IBN SINA: A database for research on processing and understanding of Arabic manuscripts images

Reza Farrahi Moghaddam
Synchromedia Laboratory
ETS, Montréal, (QC) Canada
H3C 1K3
reza.farrahi@synchromedia.ca

Mohamed Cheriet
Synchromedia Laboratory
ETS, Montréal, (QC) Canada
H3C 1K3
mohamed.cheriet@etsmtl.ca

Mathias M. Adankon
Synchromedia Laboratory
ETS, Montréal, (QC) Canada
H3C 1K3
mathias.adankon@synchromedia.ca

kostyantyn Filonenko
McGill University
kfilonenko@hotmail.com

Robert Wisnovsky
Morrice Hall 026
McGill University
robert.wisnovsky@mcgill.ca

ABSTRACT

This paper describes the steps that have been undertaken in order to develop the IBN SINA database, which is designed to apply learning techniques in the processing and understanding of document images. The description of the preparation process, including preprocessing, feature extraction and labeling, is provided. The database has been evaluated using classification techniques, such as the SVM classifiers. In order to make the database compatible with these classifiers, the labels of the shapes have been translated into a set of bi-class problems. Promising results with the SVM classifiers have been obtained.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]; J.5 [**Arts and Humanities**]

1. INTRODUCTION

Transliteration and understanding historical manuscripts are two challenging problems for DIAR community. This is especially the case with Arabic manuscripts, which exhibit a wide variety of handwriting styles [8]. A key element in the development of high-performance methods is the availability of related databases. However, databases of Arabic handwritten documents are still in their primary development phases [11]. In [3], a comprehensive database for Arabic handwriting recognition is presented. It includes digits, numerical strings, and 70 words. In has been developed using a specially designed data-entry form. In [2], a database for the recognition of legal amounts and Arabic sub-words in Arabic cheques has been developed. The database contains 1547 legal amounts and 23325 sub-words. The cheque images have been segmented, binarized and denoised. The labeling of sub-words has been performed manually using an interface and based on a pre-defined vocabulary. Finally, the labels have been verified by comparing the interpreted values of each cheque obtained from the legal and courtesy amounts' labels. There exist some similar databases for English-language documents. In [9], a database for research on handwritten text recognition has been created. It consists of city names, state names, and ZIP codes scanned from mail in a specific post office. Different parts of mail address of each scanned mailpiece are linked inside the database. Even full sentence (English) databases have been developed [13, 12]. The database in [13] has been derived from the LOB corpus [10]. The rationale behind full-sentence databases is to provide the ability to use linguistic knowledge beyond the lexicon level.

Following [6], segmentation-free methods have shown promising results in word spotting (WS) applications. Although, word spotting techniques are able to extract and index many keywords within a manuscript, complete transliteration requires higher performance. In this paper, work on the development of a database based on the word spotting features is described. The database contains verified and validated labels of sub-words or connected components (CCs) of an Arabic manuscript as described in the following sections.

The organization of the paper is as follows. In section 2, the process of data selection and the description of the manuscript used for the development of the database are discussed. In section 3, preprocessing of the document images is presented. Feature extraction based on the WS features is described in section 4. In section 5, automatic labeling of the shapes using the library developed by the WS method is presented. In a novel representation, in section 6, the labels are translated into several bi-class problems. The evaluation and validation of the database at this stage is presented in section 7. The process of verification of the labels by Expert and re-evaluation of the database is discussed in section 8. Finally, the conclusions and some prospects for the future work are presented.

DAS '10, June 9-11, 2010, Boston, MA, USA

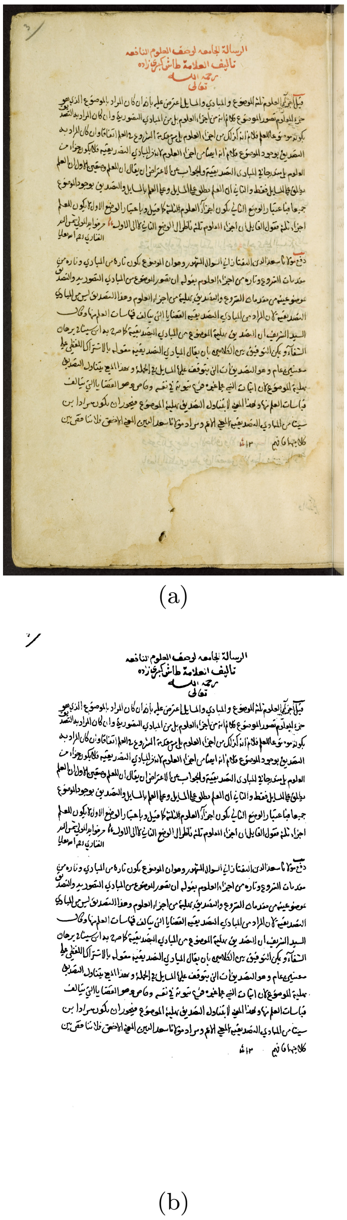
2. DATA SELECTION

The database is built on a manuscript-image provided by the Institute of Islamic Studies (IIS), McGill University, Montreal. IIS is home to the *Rational Sciences in Islam*[1] (RaSI) project, which is in the process of creating a large-scale, unique database of Islamic philosophical and scientific manuscripts, most written in Arabic, but some in Persian and Turkish [16]. One of RaSI's component initiatives is the *Post-classical Islamic Philosophical Database Initiative*[1] (PIPDI), directed by Prof. Robert Wisnovsky. PIPDI is responsible for the indexing of metadata pertaining to, and data contained in, approximately 600 works of philosophy produced during the post-classical period of Islamic intellectual history (1050-1850). Its dataset is stored in a 30-terabyte storage space.

The author of the manuscript was Sayf al-Din Abu al-Hasan Ali ibn Abi Ali ibn Muhammad al-Amidi (d. 1243A.D.). The title of the manuscript is *Kitab Kashf al-tamwihat fi sharh al-Tanbihat* (i.e., Commentary on Ibn Sina's *al-Isharat wa-al-tanbihat*). Among all of Avicenna's philosophical works, his *al-Isharat wa-al-tanbihat* received the most attention from subsequent philosophers and theologians. The attention paid to this work was particularly intense in the period between the late twelfth century to the first half of the fourteenth century, when more than a dozen comprehensive commentaries were composed. These commentaries were one of the main methods of approaching, understanding and developing Avicenna's philosophy, and therefore any study of the post-Avicenna's philosophy needs to pay specific attention to this commentary tradition. *Kashf al-tamwihat fi sharh al-Tanbihat*, one of the early commentaries written on *al-Isharat wa-al-tanbihat*, is an unpublished commentary which still awaits critical edition by scholars. The document images have been obtained using camera imaging (21 Mega pixels, full frame CMOS sensor, 1/4 sec shutter speed) at 300 DPI resolution. Each full color image is around 800KBytes in size. The selected dataset consists of 51 folios which correspond to 20722 CCs (almost 500 CC on each folio).

3. IMAGE PREPROCESSING

Before extraction of the connected components (CCs, subwords), the document images have been preprocessed to enhance them and correct the degradation. For this purpose, a preprocessing procedure based on the multi-level classifiers [7] has been used. Based on a few parameters, such as the stroke width w_s, the line height h_l, and the line extent h_s, a set of multi-level classifiers have been obtained for each input image. In this work, we use the stroke map (SM), the edge profile (EP), the stroke gray level (SGL), and the estimated background (EB) [7]. Then, the restored images are obtained by combining the classifiers [6]. For the selected manuscript, the following set of parameters is used: $w_s = 7$, $h_l = 130$ and $h_s = 35$. A sample document image and its binarized version are provided in Figure 1. Having the restored document images, the CCs have been extracted. CCs are connected black regions on the binarized images, and can be easily obtained. The main goal is to label those CCs. To achieve this, the clustering technique of our word spotting (WS) method [6] has been used to assign an initial label to each CC. This procedure is described in the next section.



(a)

(b)

Figure 1: A sample document image. a) original image. b) The binarized version after preprocessing.

4. FEATURE EXTRACTION

We follow [6] in which a clustering technique has been used where the CCs are clustered around basis CCs (BCCs) selected from the whole set of the CCs. In this way, a library of the CCs is built in which similar CCs can be determined easily based on their BCC. In order to compare CCs and measure their similarity, a set of skeleton-based features is used. In the next sub-section, a general overview of the features is provided.

4.1 Feature extraction as a transformation to feature space

[1] http://islamsci.mcgill.ca/RASI/;
http://islamsci.mcgill.ca/RASI/pipdi.html

Here, as a general approach, we consider several transformations on the skeleton image to several spaces representing different aspects of the shape under study. Let's consider u and u_{skel} be the images of a CC and its skeleton respectively, where $u_{skel} : \Omega_{skel} \rightarrow \{0, 1\}$, and $\Omega_{skel} = \Omega_u \subset \Omega \subset \mathbb{R}^2$. The domain Ω is the domain of the whole page that hosts the CC under study. Let's call T the set of transformations which map u_{skel} to the proper spaces: $T = \{T_i | T_i : \Omega_{skel} \rightarrow (\mathbb{R}^{m_i})^{n_i}, i = 1, \cdots, n_T\}$, where n_T is the number of transforms, m_i depends only on the transformation T_i, while n_i depends also on the complexity of the u_{skel}. In other words, the transformation T is a variable transform for each CC. Especially, the dimension of the target spaces are variable. In summary, T provides a one-dimensional representation for the skeleton images:

$$\begin{aligned} u_{skel} &\overset{T}{\rightarrow} v = \{f_i\}_{i=1}^{n_T}, \\ (\mathbb{R}^{m_i})^{n_{i,u_{skel}}} \ni f_i &= T_i(u_{skel}) = \{\phi_{i,j}\} \end{aligned} \tag{1}$$

Topological features of skeletons have been used extensively in object recognition. In this work, we use a set of topological features adapted to document images. The first transformation, T_1, assigns a set of features to each branch point (BP) of a CC, depending if that BP is connected to a loop, is connected to an end point (EP), and is connected to another BP. Therefore, $m_1 = 1$, and n_1 is equal to the number of BP of CC. The second transformation, T_2, generates topological features associated with EPs. It is set based on if the EP is connected to a BP, is connected to another EP, and what is its vertical state with respect to vertical CM.

Also, the states of the EPs and BPs with respect to the dots (singular points, SPs) are converted into two additional feature sets. For BPs, it takes into account the vertical location of the dot (if it is above or below the BP) and is compiled as T_3. It has two features. Therefore, $m_3 = 2$. The next transformation, T_4, does the same job for EPs. However, because of high degree of variations in the position of dots with respect to EPs, we just consider existence of a dot near the EPs ($m_4 = 1$). In both T_3 and T_4, nearest state is calculated implicitly; each dot is assigned to closet BPs (and EPs). To tolerance the writing variations, a dot is assigned to its two nearest BPs (and EPs). The last topological transformation, T_5, describes the state of dots. In , there are 5 topological descriptor sequences for each skeleton. It is worth noting that although we call $T_i, i = 1, \cdots, 5$, topological transformation, geometrical features are presented implicitly within them. For example, the definition of above and below for an EP is essentially geometrical. However, as the transformation in the following subsection considers the actual shape of branches between EPs and BPs, we prefer to label it a geometrical transformation.

Although topological descriptors contain a large amount of skeleton information, they are not sufficient for the complete description of a stroke skeleton. Here, an additional transformation, T_6, is introduced and is assigned to each EP based on the geometrical attributes of the skeleton branches, depending if the associated branch to the EP is clockwise or not, if the branch is S-shape or not, and what is vertical location of EP with respect to its corresponding BP. The S-shape state and direction of a branch are determined using numerical fitting of a Bézier curve on the branch [4][2].

In order to have a coherent set of features for all the shapes, a limit, l_{point} on the number of the different points is assumed. In this way, for example, if for a shape there are more than that l_{point} BPs, all BPs after l_{point} are dropped. If the number of points is less than l_{point}, it will be filled by zeros. In this work, we assume $l_{point} = 6$. In this way, 84 skeleton-based features are assigned to each shape. Also, 8 global features are also assigned: vertical center of mass, aspect ratio, height ratio, number of branch points, number of end points, loop feature, horizontal frequency, and dot feature [5]. Therefore, the total number of features assigned to each shape is 92. In the next section, the process of clustering the shapes in a library is discussed.

4.2 Building the library of basis connected components (BCCs)

Based on the aforementioned features, a distance can be calculated between each two CCs [5]. Using this distance function, the library has been built. Now, if we label the BCC (prototype) of each cluster in the library, all CCs attached to that BCC would receive the same label. In Figure 2, a schematic view of the library is shown. As can be seen from the figure, one of the CCs is mis-clustered on a wrong BCC. These mis-clustering errors will be corrected later by human experts. The propagation of labels from the BCCs to the CCs reduces the labor load for labeling, as only manual labeling of the BCCs is required.

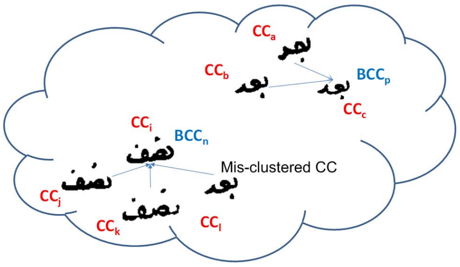


Figure 2: A schematic view of the library showing how CCs are clustered around BCCs.

5. AUTOMATIC LABELING

Having the library at hand, using a user-friendly interface, a label is assigned manually to each BCC. In order to avoid any problem with different platforms, Fingilish encoding[3], which assigns an ASCII code to each Arabic letter, has been used (see Figure 4). The next step, propagating the labels to all CCs has been performed easily based on the assigned labels to BCCs. In this way, the labor load of labeling is reduced by a factor of $1,098/20,722 = 0.053$.

Although the automatically labeled database is ready, it cannot be supplied to most learning methods because of the bi-class requirement in these methods. Considering the high number of classes (there are more than 1000 BCCs) and the

[2] http://matlabdb.mathematik.uni-stuttgart.de/download.jsp?MC_ID=7&SC_ID=8&MP_ID=480

[3] http://fingilish.net/

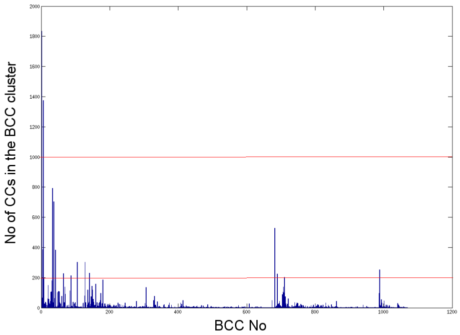
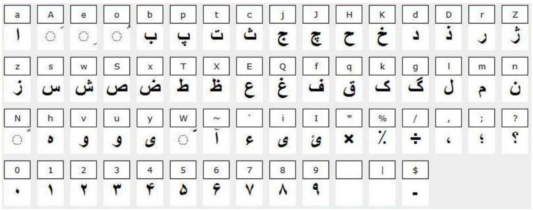


Figure 3: Statistics of the library. Frequency of all BCCs is provided. There are 1,098 BCCs in the library.

lack of correlation between them (because of the variable length of the label strings), the traditional reformulation of the problem into a set of bi-class problems is not straightforward. To break this limit, in the next section, a novel bi-class sub-representation of labels is presented.



a	A	e	o	b	p	t	c	j	J	H	K	d	D	r	Z
ا	◌َ	◌ِ	◌ُ	ب	پ	ت	ث	ج	چ	ح	خ	د	ذ	ر	ژ
z	s	w	S	x	T	X	E	Q	f	q	k	g	l	m	n
ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ک	گ	ل	م	ن
N	h	v	u	y	W	~	`	i	I	*	%	/	,	;	?
◌ً	ه	و	و	ی	◌ّ	آ	ء	ی	ئ	×	٪	÷	،	؛	؟
0	1	2	3	4	5	6	7	8	9		|	$			
۰	۱	۲	۳	۴	۵	۶	۷	۸	۹			ـ			

Figure 4: Fingilish encoding table.[3]

6. CONVERSION OF STRING LABELS TO BI-CLASS PROBLEMS

As discussed above, the direct conversion of string labels in bi-class labels is not obvious. Here, in order to solve this problem, the string labels are first converted in the sets of letters. For example, if the label assigned to a particular CC is "adl", the corresponding set will be {"a", "d", "l"}. In this way, the order within labels is broken. Now, we define bi-class problems as many as the number of letter presented in the string labels. The bi-class problems are shown by P_ω, where ω is one of the letters. For example, for letter "a", we have a bi-class problem $P_{\rm a}$ in which any CC that has this letter in its set of letters is labeled positive $(+1)$, otherwise it is labeled negative (-1). Therefore, the previous example with the string label "adl" will appear as a positive sample for three bi-class problems ($P_{\rm a}$, $P_{\rm d}$, and $P_{\rm l}$), and will show up as a negative sample for the rest of the problems (see Figure 5). This way of re-formulating labels is robust, and is generic and independent of the encoding used and the language. In our case, and using the Fingilish encoding, we have 54 problems (however, some of problems have zero number of positive samples because the number of Arabic letters is less than 54). Figure 6 shows the distribution of the number of the positive sample over the P_ω bi-class problems.

In order to provide statistically meaningful and un-biased problems, only problems with more than 1,000 positive samples are kept. Also, a redundant naming is used for the problems; each problem is named as XY where X is its corresponding letter and Y is either "L" or "U" if the letter is either *lower* or *upper* case, respectively. For example, $P_{\rm a}$ is named as "aL". The statistics of the 12 problems with more than 1,000 positive samples are shown in Figure 7.

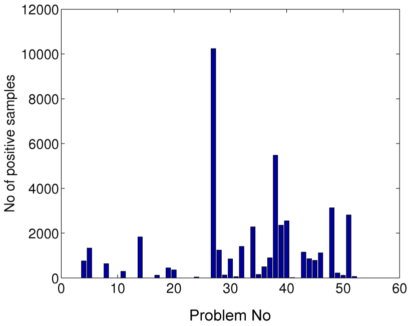


Figure 6: Statistics of the bi-class problems.

In the next section, support vector machines (SVMs) are used to evaluate and validate the automatically generated labels.

7. EVALUATION AND VALIDATION OF THE DATABASE

In order to validate the database, SVMs classifiers are used. SVMs are particular linear classifiers that are based on the margin-maximization principle [15]. They are powerful classifiers that have been used successfully in many pattern recognition problems, and have also been shown to perform well in biometrics recognition applications [1].

Considering a binary classification problem with training data $\{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ where $x_i \in \mathcal{R}^d$ and $y_i \in \{-1, 1\}$, the SVM attempts to find the hyperplane $< w, b >$ that maximizes the margin with minimum error:

$$\min_{w,b,\xi} \frac{1}{2} w' w + C \sum_{i=1}^{\ell} \xi_i \tag{2}$$

$$\text{s.t.} \quad : y_i[w'\phi(x) + b] \geq 1 - \xi_i \quad \forall i = 1, ..., \ell \tag{3}$$

$$\xi_i \geq 0 \quad \forall i = 1, ..., \ell \tag{4}$$

where w' denotes the transpose of w, ϕ is the mapping function used implicitly via the kernel function $k(x_i, x_j) = \phi(x_i).\phi(x_j)$ for non linear problem, C is used to balance the

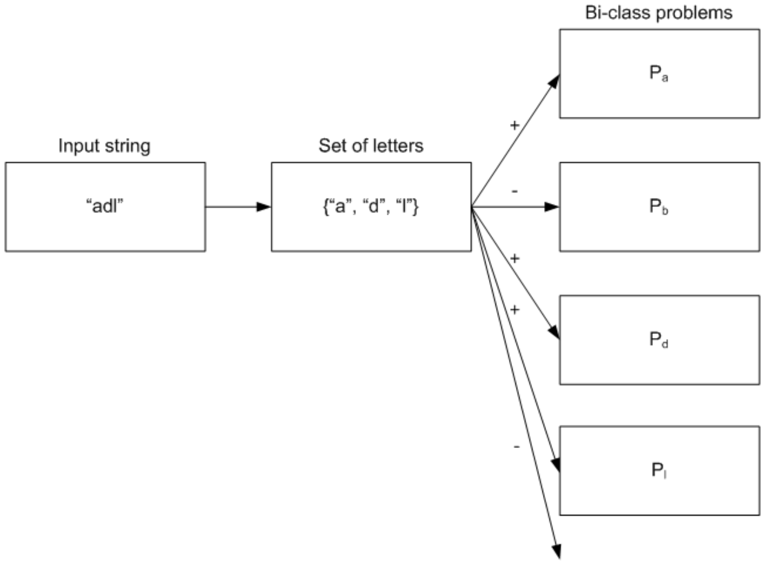


Figure 5: A schematic diagram on how bi-class problems are defined.

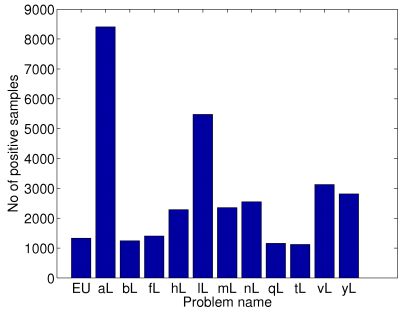


Figure 7: Statistics of the bi-class problems after automatic labeling. Only problems with more than 1000 positive samples are kept.

trade-off between maximizing the margin and minimizing the training error and ξ is the slack variable that quantifies SVM error.

In the primal form, the Lagrangian of this problem is as follows:

$$L = \frac{1}{2}w'w + C\sum_{i=1}^{\ell}\xi_i$$

$$-\sum_{i=1}^{\ell}\alpha_i\{y_i[w'\phi(x)+b] \geq 1-\xi_i\}\sum_{i=1}^{\ell}\lambda_i\xi_i$$

with the Lagrange multipliers $\alpha_i \geq 0$ and $\xi_i \geq 0$ for all i.

After taking the conditions for optimality, we obtain the classifier

$$f(x) = sign[\sum_{i=1}^{\ell}\alpha_i y_i k(x_i, x) + b]$$

where α is the solution of the following optimization problem:

$$\max_{\alpha,b}\sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j k(x_i, x_j)$$

$$s.t.:\sum_{i=1}^{\ell}\alpha_i y_i = 0 \quad and \quad 0 \leq \alpha_i \leq C, i = 1, ..., \ell$$

In our case, the threshold b is computed by minimizing the balanced error rate on the validation set.

7.1 Experiment setup

We used a radial basis function (RBF) kernel $k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$ where x_i and x_j are two typical feature vectors and γ is the kernel parameter. Thus, before building the model, we apply data normalization which is a popular practice in machine learning when RBF kernel is used. This technique puts the variables in a restricted range (with a zero mean and 1 standard deviation).

Since , all the binary problems we have are unbalanced, we used different hyperparameter C for controlling the training error impact, C_+ for positive samples and $C_- = C_+ * n_+ / n_-$ for negative samples where n_+ and n_- represent the number of positive and negative samples [14].

We perform model selection, optimization of the hyperparameter C_+ and kernel parameter γ, on the validation set. The optimal value for γ is 0.02 and for C_+ is 10.

Several experiments have been conducted on each problem in the database. In each experiment, a random fraction (60 percent) of the samples is selected as the training set, and 20 percent is chosen as the validation set. The remaining samples have been used as the test set in each experiment. After performing model selection, we re-train the classifier with the training set combined with the validation set. We repeated this operation 10 times.

7.2 Evaluation method and Results

Because of the unbalanced number of positive and negative samples in each problem, the classic error rate is not a suitable measure. We use the balance error rate (BER), which is the average of the misclassification rates on examples drawn from positive and negative classes. BER is defined as follows:

$$\text{BER} = \frac{1}{2}\left(\frac{\text{FN}}{\text{TP}+\text{FN}} + \frac{\text{FP}}{\text{FP}+\text{TN}}\right)$$

where FN, TP, FP, and TN are *false negative*, *true positive*, *false positive*, and *true negative* respectively.

For each problem and each trial, after we build the SVM model, the BER is calculated on the test samples. Table 1 provides the mean and the standard deviation of the BER for each problem. As can be seen from the table, problem aL, which corresponds to the letter Aleph in the Arabic language, has the lowest BER. Several sources can be considered for the error: ambiguity in the clustering process of the WS method, and human mistakes in the manual labeling stage are two main sources of error at this point. In the next section, the process of verification of the labels by Expert is presented. In this way, we expect that the error rate will reduce, because this process corrects the first source of error (mis-clustering).

8. VERIFICATION OF LABELS BY HUMAN EXPERTS AND RE-VALIDATION

Although automatically labeled CCs are available now, in order to use this database as a ground-truth reference for future research, the labels of all CCs have been verified by human experts. This step has been performed by colleagues from the IIS using a user-friendly interface. In this way, possible errors in automatic labeling are corrected.

The label verification by human experts resulted in there are 15 problems with more than 1,000 positive samples, instead of 12 problems corresponding to the automatically generated labels in section 6, The statistics of the new problems are shown in Figure 8. It shows that the there exists a fraction of mis-clustering in the automatic labeling which is related to ambiguity in the description of shapes. It is confirmed by the increase in the error rate of some of the problems (for example, in the problem bL). Again, the SVM is used to evaluate the database. The BER performance of the SVM is presented in Table 2. As can be seen from the tables, the BER is reduced for almost all the problems. It can be related to the correction of mis-clustering error. However, there are some problems, such as yL, for which the BER is increased. This phenomenon can be explained by the overlapping of the shapes in the feature space caused by the absence of distinctive features (dots) or weakness in the skeleton features.

Problem Name	BER	σ_{BER}
aL	5.4354	0.3005
bL	13.7449	1.1050
EU	13.1266	0.8950
fL	12.1452	0.8876
hL	9.3122	0.5665
lL	9.6199	0.4618
mL	13.8860	0.5710
nL	13.8860	0.5710
qL	10.1014	0.8724
tL	13.9847	1.2674
vL	8.9582	0.4777
yL	13.1692	0.8257

Table 1: The BER and its standard deviation for 12 problems with more than 1000 positive samples before the human expert verification of the labels.

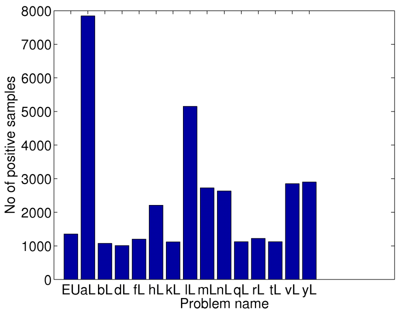


Figure 8: Statistics of the bi-class problems after expert verification.

9. CONCLUSIONS AND FUTURE PROSPECTS

A database of labeled CCs (subwords) of Arabic language has been developed. The data is extracted from an Arabic handwritten manuscript. The extracted shapes have been labeled in two phases, one in an automatic way (using word spotting clustering), and the other by human experts. The

Problem Name	BER	σ_{BER}
EU	10.8930	1.2652
aL	3.8935	0.3077
bL	19.1191	1.6135
dL	10.1375	0.7739
fL	20.7011	1.3613
hL	7.0997	0.3692
kL	13.7881	1.2125
lL	7.4874	0.4518
mL	13.6976	0.6121
nL	16.8097	0.9112
qL	15.6276	0.8917
rL	15.2643	1.1948
tL	19.5261	1.6815
vL	8.6008	1.0094
yL	17.7966	0.7437

Table 2: The BER and its standard deviation for 15 problems with more than 1000 positive samples after the human expert verification of the labels.

labels have been converted into bi-class problems in order to open the door for application of available state-of-the-art bi-class learning techniques (such as SVM classifiers and machines for semi-supervised and active learning, for example, WCCI'10 competition program[4]). The database is evaluated using the SVM classifiers and the BERs are presented before and after human expert verification. The database is unique in terms of the number of subwords (more than 1,000), and the fact that it has been generated form a real manuscript.

In future work, application of the developed models to generate the labels of new shapes (CCs) will be considered. Also, reducing the BER by improving the preprocessing step, and also better skeleton extraction and description are other possible directions.

10. ACKNOWLEDGMENTS

They would like to thank the NSERC of Canada for their financial support.

11. REFERENCES

[1] M. M. Adankon and M. Cheriet. *Encyclopedia of Biometrics*, chapter Support Vector Machine, pages 1303–1308. Springer, 2009.

[2] Y. Al-Ohali, M. Cheriet, and C. Suen. Databases for recognition of handwritten arabic cheques. *Pattern Recognition*, 36(1):111–121, Jan. 2003.

[3] H. Alamri, J. Sadri, C. Suen, and N. Nobile. A novel comprehensive database for Arabic off-line handwriting recognition. In *ICFHR'08*, 2008.

[4] G. Farin. *Curves and surfaces for computer aided geometric design (5th ed.): a practical guide.* Academic Press Professional, Inc., 2001.

[5] R. Farrahi Moghaddam and M. Cheriet. A robust word spotting system for historical arabic manuscripts based on skeleton features. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, Submitted.

[6] R. Farrahi Moghaddam and M. Cheriet. Application of multi-level classifiers and clustering for automatic word-spotting in historical document images. In *ICDAR'09*, pages 511–515, Barcelona, Spain, July 26–29 2009.

[7] R. Farrahi Moghaddam and M. Cheriet. RSLDI: Restoration of single-sided low-quality document images. *Pattern Recognition*, 42:3355–3364, 2009.

[8] A. Gacek. *Arabic Manuscripts: A Vademecum for Readers.* Handbook of Oriental Studies. Section 1 The Near and Middle East, 98. Leiden; Boston: Brill, 2009. ISBN-10: 90 04 17036 7.

[9] J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[10] S. Johansson, G. Leech, and H. Goodluck. Lancaster-oslo/bergen corpus, http://khnt.hit.uib.no/icame/manuals/lob/index.htm. Department of English, University of Oslo, Oslo, 1978.

[11] L. Lorigo and V. Govindaraju. Offline arabic handwriting recognition: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):712–724, 2006.

[12] U.-V. Marti and H. Bunke. A full english sentence database for off-line handwriting recognition. In *ICDAR'99*, pages 705–708, 1999.

[13] U.-V. Marti and H. Bunke. The iam-database: an english sentence database for offline handwriting recognition. *IJDAR*, 5(1):39–46, Nov. 2002.

[14] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring. In *ICML'99*, 1999.

[15] V. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, New York, 1998.

[16] R. Wisnovsky. *Philosophy, Science and Exegesis in Greek, Arabic and Latin Commentaries*, volume 2, chapter The nature and scope of Arabic philosophical commentary in post-classical (ca. 1100-1900 AD) Islamic intellectual history: Some preliminary observations, pages 149–191. Institute of Classical Studies, London, 2004.

[4] http://www.wcci2010.org/;
http://www.causality.inf.ethz.ch/activelearning.php?page=synopsis;
http://www.causality.inf.ethz.ch/al_data/IBN_SINA.html