,000 images (recto and verso of each fragment). The digital preservation of another 140,000 fragments at the Taylor-Schechter Genizah Collection at Cambridge is now underway. The images are being made available to researchers online at www.genizah.org.

Unfortunately, most of the leaves that were found were not found bound together. Worse, many are fragmentary, whether torn or otherwise mutilated. Pages and fragments from the same work (book, collection, letter, etc.) may have found their way to disparate collections around the world. Some fragments are very difficult to read, as the ink has faded or the page discolored. Accordingly, scholars have expended a great deal of time and effort on manually rejoining leaves of the same original book or pamphlet, and on piecing together smaller fragments, usually as part of their research in a particular topic or literary work. Despite the several thousands of such joins that have been identified by researchers, very much more remains to be done [14].

Writer identification A related task to that of join finding is the task of writer identification, in which the goal is to identify the writer by morphological characteristics of a writer's handwriting. Since historical documents are often incomplete and noisy, preprocessing is often applied to separate the background and to remove noise (see, for instance, [3, 13]). Latin letters are typically connected, unlike Hebrew ones which are usually only sporadically connected, and efforts were also expended on designing segmentation algorithms to disconnect letters and facilitate identification. See [5] for a survey of the subject. The identification itself is done either by means of local features or by global statistics. Most recent approaches are of the first type and identify the writer using letter- or grapheme-based methods, which use textual feature matching [16, 2]. The work of [3] uses text independent statistical features, and [4, 7] combine both local and global statistics.

Interestingly, there is a specialization to individual languages, employing language-specific letter structure and morphological characteristics [4, 16, 7]. In our work, we rely on the separation of Hebrew characters by employing a keypoint detection method that relies on connected components in the thresholded images.

Most of the abovementioned works identify the writer of the document from a list of known authors. Here, we focus on finding join candidates, and do not assume a labeled training set for each join. Note, however, that the techniques we use are not entirely suitable for distinguishing between different works of the same writer. Still, since writers are usually unknown (in the absence of a colophon or signatures), and since joins are the common way to catalog Genizah documents, we focus on this task. Additional data such as text or topic identification, page size and number of lines can be used to help distinguish different works of the same writer.

The LFW benchmark To provide a clean computational framework for the identification of joins, we focus

in our evaluation on the problem of image pair-matching (same/not-same), and not, for example, on the multiclass classification problem. Specifically, given images of two Genizah leaves, our goal is to answer the following simple question: are these two leaves part of the same original work or not? Previous studies have shown that improvements obtained on the pair-matching problem carry over to other recognition tasks [22] .

The benchmark we constructed to evaluate our methods is modeled after the recent Labeled Faces in the Wild (LFW) face image data set [10]. The LFW benchmark has been successful in attracting researchers to improve face recognition in unconstrained images, and the results show a gradual improvement over time [9, 22, 17].

3. Methods

Each leaf in the Genizah may be torn into several fragments and is represented by two or more images depicting the two sides and possibly multiple images of the same side. The join identification technique follows the following pipeline. First, the leaf images are preprocessed so that each image is segmented into fragments, and each fragment is binarized and aligned horizontally by rows. Next we detect keypoints in the images, and calculate a descriptor for each keypoint. All descriptors from the same leaf are combined, and each leaf is then represented by a single vector. The vectorization is done by employing a dictionary which is computed offline beforehand. Finally, every pair of vectors (corresponding to two leaves) are compared by means of a similarity score. We employ both simple and learned similarity scores and, in addition, combine several scores together by employing a technique called stacking.

3.1. Preprocessing

The images we employ were obtained from the Friedberg Genizah Project (`www.genizah.org`) and are given as 300–600 dpi JPEGs, of arbitrarily aligned documents placed on different backgrounds. Furthermore, the images contain superfluous parts, such as paper tags, rulers, color tables, etc. An example, which is relatively artifact free, is shown in Figure 1(a). The written side of each fragment is manually identified and, and the images are manually aligned by rotating them by a multiple of $90°$. Then, preprocessing is applied to separate the fragment from the rest of the image, and to provide an accurate alignment according to the direction of the lines.

Foreground segmentation The process of separating the fragments from the background depends on the way the image was captured. In this work we employ images from the AIU and ENA collections. The images of the AIU collection were taken on a distinct cyan graph paper background, and the per-pixel segmentation is performed by an SVM classifier that inspects the RGB values of each pixel.

The ENA collection is more challenging, since backgrounds vary and include items such as gray graph paper or blank beige background. Moreover, some of the documents are in plastic sleeves, and some have a black folder stripe on the side. Per-pixel segmentation is done by thresholding in the saturation domain.

To create a region-based segmentation of the fragments, and to improve segmentation, we mark the connected components of the detected foreground pixels, and—for ENA images—we calculated the convex hull of each component. Those steps retain almost all of the relevant parts of the images, while excluding most of the background.

Detection and removal of connected rulers Labels, ruler, color swatches and any other non-relevant component that fall in separated regions are manually removed. In some images, especially large documents, a ruler is adjacent to the actual fragments and is not separated by the region-segmentation process. Since we know the type of ruler used, we located them by a detector based on applying SIFT [15] and RANSAC [8]. Then, the containing region is segmented by color and removed.

Binarization The regions detected in the foreground segmentation process are then binarized using the autobinarization tool of the ImageXpress 9.0 package by Accusoft Pegasus. To cope with failures of the Pegasus binarization, we also binarized the images using the local threshold set at 0.9 of the local average of the 50×50 patch around each pixel. The final binarization is the pixel-wise AND of the two binarization techniques. Pixels nearby the fragment boundary are set to zero. An example result is shown in Figure 1(b).

Auto-alignment Each region is rotated so the rows (lines of text) are in the horizontal direction. This is done using a simple method, which is similar to [1, 20]. We compute an alignment score s for each rotation angle from $-45°$ to $45°$, based on the projection of all the binary pixels onto the vertical axis; that is, we inspect the sum of all pixels along horizontal lines. We then identify the local peaks and valleys at a defined window size, and calculate the score:

$$s = \frac{1}{N} \sum_{n=1}^{N} \left(\frac{y_h^{(n)} - y_l^{(n)}}{h^{(n)}} \right)$$

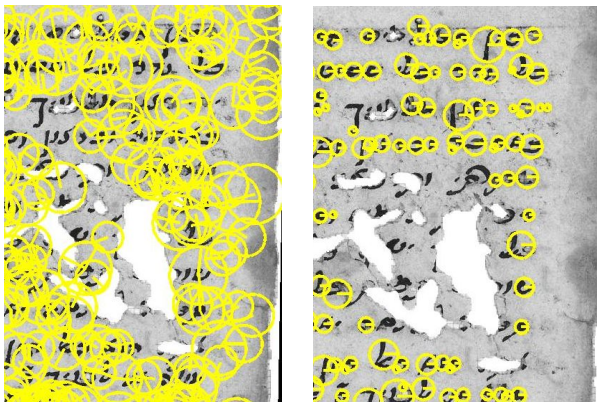
where N is the number of peaks found in the profile, $y_h^{(n)}$is the value of the nth peak, $y_l^{(n)}$, the value of the highest valley around the nth peak, and $h^{(n)}$ is a normalization by the height of the binary mask of the document at that point. This last normalization is applied since the documents are not guaranteed to be rectangular or to have the same width at each position. The document is rotated by the angle that gives the highest score.

3.2. Keypoint detection

We employ a bag-of-features based method, in which the signature of the leaf is based on descriptors collected from local patches in its fragments, centered around keypoints. The simplest method for selecting keypoints, which is often effective in general object recognition tasks [12], is to select keypoints on a grid. Grid points are only considered if they belong to the foreground. Another popular method is to employ the Difference of Gaussian (DoG) operator used by the SIFT keypoint detector [15]. In our experiments with the SIFT detector, we rely on the natural scale and direction of the detector. We have experimented with a few thresholds for the peak threshold parameter, finally selecting 0.005.

A third method for keypoint detection uses the fact that, in Hebrew writing, letters are usually separated. We start by calculating the connected components (CC) of the binarized images. To filter out fragmented letter parts and fragments arising from stains and border artifacts, we compare the size of the CC to the height of the lines which is estimated similarly to the alignment stage above. The scale of each detected keypoint is taken as the maximum dimension of the associated CC.

The CC method has the advantage of using the actual letters of the document, however, it is dependent on correct alignment of fragments (some have multiple line directions) and deals poorly with connected letters. Figure 2 shows the keypoints found using the SIFT and CC detectors.



(a) (b)

Figure 2. Keypoint detection methods. (a) using the DoG operator[15]. (b) using the proposed CC method.

3.3. Local descriptors

Each keypoint is described by a descriptor vector. We used the following descriptors: SIFT, PCA-SIFT, binary aligned patch, and binary vertically aligned patch. SIFT [15] and PCA-SIFT [11] are popular descriptors, which encode histograms of gradients in the image. Figure 3 illustrates the application of SIFT to one fragment.

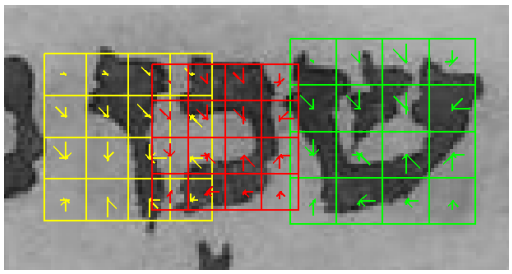


Figure 3. SIFT descriptors on three nearby detected keypoints

In the binary aligned patch representation, the patch composed of the pixels of the detected keypoint is first stretched to a fixed size of 32×32 pixels, and then the image values are recorded. A somewhat more effective way is to stretch the patch at the same scale on both axes until the height becomes 32 pixels and then crop or zero-pad the resulting patch to a width of 64 pixels.

3.4. Dictionary creation and vectorization

Bag-of-feature techniques [6] rely on a dictionary that contains a representative selection of descriptors obtained on various interest points. To this end, we set aside a small dataset of 150 documents. We detect keypoints in those documents, by the appropriate method for each experiment, and subsample a large collection of $20,000$ descriptors. These are then clustered by the k-means algorithm to obtain a dictionary of varying sizes. Given a dictionary, we employ both histogram-based and distance-based methods to encode each leaf as a vector.

In histogram type vectorization methods [6], one counts, for each cluster-center in the dictionary, the number of leaf descriptors (in the encoded image) closest to it. The result is a histogram of the descriptors in the encoded leaf with as many bins as the size of the dictionary. We have experimented with two types of multiplicative normalization. In the first type, we calculate the L2 norm of the resulting vector and divide it by this norm. In the second one, we do the same, by fixing the L1 norm to be 1. While the latter may seem better motivated, it is the former that performs better.

Distance-based representation techniques [19] are based on computing the minimum distance to all descriptors of the given leaf, for each cluster center in the dictionary. We employ two versions. In the first, the distances are used, and in the second we convert distances to similarities by taking the exponent of the distance times -0.001.

3.5. Similarity computation

For every pair of leafs, we need to determine whether they are from the same join or not. This is done by computing a similarity score or combining several similarity scores. The basic scores we use are the L2, L1 and Hellinger norms. The latter is simply the L2 norm applied to the square root of each element in the vector. This norm, similar to the χ^2

norm, is effective for L1 normalized histograms.

We also employ learned similarities. There are many metric learning algorithms; however, not all of them are suitable for training on pairs. We have experimented with two types of metric learning that have been shown to be successful in the LFW benchmark—one is SVM based, and the other is the LDA-based One Shot Similarity score (OSS).

SVM of vector of absolute differences In this technique, which was shown to be effective on the LFW dataset [17], one simply trains an SVM classifier on the vector of absolute differences between the two vectors of every training pair (recall that the training pairs are labeled as positive or negative). Given a new pair, the absolute differences are computed at every coordinate and the trained SVM is applied to the resulting vector. The signed distance from the separating hyperplane is the reported similarity. Higher values indicate better matching leafs.

One Shot Similarity The OSS [22] is a similarity learning technique designed for the same/not-same problem. Given two vectors $\mathbf{p}$ and $\mathbf{q}$ their OSS score is computed by considering a training set of background sample vectors $\mathbf{A}$. This set of vectors contains examples of items different from either $\mathbf{p}$ and $\mathbf{q}$ (that is, they do not belong in the same class as neither $\mathbf{p}$ nor $\mathbf{q}$). Note, however, that these training samples are otherwise unlabeled. In our experiments, we take the set $\mathbf{A}$ to be one split out of the nine splits used for training at each iteration (see Section 4).

A measure of the similarity of $\mathbf{p}$ and $\mathbf{q}$ is then obtained as follows. First, a discriminative model is learned with $\mathbf{p}$ as a single positive example, and $\mathbf{A}$ as a set of negative examples. This model is then used to classify the second vector, $\mathbf{q}$, and obtain a classification score. The nature of this score depends on the particular classifier used. We, following [22], employ an LDA classifier, and the score is the signed distance of $\mathbf{q}$ from the decision boundary learned using $\mathbf{p}$ (positive example) and $\mathbf{A}$ (negative examples). A second such score is then obtained by repeating the same process with the roles of $\mathbf{p}$ and $\mathbf{q}$ switched: this time, a model learned with $\mathbf{q}$ as the positive example is used to classify $\mathbf{p}$, thus obtaining a second classification score. The final OSS is the sum of these two scores.

3.6. Classification and combination of features

For recognition, we need to convert the similarity values of Section 3.5 to a decision value. Moreover, it is beneficial to combine several similarities together. For both these tasks we employ linear SVM (fixed parameter value $C = 1$), as was done in [22, 21]. In the case of one-similarity, the similarity is fed to the SVM as a 1d vector and training is performed on all training examples. In this case the SVM just scales the similarities and determines a threshold for classification.

In order to combine several similarities together we use the SVM output (signed distance from hyperplane) obtained from each similarity separately and construct a vector. This vector is then fed to another SVM. The value output by the last classifier is our final classification score. This method of combining classifier output is called stacking [23]. When employing it, care should be taken so that no testing example is used during training. Specifically, the learned similarities above (SVM-based and OSS) need to be computed multiple times.

4. The newly proposed Genizah benchmark

Our benchmark, which is modeled after the LFW face recognition benchmark [10], consists of 1944 leafs, all from the New York and Paris collections. There are several differences vis-à-vis the LFW benchmark. First, in the LFW benchmark the number of positive pairs (images of the same person) and the number of negative pairs are equal. In our benchmark, this is not the case, since the number of known joins is rather limited. Second, while in the LFW benchmark, a negative pair is a pair that is known to be negative, in our case a negative pair is a pair that is not known to be positive. This should not pose a major problem, since the expected number of unknown joins is very limited in comparison to the total number of pairs.

There are two views of the dataset: View 1, which is meant for parameter tuning, and View 2, meant for reporting results. View 1 contains three splits, each containing 300 positive pairs of leaves belonging each to the same join, and 1200 negative pairs of leaves that are not known to belong to the same join. When working on View 1, one trains on two splits and tests on the third.

View 2 of the benchmark consists of ten equally sized sets. Each contains 196 positive pairs of images taken from the same joins, and 784 negative pairs. Care is taken such that no known join appears in more than one set.

To report results on View 2, one repeats the classification process 10 times. In each iteration, nine sets are taken as training, and the results are evaluated on the 10th set. Results are reported by constructing an ROC curve for all splits together (the outcome value for each pair is computed when this pair is a testing pair), by computing statistics of the ROC curve (area under curve, equal error rate, and true positive rate at a certain low false positive rate) and by recording average recognition rate for the 10 splits.

5. Results obtained on the new benchmark

In order to determine the best methods and settings for join identification we have experimented with the various aspects of the algorithm. When varying one aspect, we fixed the others to the following default values: the connected component method for keypoint selection algorithm, the SIFT descriptor, a dictionary size of 500, L2 normalized histogram for vectorization, and SVM applied to absolute difference between vectors as the similarity measure.

Results for the parametric methods (keypoint detection method, descriptor type and parameters and dictionary size), were compared on View 1. Results for the various norms and vectorization methods were compared on View 2, since they do not require fitting of parameters.

Figure 4(a) compares the performance of the various keypoint detectors. For each of the detector types, a new dictionary was created and the entire pipeline was repeated. The presented results, which are obtained for the best parameters of each of the three methods, demonstrate that the proposed CC based keypoint detector does better than the SIFT DoG keypoint detector. Unlike leading object recognition contributions on datasets such as the Caltech 101 [12], placing keypoints on a grid performs worse.

The results of comparing various local descriptors are presented in Figure 4(b). As can be seen the SIFT descriptor does better than the patch based descriptors. Given the noisy nature of the underlying images, this is not entirely surprising, however, the PCA-SIFT descriptor did not perform very well. An interesting alternative left for future work, inspired by recent work [16], is to take the outline of the binarized characters in each patch as a descriptor.

The dictionary size seems to have little effect on the results (not shown). For both the histogram based features and the distance based features, performance seemed stable and only slightly increasing when increasing the number of clusters beyond 400. Also omitted is the performance of the exponent of distance based representation for various values of the scale parameter. Performance is pretty stable with regard to this parameter over a large range of values.

Table 1 depicts the results of the various vectorization methods and similarity measures. It contains one table for each success measure, and within each table one cell for each vectorization/similarity combination. As can be seen, the best performing method by any of the four scores is the one combining a histogram representation normalized in L2, along with SVM for the similarity measure. Next follows a similar method employing OSS similarity instead.

As can be expected from previous work, combining multiple similarities together improves results. In Table 2 we compare combinations of various groups. In general, the unlearned histogram-based norms combined together do slightly better than the similar distance-based group. The combination of even all 12 unlearned norms did not do better than the best single-feature learned norm (Hist with SVM). Indeed, the combination of all SVM norms (4 different vector representations) is a leading method, second only to the combination of all 20 norms together. This final combination obtains a true positive rate of 82.9% for a false positive rate of 0.1%.

Finally, Figure 5(a) presents ROC curves obtained for various norms applied to the histogram vector representation normalized to have a unit L2 norm. While the improvements seem incremental, they actually make a significant difference in the low-false positive region (Figure 5(b)).

6. Newly found joins

As mentioned above, the negative pairs we work with are not necessarily negative. This does not affect the numerical results much, since the fraction of joins is overall-low, however it implies that there may exist unknown joins in the set of leaves that are currently available to us. We have applied our classification technique to all possible pairs of leaves and then looked at the 30 leaf pairs that are unknown to be joins, but which receive the highest matching scores.

By the time of this paper's submission we had computed these all-pairs similarities using the histogram (L2 normalized) and the SVM-scores. We then submitted the resulting pairs to a human expert for validation. The manual labor involved was about 2 and a half hours. The results of this validation are given in the accompanying supplementary material. Eighty percent of the newly detected join candidates were actual joins. Seventeen percent are not joins, and 1 pair could not be determined.

7. Conclusion and future work

We have presented a framework for identifying joins in Genizah fragments, which already provides a value for Genizah researchers by identifying unknown joins. A benchmark is developed and used for the construction of effective algorithms, borrowing from existing experience in the field of face recognition. We plan to make our benchmark together with the original and processed images and encodings available for the rest of the community in order to facilitate the efficient development of future algorithms.

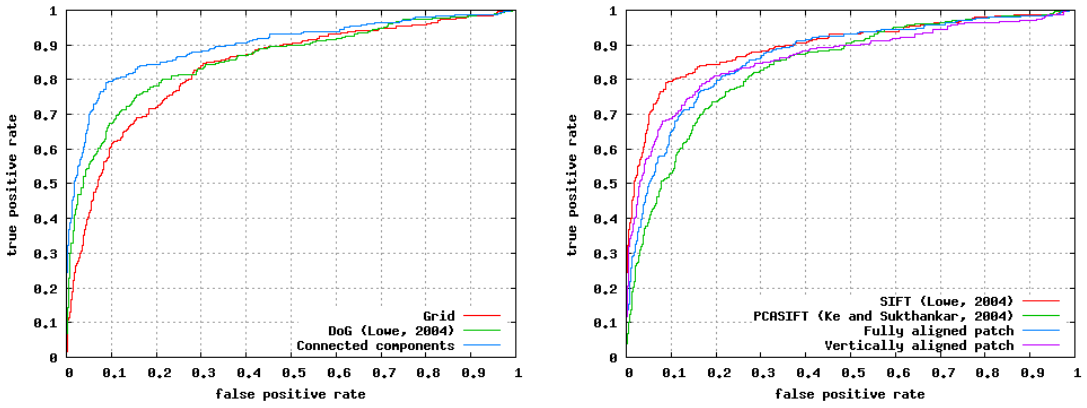
Our future plans focus on improving all aspects of the algorithms, as well as including new sources of information such as analysis of the shape of the fragment (fragments of the same join are likely to have the same overall shapes and holes), and the automatic classification of fragment material (paper/vellum). The high-resolution scanning of the Genizah documents is still taking place, and so far we were able to examine only about a percent of the fragments known to exist. Note, however, that the methods we employ are efficient and may be employed to the entire corpus in due time.

References

[1] K. Baird. Anatomy of a versatile page reader. *Proceedings of the IEEE*, 80(7), 1992. 3

[2] A. Bensefia, T. Paquet, and L. Heutte. Information retrieval based writer identification. In *Int. Conf. on Document Analysis and Recognition*, 2003. 2

[3] S. Bres, V. Eglin, and C. V. Auger. Evaluation of Handwriting Similarities Using Hermite Transform. In *Frontiers in Handwriting Recognition*, 2006. 2



(a) (b)

Figure 4. ROC curves for various keypoint detectors and local descriptors obtained for View 1. (a) Results obtained for sampling keypoints on a uniform grid, sampling keypoints according to the DoG method [15], and sampling by the proposed connected components method. (b) Results obtained for various local descriptors: SIFT [15], PCA-SIFT [11], and the two proposed patch-based methods.

Area under ROC

	L2	L1	Hlngr	SVM	OSS
Hist(L2)	0.9283	0.9431	0.9070	0.9759	0.9485
Hist(L1)	0.8711	0.9065	0.8772	0.9348	0.9323
Dist	0.9253	0.9268	0.9291	0.9511	0.9652
Exp(-d)	0.9281	0.9295	0.9271	0.9488	0.9656

Equal error rate

	L2	L1	Hlngr	SVM	OSS
Hist(L2)	0.1398	0.1321	0.1704	0.0699	0.1077
Hist(L1)	0.1895	0.1591	0.1867	0.1418	0.1235
Dist	0.1366	0.1357	0.1346	0.1032	0.0816
Exp(-d)	0.1352	0.1334	0.1357	0.1056	0.0811

Mean success (recognition) rate

	L2	L1	Hlngr	SVM	OSS
Hist(L2)	0.9342	0.9304	0.9107	0.9586	0.9497
Hist(L1)	0.8008	0.9157	0.8915	0.8000	0.9350
Dist	0.9314	0.9358	0.9349	0.9510	0.9561
Exp(-d)	0.9357	0.9398	0.9339	0.9515	0.9558

True positive rate at false positive rate of 0.001

	L2	L1	Hlngr	SVM	OSS
Hist(L2)	0.5316	0.4765	0.3311	0.7327	0.6704
Hist(L1)	0.1332	0.3515	0.2362	0.4235	0.6724
Dist	0.5480	0.5816	0.5362	0.6184	0.6541
Exp(-d)	0.5566	0.5918	0.5551	0.6760	0.6291

Table 1. Comparison of various vector representations and similarity measures on the View 2 dataset. Four success measures are presented. All scores are averaged over the 10 folds. The standard error (SE) for the mean success rate is omitted for brevity. Typical values for the SE range between 0.005 and 0.01. The four vectorization methods are: bag of keypoint histograms normalized to have a unit L2 norm; similar histogram normalized to have a sum of 1; distance based keypoint representation; and exponent of minus the distance divided by 1000. The norms compared are the L2, L1, and Hellinger norms, as well as SVM scores on the absolute values of the difference between each pair and the One Shot Similarity.

[4] M. Bulacu and L. Schomaker. Automatic handwriting identification on medieval documents. In *Int. Conf. on Image Analysis and Processing*, 2007. 2

[5] R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *PAMI*, 18, 1996. 2

[6] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004. 4

[7] I. Dinstein and Y. Shapira. Ancient hebraic handwriting identification with run-length histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 1982. 2

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6), 1981. 3

[9] G. Huang, M. Jones, and E. Learned-Miller. Lfw results using a combined nowak plus MERL recognizer. In *ECCV Faces in Real-Life Images Workshop*, 2008. 3

[10] G. Huang, M. Ramesh, T. Berg, and E. Learned-

Combination	Area under ROC	Equal error rate	Mean success ± Standard Error	TP rate at FP rate of 0.001
All 6 basic histogram norms	0.9528	0.1053	0.9486 ± 0.0052	0.6357
All 6 basic distance norms	0.9495	0.1102	0.9450 ± 0.0052	0.5658
All 12 basic norms	0.9571	0.0959	0.9514 ± 0.0044	0.6454
All 4 SVM similarities	0.9798	0.0633	0.9648 ± 0.0076	0.7811
All 4 OSS similarities	0.9703	0.0725	0.9583 ± 0.0048	0.7276
All 5 similarities of Hist (L2)	0.9769	0.0612	0.9671 ± 0.0057	0.7755
All 20 similarities together	0.9838	0.0531	0.9735 ± 0.0070	0.8286

Table 2. Results obtained on View 2 for various combinations of similarity measures. The combinations are (1) L1, L2, or Hellinger norms for histograms normalized by L1 or by L2; (2) same three norms for distances and for exponent of minus the distances divided by 1000; (3) the combination of (1) and (2) above; (4) the four SVM similarities obtained for the two histograms and the two distance based representations; (5) the four One Shot Similarities obtained on the four vector representations; (6) all similarities computed on the bag of keypoints histogram representation which is normalized to unit L2 norm; and (7) a combination of (3), (4), and (5).

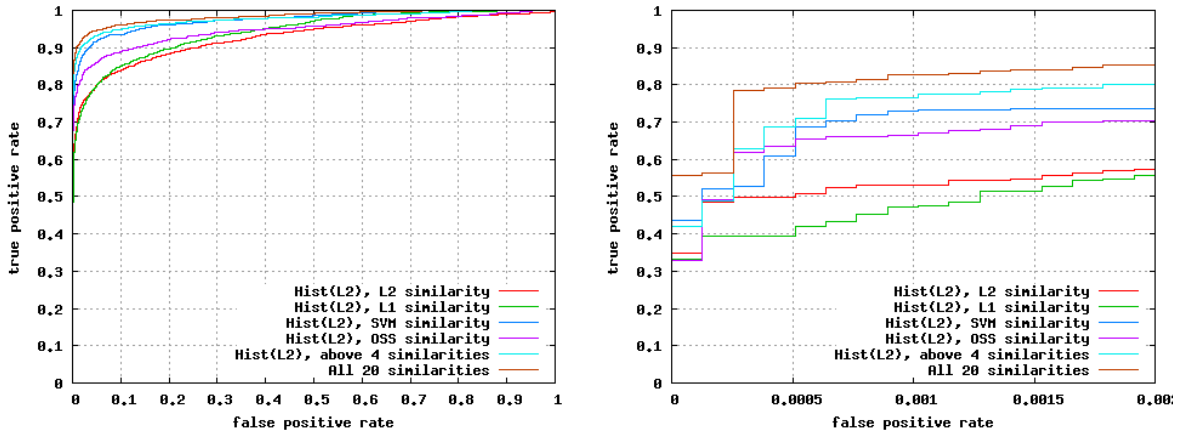


Figure 5. ROC curves averaged over the 10 folds of View 2. The plots compare the results obtained for the L1, L2, SVM and OSS similarities obtained on the histogram (L2 normalization) representation. Also shown are the combination of the above four similarities and the result of combining all 20 similarities obtained for the four vector representations (2 histogram based and 2 distance based; 5 similarities for each). (a) Full ROC curves; (b) A zoom-in onto the low false positive region.

Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. UMASS, TR 07-49, 2007. 3, 5

[11] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. *CVPR*, 2004. 4, 7

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 4, 6

[13] G. Leedham, S. Varma, A. Patankar, and V. Govindarayu. Separating text and background in degraded document images. In *Frontiers in Handwriting Recognition*, 2002. 2

[14] H. G. Lerner and S. Jerchower. The Penn/Cambridge Genizah fragment project: Issues in description, access, and reunification. *Cataloging & Classification Quarterly*, 42(1), 2006. 2

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60, 2004. 3, 4, 7

[16] M. Panagopoulos, C. Papaodysseus, P. Rousopoulos, D. Dafi, and S. Tracy. Automatic writer identification of ancient Greek inscriptions. *PAMI*, 31(8), 2009. 2, 6

[17] N. Pinto, J. DiCarlo, and D. Cox. How far can you get with a modern face recognition test set using only simple features? In *CVPR*, 2009. 3, 5

[18] S. C. Reif. *A Jewish Archive from Old Cairo: The History of Cambridge University's Genizah Collection*. Curzon Press, Richmond, England, 2000. 2

[19] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, 2005. 4

[20] S. N. Srihari and V. Govindaraju. Analysis of textual images using the Hough transform, 1989. 3

[21] L. Wolf, S. Bileschi, and E. Meyers. Perception strategies in hierarchical vision systems. In *CVPR*, 2006. 5

[22] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in ECCV*, 2008. 3, 5

[23] D. H. Wolpert. Stacked generalization. *Neural Netw.*, 5(2), 1992. 5