Augmenting historical manuscripts with automatic hyperlinks

Xiaoyue Wang
Department of Computer Science and Engineering
University of California Riverside
Riverside, USA
xwang@cs.ucr.edu

Eamonn Keogh
Department of Computer Science and Engineering
University of California Riverside
Riverside, USA
eamonn@cs.ucr.edu

***Abstract*—Hyperlinks are so useful for searching and browsing modern digital collections that researchers have longer wondered if it is possible to retroactively add hyperlinks to digitized historical documents. There has already been significant research into this endeavor for historical *text*; however, in this work we consider the problem of adding hyperlinks among *graphic* elements. While such a system would not have the ubiquitous utility of text-based hyperlinks, as we will show, there are several domains where it can significantly augment textual information.**

While OCR of historical text is known to be a difficult problem, the actual words themselves are inherently discrete. Thus, two words are either identical or not. This means that off-the-shelf machine learning algorithms, including semi-supervised learning, can be easily used. However, as we shall demonstrate, semi-supervised learning does not work well with images, because we cannot expect binary matching decisions. Rather we must deal with degrees of matching. In this work we make the novel observation that this "degree of matching" biased algorithms make overly confident predictions about simple shapes. We show that a simple technique for correcting this bias, and demonstrate through extensive experiments that our method significantly improves accuracy on diverse historical image collections.

Keywords-Historical Manuscripts, Hyperlinks, Semi-Supervised Learning

I. INTRODUCTION

The utility of hyperlinks for searching and browsing modern digital collections has motivated researchers to attempt to retroactively add hyperlinks to digitized historical documents [11]. There has already been significant research into this endeavor for historical text; one example is Schilit et al. creating links on quotations between book passages [17]; however, in this work we consider the problem of adding hyperlinks among graphic elements.

As a simple motivating example, consider the famous Manesse Codex [9] (*Große Heidelberger Liederhandschrift*), an illuminated manuscript in codex form, copied and illustrated between 1304 and 1340 in Zurich. The stylized and archaic text is difficult to read, and one must imagine, difficult for an OCR system [15][8] to parse. However, as shown in Figure 1, we could attempt to automatically cross reference pages by noting that a similar heraldic shield, with similar shape and color, appears in two pages.

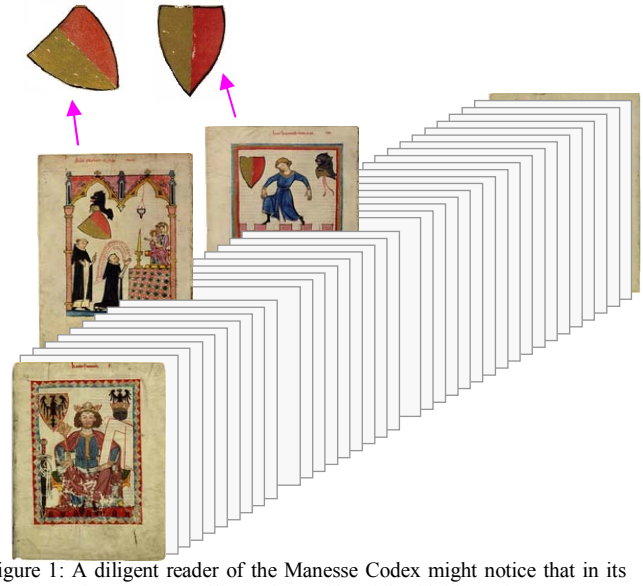

Figure 1: A diligent reader of the Manesse Codex might notice that in its 424 pages there are many illustrations of heraldic shields, and on pages 48v and 59v, the same shield appears.

In this case, the repeated motif is not a coincidence; the pages in question refer to Bruder Eberhard von Sax and Herr Heinrich von Sax, respectively.

As we shall see, in addition to heraldic shields, other cultural artifacts and natural history domains may lend themselves to a retroactive placement of hyperlinks. Creating hyperlinks on these historical data could have significant impact in historical studies by providing better understanding of the manuscripts in the context of contemporary and later work.

To create hyperlinks is the example above; we require algorithms that can match images using multiple visual features (shape and color). However, in this work we only consider creating hyperlinks by matching similar shapes. We ignore this issue of color matching for simplicity of presentation, other than to note that this is a relatively easy problem to solve [19].

An obvious question to ask is which shape matching algorithm to choose in this context of historical manuscripts. The algorithm should meet the following two requirements: (i) it should work well on shape data taken from historical manuscripts, which by their very nature may be degraded and noisy; (ii) it should be scalable to the large datasets we wish to consider.

There are many shape matching algorithms that can build reliable classifiers upon fully labeled data. However,

none of these algorithms can be considered in historical document studies, where most of the data is unlabeled. In fact, annotating unlabeled data is generally recognized as a difficult problem to solve. The traditional method of annotating data manually is difficult, expensive and (human) time consuming; even when aided by state-of-art automatic annotation algorithms.

Considering the difficulty of obtaining labeled shapes from ancient manuscripts, a shape matching method that can perform well on limited amount of labeled shapes is highly desirable. In this work we will show that a classifier using a semi-supervised learning framework will solve the problem. A semi-supervised learning classifier starts with only a few labeled instances and gradually learns labels for those unlabeled instances, adding the most confidently predicted instance to the training set. We propose a novel semi-supervised technique to build shape classifiers upon only limited amount of labeled shapes from historical data.

It is clear that the utility of any semi-supervised learning classifier for shapes will critically depend on the distance measure we choose. We claim that current distance measures are inadequate for dealing with the diverse shapes we may expect to encounter in this domain. Many existing measures do work well if all the objects have approximately the same complexity[1], but can fail if the dataset contains a mixture of complex and simple objects. In this work, we show how to define a new distance measure that can reflect the true degree of similarity between shapes, even in the face of such diverse datasets. We show that the new distance measure is significantly better than traditional ones when using a semi-supervised learning framework.

A. *Our Motivating Observation*

One of the most common ways of computing shape similarity is to match based on their features in the "time series" domain [18][7][21]. The basic idea is to transform the two-dimensional shapes to a one-dimensional "time series" representation, and then we can employ distance measures and indexing structures available for time series, e.g. Euclidean Distance measure.

While examining this way of using Euclidean distance to compute shape similarity, we observed a fact that informs the rest of this work. We noticed that different shapes exhibit different "*complexity*", and that all things being equal, two simple shapes are more similar to each other than two complex shapes, when subjectively humans would claim they are equally similar. That is to say, given pairs with the same subjective similarity, the ones with more complex shapes will produce larger Euclidean distances than the pairs with simple shapes. We can demonstrate this idea with two synthetic pairs that are of the same subjective similarity.

Figure 2 shows the time series representations of shapes for two seashells, one simple shape A and one complex shape B. For each of the two shapes, we create a synthetic "*twin*" shape by copying it. Then to the twin shapes, we add identical amounts of distortion (We defer the details on distorting shapes to Section III). Given that we have added an identical amount of distortion to each shape, we might expect to find d(A,A') = d(B,B'); however, this is not the case.

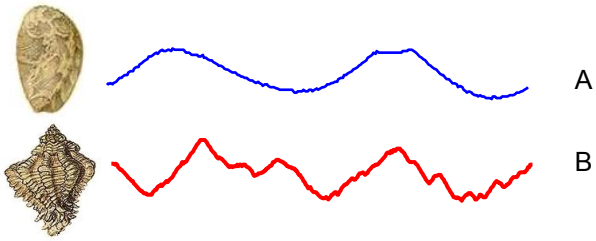

Figure 2: Time series representations of shapes for two seashells of different complexity, one simple shape A and one complex shape B.

Figure 3 shows the newly generated time series of shape A' and B', together with their original copies A and B. The dendrogram in Figure 3 shows that the pair of simple shapes (A and A') has a much smaller Euclidean distance than the pair of more complex shapes (B and B'), in spite of the fact that these two pairs share the same subjective similarity.

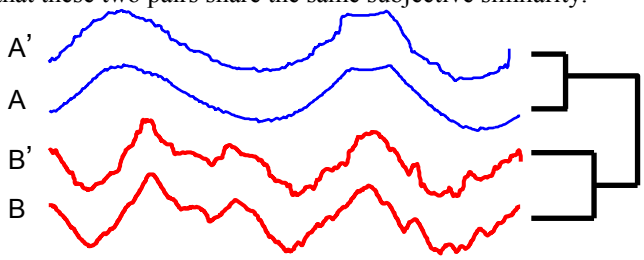

Figure 3: Dendrogram of two exact similar pairs of time series of shape.

In light of this observation, we find that the Euclidean distance measure does not reflect the true relative degree of similarity between pairs of objects which have different complexities. This finding has implications for any instance-based machine learning technique, since there will be a systematic bias in learning.

To mitigate this problem we introduce a complexity-invariant distance measure, which is essentially the Euclidean distance measure augmented by a complexity penalty. An extensive empirical evaluation on real datasets demonstrates that the new distance measure generally produces greater classification accuracy than the classic Euclidean distance measure.

The rest of this paper is organized as follows: In Section II, we discuss background materials and related work, before we formally introduce our solution in Section III. An empirical evaluation of the proposed algorithm is provided in Section IV. Finally, in Section V we offer some conclusions and directions for future work.

II. BACKGROUND AND RELATED WORK

A. *Semi-supervised Learning*

There are many semi-supervised learning methods in the literature [22], and some often-used ones include: generative models [13], self-training [1], co-training [2], transductive support vector machines [5], and graph-based methods [10]. However, all the methods above except the self-training one

[1] We will define "complexity" later, but intuitively an apple has a simple shape and a comb a complex one.

have strong restrictions/assumptions either in choosing models, features, similarity functions or kernels of the problem set. Given the fact that historical document data is highly diverse, we make as few assumptions as possible about the problem structure. For this reason, we use the self-training method that has the fewest requirements on the problem set.

In self-training, the initial classifier is trained by the small amount of labeled data. This classifier model is then used to classify the unlabeled data. The most confidently classified unlabeled object, together with its predicted label, is added to the training set. This process is repeated until some certain stopping criterion is met. In this work, we adopt a generic self-training scheme.

B. *Shape Macthing*

In this section, we discuss the shape representation we adopt in this work.

There are a plethora of shape measures in literature; the paper [18] provides an excellent review. However, in the specific context of large datasets, e.g. historical archives, the time series representation of shapes is a good choice, as it is simple, parameter-free and relatively invariant to distortions [7].

The basic idea is to transform the two-dimensional shape into a one-dimensional "time series". Figure 4 gives a visual intuition of the transformation. The distance from every point on the profile to the center is measured and treated as the Y-axis of a time series of length n.

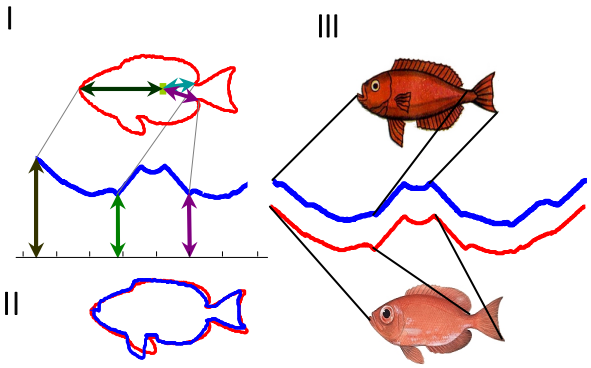

Figure 4: I) A visual intuition of the conversion of a two-dimensional shape to a one-dimensional "time series". II) Two shapes that are similar in the shape space will also be similar in the time series shape. III) Here we compare a fish (*Priacanthus arenatus*) from a 40-year-old Cuban manuscript to a related fish (*Priacanthus hamrur*) from a manuscript published in 1899.

Given that we are able to represent shapes in the time series domain, this still leaves the question of which distance measure to choose. Various distance measures exist that are available for indexing time series. Studies in [7] and elsewhere show that the Euclidean distance on this representation is at least as competitive as more complex measures and representations on various shape matching problems.

Suppose we have two time series of shapes, A and B of length n.

$$A = a_1 a_2 ... a_i ... a_{n-1} a_n$$
$$B = b_1 b_2 ... b_i ... b_{n-1} b_n$$

The Euclidean Distance(ED) between these two shapes is

$$ED(A,B) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

III. A NOVEL SEMI-SUPERVISED LEARNING METHOD

A. *Complexity-invariant Distance Measure*

We propose to adjust the classic Euclidean Distance (ED) using a parameter r. The distance for shapes of different complexities will be penalized by different values of r, learned directly from the shapes themselves. A Complexity-Invariant Distance (CID) measure which extends from ED is defined as follows:

$$CID(A,B) = \frac{ED(A,B)}{r}$$

The value of r is not a fixed number and it is determined on a case-by-case basis. Intuitively, a good r is the one with a small value when A and B are simple shapes, so that more penalty will be added to their original ED distance. In cases of comparing two complex shapes, we would expect r to be a large value that would cause a smaller penalty to ED distance. Given the fact that r is dependent on the complexity of both A and B, we introduce the definition of r as:

$$r(A,B) = \frac{complexity(A) + complexity(B)}{2}$$

To determine the value of r, we need to have an appropriate evaluation of the shape complexity. Recall the observation on the relationship between shape complexity and similarity distances as discussed in Section I: pairs of similar but complex shapes will produce a larger distance than the ones of similar but simple shapes. In light of this observation, we propose measuring the complexity for a shape based on the distance between a synthetic pair of shapes, which is formed by the original shape and its similar example shape.

For each shape we create one pseudo non-identical twin of a shape, which is generated by copying then slightly distorting the original shape. The intuition behind this idea is as follows: We notice that the pseudo-example shape will highly resemble the original shape due to a limited amount of distortion. Given the fact that two similar shapes will be likely to have similar values in shape complexity, the pseudo-example shape will be close to the original one in shape complexity. If the original copy of the time series is complex, the newly generated example will also be a complex shape. In contrast, the pseudo-example will be a simple one given the original shape is simple. In this sense, if the distance between a synthetic pair yields a large value, we will expect that both shapes in this pair are complex shapes. Therefore, the original copy of the shape will be a complex one. In contrast, we will know the original shape is simple if the distance between the pair is small.

A method of generating the pseudo-example by distorting the time series representation is described in Table 1.

TABLE 1: ALGORITHM TO GENERATE PSEUDO-EXAMPLE

Algorithm Generate_Pseudo-example (A,*p*)	
	randomly choose p percent of points from time series A; remove these points from A, then get the new time series B; upsample B to be same length of A; return B;

We warp the two shapes in the time series domain by removing the same amount of random points from the original time series and then upsampling them to the original length of the time series. To average out the effect of the randomness in creating the similar example, we iterate the process by 10 times. During the i^{th} iteration, we create a similar example B_i, and the shape complexity of A is determined by calculating the average value of ED distances for these 10 pairs of A and B.

$$complexity(A) = \underset{1 \le j \le 10}{avg} \{ED(A, B_i)\}$$

Note that in this algorithm, we have two parameters. Apart from the input of the time series to be computed for shape complexity, we also have a percentage p as a parameter, $p \in [0,100)$. This variable determines how much distortion will be applied to the original time series to generate the pseudo-example. It is easy to imagine that the two extreme cases, that is, $p = 0$ or 100, would produce bad synthetic examples of shapes: $p = 0$ produces no distortion to the newly generated example and every shape would get the same value of shape complexity, which is zero; whereas p approaching 100 would generate pairs of shapes that are completely irrelevant, and therefore complexities calculated from such pairs would only be random values. The experiments in Section IV show that $p = 50$ will produce good pseudo-examples.

B. Semi-supervised Classifier with CID

In this section, we introduce a novel semi-supervised classifier which leverages off our new distance measure (CID) that is invariant to the complexities of shapes. We employ the one-nearest-neighbor with CID measure as the classifier and the algorithm is outlined in Table 2. The algorithm begins by having a human label a small number (as few as just one) of examples from each class. Thereafter the algorithm labels all the remaining examples.

TABLE 2: SEMI-SUPERVISED CLASSIFICATION ALGORITHM
ALGORITHM TO GENERATE PSEUDO-EXAMPLE

Algorithm Semi-supervised_Classification (m,D)	
1	choose m examples from each of n different classes in D as the training set L;
2	the rest of D constructs the testing set U;
3	accuracy(1:m*n) = 1;
4	loop j for n-m times
5	find in U the one with the smallest CID distance to the training set L;
6	accuracy(m*n+j) = Number of correctly labeled data/Number of labeled data;
7	add the object into the set L and remove it from U ;
8	end
9	return accuracy;

IV. DIVERSE HISTORICAL MANUSCRIPT DATASET

We extracted shapes of five different categories from historical manuscripts either available online or from the authors' personal collections of historical books. The earliest images are heraldic shields from the fourteenth century [9]. In contrast, the earliest images of projectile points (arrowheads) date only back to 1891 [23]. However, if the term "historical manuscript" is taken in its broadest context, then is it worth noting that petroglyphs of the Archaic Period are replete with detailed images of arrowheads. The Jeffers Petroglyphs site in southwestern Minnesota is a notable example [3]. In ongoing work, we are gathering data to allow us to test our ideas on these three to eight-thousand-year-old "texts" [16].

In some cases, we augmented the historical data with some examples from modern counterparts. Our final dataset contains 2,928 shapes in total. We have 518 fish, 561 arrowheads, 754 "butterflies" [2], 414 shields and 681 seashells. As noted above, we have already placed all of the extracted images in the public domain.

For brevity, in this work we do not discuss techniques for extracting the primitive shapes from the historical documents. This has been an area of active research for more than two decades, both for specialist collections such as historical music documents [14] and the more difficult general cases [4][12][14]. Our experiments on diverse datasets suggested that this processing step ranges from trivial to extremely difficult. We defer a detailed discussion of these issues to a parallel work.

A. Experiment Settings

In the following experiments, we test our semi-supervised learning classifier with a comprehensive set of experiments. We compare the semi-supervised approach to a fully supervised one-nearest-neighbor method.

Our new semi-supervised learning classifier is implemented under the scheme of the Semi-supervised_Classification algorithm described in Table 2. For simplicity we assume that m, the initial size of the labeled training set, is set to one in all experiments. This corresponds to the user labeling just one item from each class. To prevent unduly optimistic results due to an experienced user carefully labeling only the best exemplar, we have the computer randomly choose the objects to label.

The performance of the semi-supervised learning classifier is greatly varied, resulting from different initial labeled examples to start with. Atypical/unrepresentative labeled examples, which are the ones with unusual shapes in our scenario, would greatly deteriorate the classification accuracy. This is especially the case when the number of labeled examples is extremely small. As mentioned earlier, we could hope to choose better representative examples for each class from human subjective selection, but this method is still biased to human perception on which shapes look like a representative one within the category. Although we can rely on algorithms that can carefully choose the labeled data to start with, e.g. checking for unusual time series [20],

[2] The "butterflies" dataset also includes other species such as moths and dragonflies.

in our technique, we simply minimize the performance variance by randomly choosing the initial labeled set and running the algorithm for a few iterations. The final classification result is the averaged value over all the iterations. For all the experiments in our work, we use 30 iterations.

To test the performance of our semi-supervised learning classifier, we first construct a two-class classification problem. Out of the five different categories of shapes, we pick two categories as class 1 and class 2. Objects from all other categories are considered as non-class instances. We then design a multi-class problem. The five different categories of shapes construct five different classes.

We begin our classification process with one labeled example from each class, that is, m = 1 for the Semi-supervised_Classification algorithm.

In the following experiments, "1-NN" refers to the classic supervised learning technique, that is using one Nearest-Neighbor classifier and ED distance measure; "SSL" refers to Semi-Supervised Learning methods.

We first test our algorithm on a two-class classification problem. There are many combinations to choose the two classes out of five different categories. In this section, we show the results on three combinations.

As we incorporate non-class data into our classification problem, we further specify the method of calculating classification accuracy within the algorithm:

accuracy = Number of correctly labeled class 1 or class 2 data/Number of labeled class 1 or class 2 data

In the first experiment, we choose shields and arrowheads as the two classes to be classified. All instances from the other three categories construct the non-class set.

Experiments are first conducted to show the effectiveness of parameter p in Algorithm Generate_Pseudo-example. Different values of p can result in different values of shape complexity. Intuitively, an ideal p will be of value around 50. Those values of p will result in moderate distortions to shapes when generating the pseudo-examples, thereby making the shape complexity more accurate.

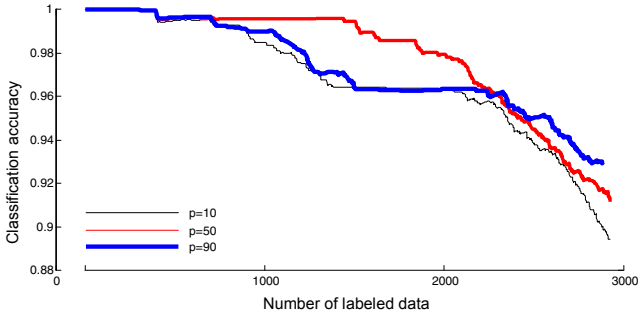

Figure 5: Classification with p = 10/50/90

Figure 5 shows three experiments with p = 10/50/90 and it works the best when p = 50. Clearly, p = 50 will produce more accurate shape complexity values that will help generate better classifiers. In the following experiments, we all use p = 50.

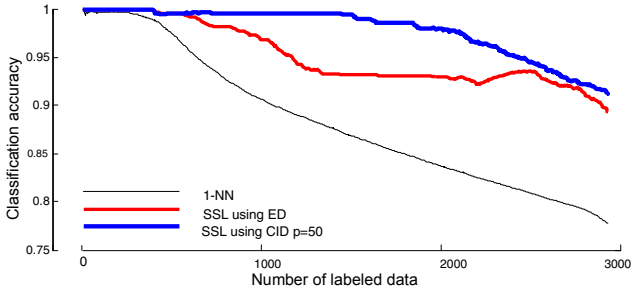

Figure 6: A two-class classification problem on shields and arrowheads.

After the discussion of parameter p, now we are in the position of showing the comparison for three different classifiers in this problem set: the fully-supervised 1-NN classifier, the SSL classifier using the ED distance measure and the SSL classifier using the CID measure. As shown in Figure 6, the two SSL methods perform better than the 1-NN algorithm, and the SSL classifiers using CID works the best.

Although the accuracy for the SSL classifier with ED or with CID does not differentiate much at the end of the two algorithms, a big disparity exists in the middle of the two labeling processes. The SSL with ED already drops close to 0.9, while the SSL classifier with CID still keeps almost perfect accuracy when it comes to the halfway point of the whole labeling process. This suggests that the CID technique, which adjusts the distance measure with shape complexity information, provides a better distance measurement.

To illustrate the generality of our algorithm, we also run experiments on other two-class problem sets. Figure 7 shows the classification results on the two-class set constructed by seashells and butterflies. In the experiment shown in Figure 8, the instances from category shields and butterflies form the two-class set. Like the results in the previous experiment, the SSL classifier with ED still performs the best.

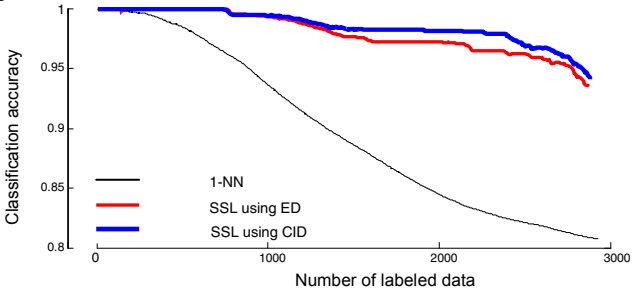

Figure 7: A two-class classification problem on seashells and butterflies.

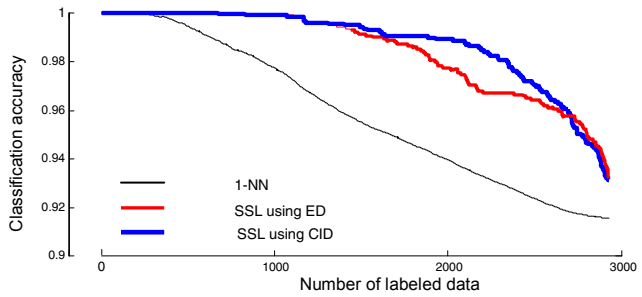

Figure 8: A two-class classification problem on shields and butterflies.

V. CONCLUSIONS AND FUTURE WORK

In this work we have leveraged off a novel observation about the effects of shape complexity on distance measures to produce a more accurate semi-supervised learning algorithm. There are several directions for future work. One obvious limitation of our work is that it only considers shape information. Some domains, such as heraldic shields and butterflies, clearly require color and/or texture information in order to correctly discriminate objects. Finally, we plan to release an open source code repository for all our work, and archive the extracted datasets in the UCI machine learning archive, to allow confirmation of, and extensions to, our findings.

ACKNOWLEDGMENT

We thank all the donors of datasets, and the data archivists and digital librarians who made this work possible. This work was supported by NSF awards 0803410 and 0808770.

REFERENCES

[1] J. Besemer, A. Lomsadze and M. Borodovsky. (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. *Implications for finding sequence motifs in regulatory regions. Nucleic Acids Res., 29, 2607±2618.*

[2] A. Blum and T. Mitchell. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT '98) (pp. 92–100).*

[3] D. P. Gardner. (2004). Minnesota Treasures: Stories Behind the State's Historic Places. *St. Paul, Minnesota: Minnesota Historical Society. ISBN 0-87351-471-8.*

[4] I. Granado, P. Pina, and F. Muge. Automatic Feature Extraction on Pages of Antique Books through a mathematical Morphology based Methodology. *Lecture Notes in Computer Science, vol. 1923, pp.1-13, 2000.*

[5] A. K. Jain. and A. Vailaya.(1996). Image retrieval using color and shape. *Pattern Recognition, vol. 29, pp. 1233–1244, 1996.*

[6] T. Joachims. (1999). Transductive inference for text classification using support vector machines. *In Proceedings of ICML-99, 16th International Conference on Machine Learning(Bled, SL, 1999), pp. 200–209.*

[7] E. Keogh, L. Wei, X. Xi, S. H. Lee, and M. Vlachos. LB_Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures. *In Proceedings of Very Large Databases (VLDB'06), 2006, pp 882-893.*

[8] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *IJDAR 9(2), 167–177 (2007).*

[9] W. Koschorreck, W. Werner, editors. Commentary to the facsimile edition, with essays by Wilfried Werner, Ewald Vetter, Walter Koscharreck, Hugo Kuhn, Max Wehrli and Ewald Jammers. *Kommentar zum Faksimile des Codex Manesse: Die grosse Heidelberger Liederhandschrift (Kassel: Ganymed) 1981.*

[10] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. (2005). Semi-supervised graph clustering: A kernel approach. *Proc.ICML-2005.*

[11] F. Le Bourgeois, and H. Kaileh: Automatic Metadata Retrieval from Ancient Manuscripts. *Document Analysis Systems* 2004: 75-89

[12] M. Mengucci, and I. Granado. Morphological Segmentation of Text and Figures in Renaissance Books (XVI Century). to present at *ISMM'2000 — 5th International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing, Palo Alto, USA, June 2000.*

[13] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell. (2000).Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39(2/3), pp. 103 – 134, 2000.*

[14] J. R. C. Pinto, P. Vieira, M. Ramalho, M. Mengucci, P. Pina and F. Muge. *Ancient Music Recovery for Digital Libraries. ECDL 2000: 24-34.*

[15] T. M. Rath, and R. Manmatha. Word spotting for historical documents. *IJDAR 9(2-4): 139-152 (2007).*

[16] E. G. Riggs. (2001). Late Archaic Projectile Point Petroglyphs. *American Indian Rock Art Volume 27:279-284.*

[17] B. Schilit and O. Kolak. (2008). Exploring a digital library through key ideas. *JCDL 2008.*

[18] R. C. Veltkamp, and L. J. Latecki. Properties and Performance of Shape Similarity Measures. *In Proceedings of IFCS 2006 Conference: Data Science and Classification. July, 2006.*

[19] X. Wang, L.Ye, E. Keogh, C. Shelton: Annotating historical archives of images. JCDL 2008: 341-350

[20] D. Yankov, E. Keogh, and U. Rebbapragada (2007). Disk Aware Discord Discovery: Finding Unusual Time Series in Terabyte Sized Datasets. *ICDM 2007.*

[21] D. Yankov, E. Keogh, L. Wei, X. Xi and W. Hodges (2007). Fast Best-Match Shape Searching in Rotation Invariant Metric Spaces. *SIAM International Conference on Data Mining (SDM'07).*

[22] J. Zhu. (2005). Semi-supervised learning literature survey. Computer Sciences Technical Report TR 1530. University of Wisconsin-Madison.

[23] Annual Report of the Bureau of American Ethnology to the Secretary of the Smithsonian Institution 1891.