PERGAMON

Pattern Recognition 36 (2003) 111–121

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Databases for recognition of handwritten Arabic cheques

Yousef Al-Ohali[a,*], Mohamed Cheriet[b], Ching Suen[a]

[a]CENPARMI Computer Science, Concordia University, Suite GM-606, 1455 de Maisonneuve W., Montreal, Que. Canada H3G 1M8
[b]Laboratory for Imagery, Vision and Artificial Intelligence, École de Technologie Supérieure, University of Québec, 1100, Notre-Dame West, Montréal, Qué. Canada H3C 1K3

Received 1 February 2001; accepted 8 March 2002

Abstract

This paper describes an effort towards the development of Arabic cheque databases for research in the recognition of hand-written Arabic cheques. Databases of real-life Arabic legal amounts, Arabic sub-words, courtesy amounts, Indian digits, and Arabic cheques are described. This paper highlights some characteristics of the Arabic language and presents the various steps that have been completed to build these databases including segmentation, binarization and data tagging. It also describes a solid validation procedure including grammars and algorithms used to verify the correctness of the tagging process. Detailed descriptions of the database organization and class distribution are included. These databases aim to facilitate experimental comparisons between various recognition methods, and will be provided to all interested researchers upon request to CENPARMI.[1] © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Arabic OCR; Cheque processing; Image processing; Databases of Arabic handwriting; Database of Indian digits

1. Introduction

Cheque processing involves all the tasks a bank officer may have to do to process an incoming cheque for a client. This includes: accessing account numbers, verifying names and signatures on the cheque, verifying the date of the cheque, matching the legal and courtesy amounts, and verifying the credit of the issuer.

During the past few years, extensive efforts have been devoted to Latin cheque processing systems [1–3,13], including legal amount recognition, courtesy amount recognition, and date recognition. For training and testing purposes, researchers in Ref. [1] have used a private real-life database. Their database could not be provided to the research community due to the confidentiality of information in real-life cheques. On the other hand, researchers in Refs. [2,3] built artificial databases to train and test their systems. A set of 2600 English cheques, written by 800 writers with pre-set legal amounts, has been collected. Another set of 1900 French cheques from 600 different writers has been collected too. The legal amounts were given to the writers to produce a balanced word distribution. These databases are available for the research community.

To the best of our knowledge, no attempts have been reported towards an Arabic cheque processing system. This was partially due to the lack of a supporting infrastructure required to conduct, develop and compare such systems. A major effort to open this area is needed to develop a representative database that can be used for such purposes. This work provides real-life databases for Arabic legal amounts and courtesy amounts (written in Indian digit). It is interesting to note that Indian digits are more popular than Arabic numerals in some parts of the Arabian world.

Arabic has different character set and language rules than Latin languages. Therefore, there is a need to build Arabic databases to train, test and compare recognition systems. Building databases of Arabic cheque processing shares

[*] Corresponding author. Tel.: +1-514 -848-7954; fax: +1-514-848-4522.

E-mail addresses: yousef@cenparmi.concordia.ca (Y. Al-Ohali), cheriet@gpa.etsmtl.ca (M. Cheriet), suen@cenparmi.concordia.ca (C. Suen).

[1] CENPARMI, Concordia University, GM-606, Montreal, Quebec H3E 1M8, Canada.

PII: S0031-3203(02)00064-X

most of the difficulties and challenges with its Latin-based counterparts that are related to the confidentiality of such data. In addition, Arabic has a larger vocabulary for legal amounts than English and French. The large number of secondary components causes more difficulties at the segmentation level. Moreover, grammatical rules for the Arabic legal amounts allow for large variations in writing similar amounts.

In this paper, we describe a set of databases that might be used to train, evaluate and compare Arabic cheque processing systems. These databases comprise real-life data, which make recognition systems more adjustable to real-world applications. In addition, these databases have gone through a solid validation process, which makes erroneous labelling practically impossible.

By choosing to collect real-life data, we eliminated the bias involved in laboratory environments. However, data from real-world environment has its own drawbacks. This is observed from the uneven distribution of sub-word classes in the collected data that may cause some training problems. This is particularly true for those classes that are rarely used. Training and testing of such classes become more difficult and may impose restrictions on the type of classifiers.

To facilitate experimental comparisons among various recognition methods, the databases are divided into training and testing sets. These databases are available to the research community upon request to the Centre for Pattern Recognition and Machine Intelligence (CENPARMI) (see footnote 1).

The rest of this paper is organized as follows. After this introduction, we give a brief description of the Arabic written language focusing on the current application, and mentioning the major characteristics that may affect a pattern recognition system. Data collection is covered next, followed by the pre-processing section. Tagging comes next, followed by validation, and databases that resulted from this project. After that, we discuss the impact of this work on existing and future recognition systems. We finish this paper with some concluding remarks.

2. Brief description of the Arabic written legal and courtesy amounts

Arabic is used in more than 20 countries by more than 200 million people. Unlike Latin, Arabic is written from right to left in cursive script. Out of the 28 basic Arabic letters, 22 are cursive letters while 6 are non-cursive. Within one word, a cursive letter should be connected to the succeeding letter, while non-cursive letter cannot be connected to any succeeding letter. Thus, an Arabic word may be decomposed into more than one sub-word, each represents one or more connected letters with their corresponding secondary components [4,5].

In addition, Arabic defines two types of secondary components: dot and Hamzah (a zigzag-like shape). The number and position of secondary components play a factor in identifying different letters. Moreover, Arabic allows the presence of diacritics that control the pronunciation of words and possibly their meanings. However, such diacritics are only used in very formal documents or in cases of contextual ambiguity [4,5].

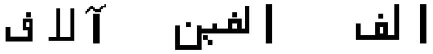


Fig. 1. Singular, double and plural forms for the word “thousand”.

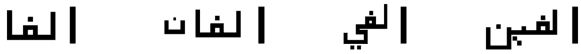


Fig. 2. Four grammatical forms of the word (two-thousand).

Fig. 3. Feminine and masculine forms of the word “three”.

Fig. 4. Two common forms for the world “hundred”.

ثلاثه
ثلاثة

Fig. 5. Secondary components of the last letter may be ignored, a common mistake.

Due to connectivity, the shape of an Arabic letter may change significantly depending on its position within a sub-word, identity of neighbouring letters, the writing font, and the way the writer connects successive letters [4,5]. Arabic handwritten letters differ in height and width.

The vocabulary of Arabic legal amounts is larger than those found in Latin languages. This is due to three major factors. First, Arabic has three different forms: singular, double, and plural (Fig. 1). This could affect both the number and the counted item (currency words in this context). Second, double and plural nouns have up to four different forms according to their grammatical positions (Fig. 2). Third, most numbers define two forms for feminine and masculine countable things (Fig. 3).

In addition, a few common spelling mistakes and/or colloquials occur in writing some Arabic numbers (Figs. 4 and 5). These factors affect the identity of letters and the number of sub-words composing a word.

We found more different words than sub-words in the Arabic legal amount lexicon. That was one of the reasons to consider sub-word as the basic unit of Arabic legal amounts.

In principle, Arabic allows legal amounts to be written in any order, i.e. starting from the most significant digit, from the least significant digit or even from the middle. However,

Arabic digits	Corresponding Indian digits
0	٠
1	١
2	٢
3	٣
4	٤
5	٥
6	٦
7	٧
8	٨
9	٩

Fig. 6. Arabic digits and their corresponding Indian digits.

eloquence measurements and personal habits excluded most permutations.

Arabic imposes certain constraints on numerical values. For instance, the word hundred does not appear with numbers more than 9, i.e. values larger than 999 should be written in the forms of the word thousand. In addition, since Arabic defines three different categories for single, double and plurals, there are normally different constraints for the numbers one and two of each numerical category $(1, 2, 100, 200\ldots)$.

With respect to courtesy amounts, there are no language constraints. Courtesy amounts are written in either Arabic or Indian digits. Fig. 6 shows the 10 Arabic digits with the corresponding Indian ones. Note that the zero is written as a dot. The decimal point is written as a large comma. The amount delimiters (dollar signs) are optional and no standard symbol is used. Whether Arabic or Indian digits are used, courtesy amounts are written in the same order, putting the least significant digit in the rightmost position.

3. Data collection

The first step toward developing a database is to find suitable sources of data. Finding a real-world source of data becomes a problem when dealing with applications that carry sensitive or private information like bank cheques. Through collaboration with Al Rajhi Banking and Investment Corporation (one of the largest banking corporations in Saudi Arabia), we were able to collect about 7000 real-world grey-level cheque images. The gathering process involved scanning the real cheques at the bank's centre, and removing all personal (private) information including names, account numbers, and signatures. The cheques were scanned in grey level at 300 dpi, and were used as the core of other databases throughout this project.

Cheques differ in the amount of external noise added by bank officers. Some cheques contained a stamp covering part of the legal amount, making the automated extraction and cleaning processes more difficult. We decided to collect two sets of data: random data and selected data. The first set contains cheques that are randomly taken from the available samples (4000 cheques), while the selected data contains only those cheques that have no stamps on top of their legal amounts (3000). For database development, we decided to start with the selected data set, which is the focus of this paper, leaving processing of the random data set for future work.

4. Pre-processing

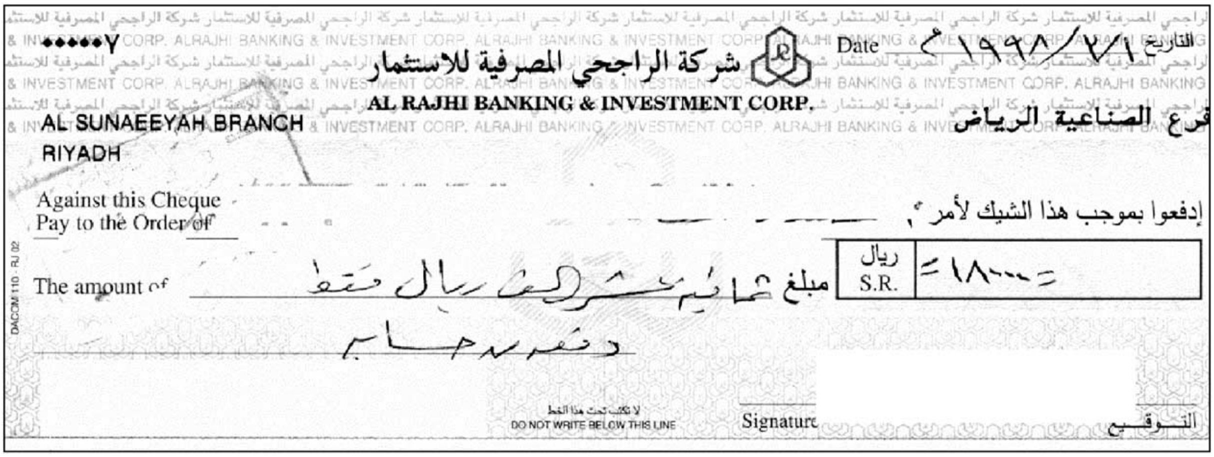
The next step was to extract fields of interest from cheque images. We concentrated on the legal and courtesy amounts, leaving the date field for future work. That was achieved by localizing the target fields on all kinds of cheque forms. In Saudi Arabia, there were only two types of cheques, which share the same format (structure) but have different sizes, and inter-field distances.

The next pre-processing step was to binarise and minimise existing noises from the segmented fields. Such noises could result from the digitization process, from the extraction process, by bank officers or by the client. The noises include lines, borders, and pre-printed texts that may appear along with the extracted fields. This step has been successfully achieved by adapting the tools available at CENPARMI (which were designed for Canadian cheques) [6,7]. Figs. 7–9 show a sample Arabic cheque and its corresponding segmented legal and courtesy amounts.

5. Tagging

Tagging intends to label each object in the cleaned legal and courtesy amounts. A tagging tool has been prepared to tag Arabic sub-words and numerals. The tagging operator is required to click at any point on each connected component of the target object and then select a tag from a pre-defined vocabulary. The tool stores the coordinates of each point, adds a delimiter and stores the tag. It allows tagging of touching objects and permits reverse action (undo).

While defining our vocabulary, we have accounted for most differences, even small ones. For instance, two different tags were used to label objects that differ only in their secondary components (Fig. 5). This gives more choices for future analysis since there is no cost to merge similar sub-classes (if such discrimination is not useful for a particular method or application). We limited the vocabulary to amounts that are less than one million. Larger amounts will most likely require manual manipulation by bank officer(s). In addition, we included in the lexicon currency and



●●●●●٧

شركة الراجحي المصرفية للاستثمار

AL RAJHI BANKING & INVESTMENT CORP.

Date ١٩٩٨/٧/١ التاريخ

AL SUNAEEYAH BRANCH
RIYADH

فرع الصناعية الرياض

Against this Cheque
Pay to the Order of

إدفعوا بموجب هذا الشيك لأمر

The amount of ثمانية عشر ألف ريال فقط مبلغ

ريال
S.R.

=١٨٠٠٠=

لا تكتب تحت هذا الخط
DO NOT WRITE BELOW THIS LINE

Signature التوقيع

Fig. 7. A sample of the Arabic cheque database.

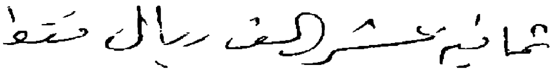


Fig. 8. Segmented legal amount (from Fig. 7).

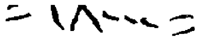


Fig. 9. Segmented courtesy amount (from Fig. 7).

other words that are frequently used in the collected data to delimit legal amounts (e.g. only).

For each cheque, the tagging tool produces four sets of tagged objects:

1. Courtesy amount: contains a sequence of coordinates and tags of objects. Objects may include Indian digits, delimiters, commas, decimal points or noise. Coordinates provide an unambiguous pointer to the object intended by each tag.
2. Indian digit: contains a reference to the original courtesy amount that produced the current object, followed by the tag of the current object.
3. Legal amount: contains a sequence of coordinates and tags of objects. Objects may include sub-words, or noise. Coordinates provide an unambiguous pointer to the object intended by each tag.
4. Arabic sub-word: contains a reference to the original legal amount that produced the current object, followed by the tag of the current object.

File naming of the courtesy and legal amounts was made to refer to the original cheque number from which these amounts were extracted. Sub-word and digit file naming were chosen in a sequential manner.

Tagging of legal and courtesy amounts were done independently. This was intentionally made to minimise chances of complex human errors that may happen to both the legal and courtesy amounts of the same cheque.

6. Validation

Although the tagging tool was designed to prevent or warn for possible errors in the tagging process, yet there are still some traces of mistakes. This is particularly true when dealing with large amounts of data. Therefore, a procedure to verify the truthfulness of the tagging process is needed. In the following two sections, we describe two procedures that were developed for this purpose.

6.1. *Automatic validation procedure*

We took an early advantage of the redundancy available in the cheque by comparing numerical values of the tags of legal and courtesy amounts of each cheque. A tagged cheque is approved (and all objects obtained from it) if the numerical values obtained from the tags of its legal and courtesy amounts match. Otherwise, further steps need to be taken to validate or correct the tagged legal and/or courtesy amounts.

Comparing the two amounts requires translation and interpretation of each sequence of tags into its numerical value. While this looks trivial for courtesy amounts, it involves a complex process in the case of legal amounts. First, each tag should be translated into the appropriate sub-word. Second, each proper sequence of sub-words needs to be translated into a correct word. This requires special attention since some words may appear as sub-words of a larger word (Fig. 10). This was achieved by means of a context-sensitive grammar developed for this purpose. Third, the sequence of words should be interpreted into a numerical value. Again, this requires special manipulation since there are various

Fig. 10. The word ten (right) appears as a sub-word in the word twenty.

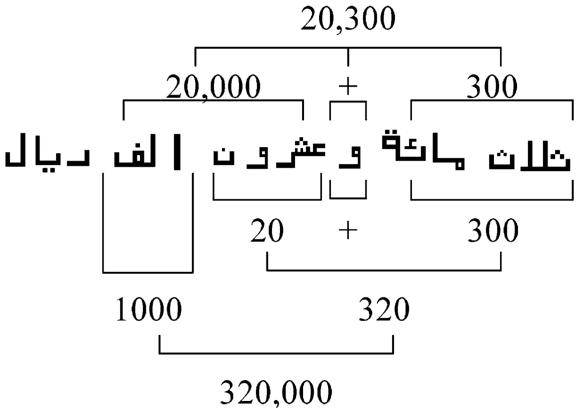


Fig. 11. One Arabic amount with two possible interpretations.

orders to write an amount in Arabic (e.g. from high order to low order).

Definition 1. A language is a group of meaningful sentences that share similar structures.

Definition 2. A grammar is a formal specification of structures allowable by a given language.

A grammar G is completely specified by four components: $\{N, T, S, P\}$ where: N is the set of non-terminal symbols, T the set of terminal symbols, S the starting symbol ($S \in N$), and P a set of production rules $A \rightarrow B$, where $A \cap N \neq \phi$, and $B \in T \cup N$, which is the set of all symbols in the grammar [14].

Definition 3. A language L is said to be ambiguous if it allows a sentence $S \in L$ to have more than one interpretation.

Lemma 1. *The language of Arabic legal amounts is ambiguous.*

Proof. To prove Lemma 1, we need to present a single example of such ambiguity. Fig. 11 shows a legal amount with two grammatically correct interpretations. □

Definition 4. A grammar G for a language L is said to be general if and only if

1. G accepts all possible sentences of L, and
2. G provides all possible interpretations of S, $\forall S \in L$.

Grammars provide a formal way to specify and interpret languages. When defining a grammar for a given language, it is essential to define a general grammar. At the same time, it is important to define a selective grammar, one that rejects most ill-formed sentences.

1. Extract the sequence of sub-words' tags.
2. Form all correct sequences of words (using the context sensitive grammer).
3. Remove words that have no numerical value (e.g. only)
4. Find all possible parse trees for the legal amount.
5. For each term (a term contains value-word(s) followed by a unit), look for embedded terms. If any, then perform steps 5-6 to each of them.
6. Replace words (terms) by values, and evaluate the term by multiplying the term value by the term unit.
7. Sum up values of all terms.

Fig. 12. Algorithm used to translate Arabic legal amount into a numerical value.

Definition 5. An ambiguous grammar is one that allows more than one parse tree for at least one sentence.

Lemma 2. *To generally define an ambiguous language, an ambiguous grammar is required.*

Proof.

1. Assume that L is an ambiguous language, $S \in L$, and S has two different interpretations: $S \Rightarrow I1$ and $S \Rightarrow I2$.
2. Assume that G is a grammar that provides a general specification for L.
3. Since $S \Rightarrow I1$, and since G is general, there is a parse tree T1 derived by G, that leads to I1.
4. Since $S \Rightarrow I2$, and since G is general, there is a parse tree T2 derived by G, that leads to I2.
5. From 3 and 4 above, we see that G provides two parse trees T1 and T2 for the same sentence S, which implies that G is an ambiguous grammar. □

Due to ambiguity of Arabic legal amounts, a general grammar is required to allow for all possible interpretations for any given sentence. In our validation procedure, a tagged cheque is approved if any of the generated legal-amount numerical values match the value generated by the tagged courtesy amount. To facilitate the translation of legal amounts into numerical amounts, each grammatical symbol is assigned a numerical value. Fig. 12 shows the algorithm used to translate Arabic legal amounts into numerical values.

In the following we illustrate the grammar used to parse Arabic legal amounts. The set of terminals T is clearly chosen as the words that compose the Arabic legal amount vocabulary, i.e. words that represent numbers, currency words and words that are frequently used to delimit legal amounts. In Arabic, as in some other languages, language rules insist different constraints for various groups of numerical words. For instance, the number 200 should be written as a single word, rather than being composed of the number two and the number hundred. This is not the case with numbers from 300 to 900. Due to this reason, different non-terminals are used to produce each group of terminals. This is shown in P$_1$–P$_3$ below.

P$_1$: NumWord → any word that has a numerical value above 0.
P$_2$: FewNumWord → words that have values between 3 and 9.
P$_3$: LessThanTenWord → words with values between 1 and 9.

Currency and unit words have various grammatical constraints too. P$_4$–P$_8$ are defined to produce each group of unit words.

P$_4$: Unit_1 → 1 ريالات ريال
P$_5$: Unit_2 → 10 عشر
P$_6$: Unit_3 → 100 مائة
P$_7$: Unit_4 → 1000 آلاف ألف
P$_8$: Unit_5 → 0.01 هللات هللة

An Arabic legal amount could be seen as a list of terms connected to each other by a connector (and, و) as shown in P$_9$ below. Each term contains a number phrase followed by a unit, or simply a unit (P$_{10}$, P$_{11}$, P$_{12}$, and P$_{13}$). A number phrase can be a simple number, a complete term, or two numbers connected together (P$_{14}$).

P$_9$: S → Term | Term Connector Term.
P$_{10}$: Term → Term3 | Term4.
P$_{11}$: Term3 → Unit1 | Unit5 | Term5.
P$_{12}$: Term4 → NumPhrase Unit1 | NumPhrase Unit4.
P$_{13}$: Term5 → FewNumWord Unit3 | LessThanTenNum– Word Unit2.
P$_{14}$: NumPhrase → NumWord | Term5 | NumPhrase Connector NumPhrase.

6.2. Human validation procedure

For the set of cheques that could not be validated automatically (i.e. rejected by the grammar), we have designed an interface to facilitate the manual validation process. For each legal amount, the operator may take one of the following two decisions:

1. Mark the legal amount to be re-tagged.
2. Reject the legal amount.

The same procedure is performed for rejected courtesy amounts. Marked amounts are fed back to the tagging tool, and then to the validation module.

6.3. Discussion

Theoretically speaking, our validation process may oversee some tagging mistakes. However, this is extremely rare in practice. We may make a point by describing what it takes to approve incorrect tagged cheque C:

1. An error would occur while tagging the legal amount of C.
2. This error should create a different, yet correct sequence of sub-words that makes a word.
3. The new sequence of words should be grammatically correct to generate a corresponding numerical value.
4. An equivalent numerical error should have occurred while tagging the courtesy amount of C, to make it equal to the one generated from the legal amount.

It is clear that such a sequence of events is hardly expected for a single cheque. Note that legal and courtesy amounts are tagged independently at different instances.

A question may arise about the set of cheques that could not be approved. Following are some of the reasons that led most unapproved-tagged cheques to this category:

1. The extraction tool may have cut the legal amount (or the courtesy amount), providing incomplete or incorrect data to the tagging tool.
2. The legal amount may have contained an unexpected spelling mistake that left the relevant sub-word untagged (tagged as OTHER symbol), leaving a gap in the legal amount.
3. The legal amount may contain a word that is out of the range covered by this study (e.g. million).
4. There may be missing sub-words (mainly letters) in the original legal amount.
5. The tagging operator may have produced some error.

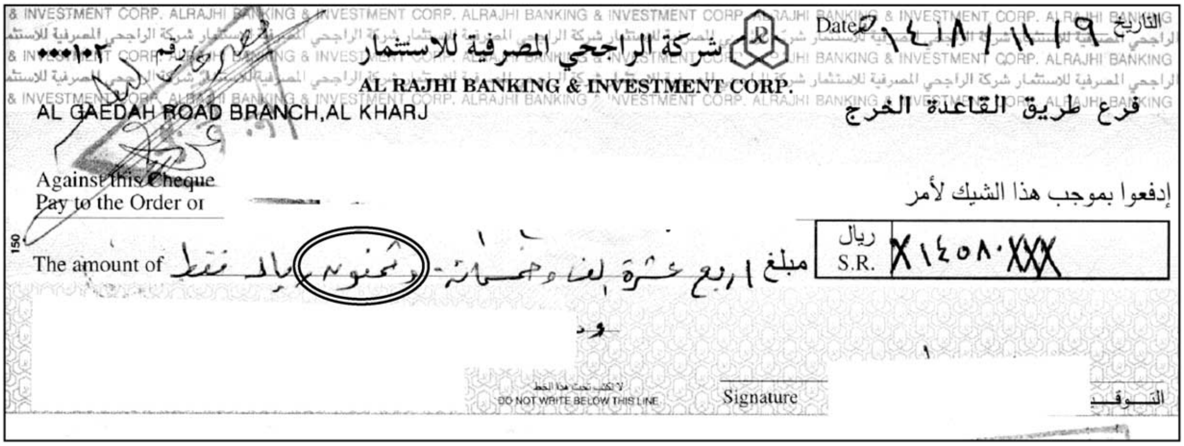
Fig. 13 shows an example of rejected cheques.

6.4. Validation results

The complete validation process approved about 83% of the 3000 tagged cheques, which provided about 29,498 sub-words and 15,175 digits. Tables 1 and 2 show the distribution of the validated sub-word/digit classes (excluding touching sub-words and touching digits). Some classes are very rare, though they do exist in the lexicon of handwritten Arabic legal amounts. Such classes should remain in the lexicon although they are not very well represented. It is important to note that this validation process guarantees the correctness of the tagged legal/courtesy amounts, and all Indian digits. Fig. 14 shows the structure of the validation module.

7. Databases

This research effort has produced a number of databases that can help researchers in various fields. These databases include Arabic legal-amounts database (2499 legal amounts), Courtesy amounts database (2499 courtesy



رقم ١٠٢

شركة الراجحي المصرفية للاستثمار
AL RAJHI BANKING & INVESTMENT CORP.

Date ١٤١٨/١١/١٩ التاريخ

AL GAEDAH ROAD BRANCH,AL KHARJ فرع طريق القاعدة الخرج

Against this Cheque
Pay to the Order of إدفعوا بموجب هذا الشيك لأمر

The amount of فقط ريال ثمنونه وخمسمائة الف اربع عشرة مبلغ

ريال S.R. X١٤٥٨٠XXX

لا تكتب تحت هذا الخط
DO NOT WRITE BELOW THIS LINE

Signature التوقيع

Fig. 13. Sample of the rejected cheques due to spelling mistake.

Table 1
Distribution of the digits data set

Digit	No. of samples
0	5367
1	1086
2	770
3	506
4	440
5	912
6	390
7	342
8	344
9	268
Delimiter	4088
Comma	347
Total	14,860

amounts written in Indian digits), Arabic sub-words database (29,498 sub-words within the domain of legal amount), and Indian digits database (15,175 digits). Each database mentioned above is divided into training and testing sets. The training set includes 66–75% of the available data. That is true for legal amounts, courtesy amounts, Indian digit classes and most sub-word classes. In few sub-word classes, this condition could not be satisfied due to insufficient samples. This ratio was chosen to provide enough training samples on one hand, and to give enough measurement of the generality of the recognition systems on the other hand. The division between training and testing data was done randomly with restrictions to ensure the ratio mentioned above.

Training and testing data sets are further divided into two sets: touching amounts and non-touching amounts. A courtesy amount is located in the touching set if it contains at least one pair of touching components. The same can be said about the database of legal amounts.

Data sets of the non-touching Indian digits and the sub-words are further divided based on their class (i.e. each class is located in a separate directory). The number of classes defined for Indian digits and Arabic sub-words were 11 and 87.

Training data of the legal amounts, courtesy amounts, Indian digits and Arabic sub-words databases are extracted from the same set of cheques. Table 3 shows the sizes of training and testing sets in each of the above four databases. The above-mentioned databases are all available in tiff format. Fig. 15 shows the structures and inter-relation between the courtesy amount database and the digits database.

Moreover, this work produced a database of complete (original) grey level cheques, which can be used for other research purposes (e.g. date processing). These databases are available to researchers through CENPARMI. In addition, it is not difficult to derive a database of Arabic words. This is achievable using the legal amounts database or the sub-words database. It is also possible to generate a database of Arabic dates from the Arabic cheques database.

8. Impact of this work on recognition systems

Many approaches have been used toward the problem of Arabic character recognition. Authors in Ref. [8] extract 9 moments of the horizontal and vertical projections from the main components of the input character. A quadratic discriminate function is proposed for the classification task. To train and test their system, a database of 50 handwritten samples was used. Reported classification rate is 99.5%. In Ref. [9], a back propagation NN with one hidden layer was used to recognize Arabic cursive words. The input

Table 2
Distribution of the sub-words data set

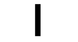
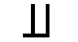
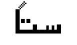
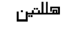
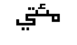
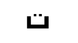
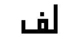
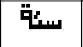
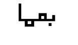
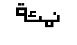
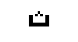
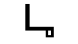
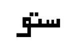
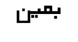
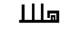
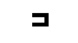
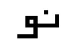
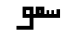
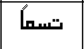
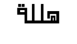
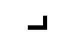
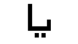
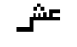
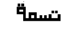
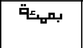
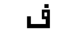
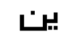
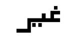
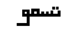
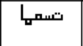
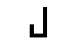
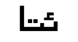
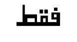
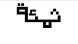
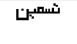
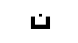
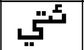
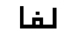
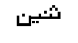
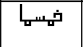
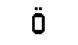
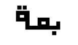
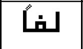
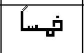
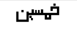
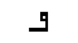
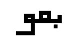
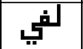
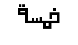
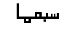
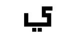
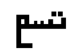
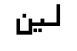
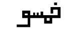
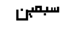
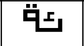
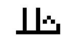
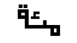
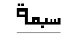
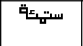
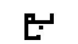
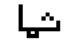
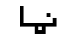
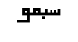
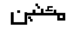
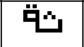
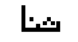
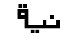
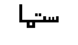
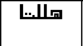
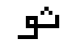
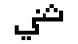
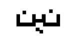
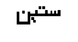
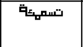
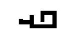
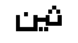
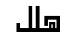
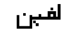
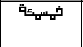
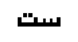
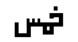
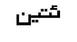
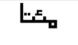
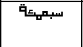
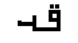
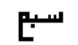
Code	Image	Count	Code	Image	Count	Code	Image	Count	Code	Image	Count	Code	Image	Count
1-00		2822	2-09		2450	3-16		1	6-06		0	4-23		0
1-05		6	2-10		1049	3-17		160	4-01		118	4-24		1
1-06		14	2-11		299	3-19		89	4-02		8	4-26		0
1-07		32	2-12		94	3-20		34	4-03		0	4-27		6
1-08		2920	2-13		1830	3-21		671	4-04		90	5-00		1
1-09		811	2-14		16	3-22		1519	4-06		58	5-02		116
1-10		1665	3-00		71	3-23		1415	4-07		0	5-03		1
1-11		1357	3-02		4	3-24		179	4-09		18	5-04		263
1-12		345	3-04		222	3-25		17	4-10		1	5-05		14
1-14		2677	3-06		117	3-27		3	4-11		392	5-06		118
1-16		53	3-07		5	3-28		1	4-13		213	5-07		1
2-00		1152	3-08		509	3-29		14	4-15		121	5-08		1
2-02		9	3-09		440	3-31		99	4-17		102	5-10		4
2-03		269	3-10		77	3-33		132	4-18		132	5-11		0
2-05		128	3-12		20	3-35		2	4-19		5	6-00		1
2-06		93	3-13		6	3-36		2	4-20		89	6-02		1
2-07		3	3-14		15	4-00		31	4-21		48	6-04		1
2-08		111	3-15		2									

image is first segmented into letters, which are then represented by a list of statistical features. The system was trained and tested on 50 sample words written by 25 different writers, and produced 98% correct recognition rate. In Ref. [10], Fourier spectra of vertical and horizontal projections from printed Arabic characters are estimated. A minimum distance classifier is then used to map the extracted features to one of 10 sets of characters. The results have shown 99.94% recognition rate using a dataset of 49 printed samples per character. In a more recent work, authors in Ref. [11] proposed a recognition-based segmentation technique. Features are extracted from each character fragment and classified using a string-matching technique. A feedback signal is issued if a character fragment is rejected by the classifier. As a result, the succeeding fragment is attached to the rejected one and the recognition process is invoked again. This system was tested using articles scanned from an Arabic book, and yielded a 90% recognition accuracy. For more methods about segmentation, feature extraction, and classification of handwritten characters, we refer the reader to Ref. [12]. Excellent surveys about Arabic character recognition methods are found in Refs. [4,5].

The above analysis of previous works in this field suggests that we address the effect of training and testing data on the performance of recognition systems. Training any recognition system on a small database certainly minimizes its performance and hides some of its strengths. Larger databases are expected to expose recognition systems to more learning patterns that could exhibit their discriminatory power. In addition, an independent realistic database avoids the bias that is often included in laboratory collected databases. Thus making proposed recognition methods more adjustable to real-world applications. Moreover, standard databases facilitate experimental comparisons among various recognition methods.

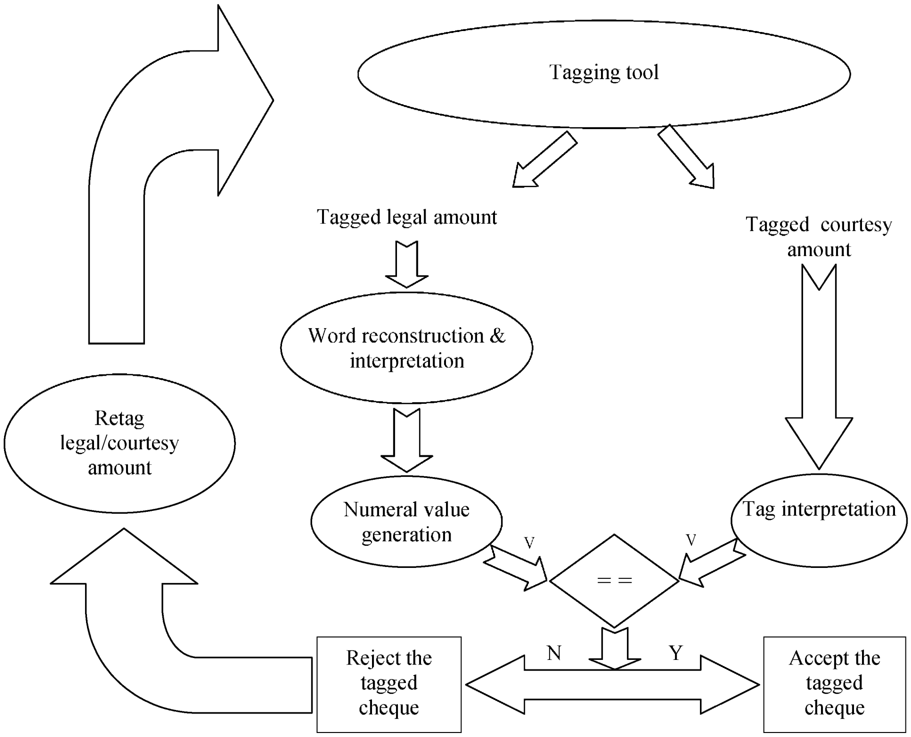


Fig. 14. The validation process.

Table 3
Distribution of databases between training and testing sets

	Sub-words (no. of samples)	Digits (no. of samples)	Legal amounts (no. of cheques)	Courtesy amount (no. of cheques)
Training				
Touching	1066	243	838	266
Not touching	19,813	10,536	941	1513
Total	20,879	10,779	1779	1779
Testing				
Touching	447	72	321	94
Not touching	8172	4324	399	626
Total	8619	4396	720	720

The current work is complementary to other research efforts that focus on the segmentation, feature extraction and/or recognition parts of Arabic pattern recognition systems. Through this effort, we hope to establish a basis for a quantitative evaluation scheme among recognition methods that target Arabic legal amounts.

9. Conclusion

A substantial amount of effort has been devoted toward building Arabic cheque databases, a very important infrastructure to develop and compare pattern recognition systems for the Arabic-based cheque-processing systems. This paper describes the main steps that have been completed to develop such databases. The paper also gives a list of useful databases that have been produced from the first batch of 3000 cheques. These databases can be obtained by a request to CENPARMI. In the future, we will work to complete the remaining 4000 cheques.

10. Summary

This paper provides a detailed description of newly developed databases to assist researchers in the field of handwritten Arabic legal and courtesy amounts recognition. It describes databases for Arabic cheques, Arabic legal amounts, Arabic sub-words, courtesy amounts written in Indian digits, and a database for Indian digits. Among the unique characteristics of these databases are (i) items were extracted from real-life data, (ii) tags of objects went through a solid validation procedure that uses dual-redundant information.

The databases were extracted from 3000 real-life cheques out of 7000 cheques collected from a financial institution. The remaining collected cheques require additional processing and are left for future research work. The resulted databases include 29,498 samples in the Arabic sub-words

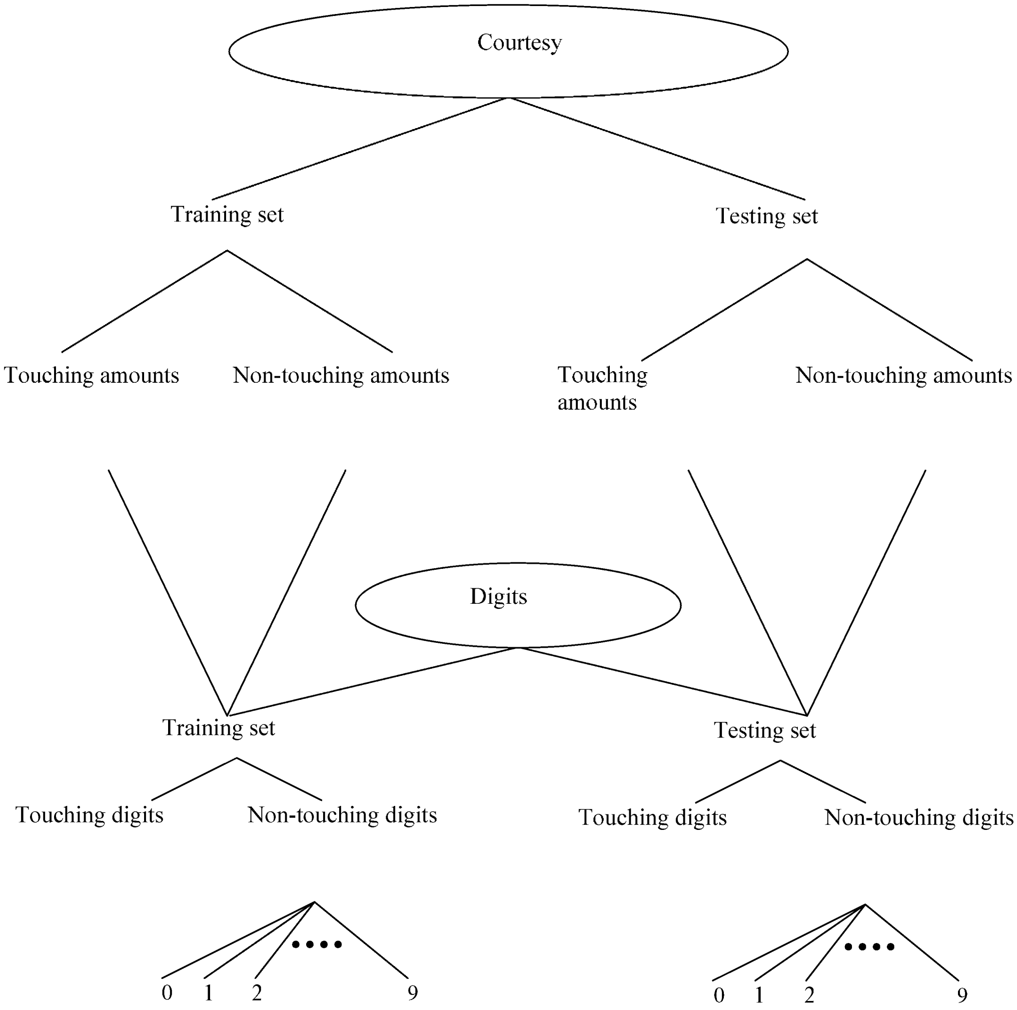


Fig. 15. Structure of the courtesy amount and digits databases.

database, 15,175 samples in the Indian digits database and 2499 samples of each of legal and courtesy amounts.

The tag validation procedure works with tagged legal and courtesy amounts. The tags of a legal amount is interpreted and translated into numerical value, which is then compared with the corresponding courtesy amount tags. Any error in the tagging process will mostly lead to un-interpretable sequence of tags, or in the worst case, will lead to mismatch in the values obtained.

The paper includes a detailed description of all the steps that were conducted throughout the development process, including the following:

(i) gathering of real-life data (cheques),
(ii) segmentation and binarization,
(iii) tagging,
(iv) validation of the tagging process, and
(v) distribution and structure of various databases.

The paper also includes a description of the major characteristics of Arabic legal and courtesy amounts with deep analysis of the linguistic and grammatical rules that affect the writing of legal amounts and their interpretations.

In addition, grammar and algorithm to parse and translate Arabic legal amounts into numerical values are included.

The databases described in this paper are available to interested researchers through CENPARMI (see footnote 1).

Acknowledgements

Our special thanks are due to Al Rajhi Banking and Investment Corp. for their understanding of our research goals and for the facilities they have provided to us.

Our thanks are passed to Dr. Hatim Aboulsamh, dean of the College of Computer and Information Sciences (CCIS) at King Saud University, to the vice-dean Dr. Mohammed

Al-Aqeel, and to the director of the research centre at CCIS, Dr. Khaled Al-Ghoneim for their help and support. Dr. Badr Al-Badr from King Abdulaziz City for Science and Technology (KACST), Riyadh has a shiny supportive touch to this work. The Saudi Cultural Bureau in Ottawa has been very co-operative and supportive of this work too.

Thanks are also due to the Natural Sciences and Engineering Research Council of Canada, and the FCAR Program of the Ministry of Education of Quebec, for their support of this research project.

Our thanks are also extended to Christine P. Nadal, at CENPARMI, for her continuous assistance in this project.

References

[1] M. Gilloux, M. Leroux, Recognition of cursive script amounts on postal cheques, Proceedings of the European Conference Dedicated to Postal Technologies, Nantes, France, June 1993, pp. 705–712.

[2] D. Guillevic, C.Y. Suen, Recognition of legal amounts on bank cheques, Pattern Anal. Appl. 1 (1) (1998) 28–41.

[3] C. Suen, L. Lam, D. Guillevic, N. Strathy, M. Cheriet, J. Said, R. Fan, Bank check processing system, Int. J. Imag. Syst. Technol. 7 (1996) 392–403.

[4] B. Al-Badr, S. Mahmoud, Survey and bibliography of Arabic optical text recognition, Signal Process. 41 (1995) 49–77.

[5] A. Amin, Off-line Arabic character recognition: the state of the art, Pattern Recognition 31 (5) (1998) 517–529.

[6] X. Ye, M. Cheriet, C.Y. Suen, K. Liu, Extraction of bankcheck items by mathematical morphology, Int. J. Document Anal. Recognition 2 (3) (1999) 53–66.

[7] X. Ye, M. Cheriet, C.Y. Suen, Model-based character extraction from complex backgrounds, Proceedings of the ICDAR99, 1999, pp. 511–514.

[8] H. Al-Yousefi, S. Udpa, Recognition of Arabic characters, IEEE Trans. Pattern Anal. Mach. Intell. 14 (8) (1992) 853–857.

[9] M. Altuwaijri, A parallel recognition system for Arabic cursive words with neural learning capabilities, Ph.D Thesis, The University of Louisiana, 1995.

[10] S. Alshebeili, A. Nabawi, S. Mohmoud, Arabic character recognition using 1-D slices of the character spectrum, Signal Process. 56 (1997) 59–75.

[11] A. Cheung, M. Bennamoun, N. Bergmann, An Arabic optical character recognition system using recognition-based segmentation, Pattern Recognition 34 (2001) 215–233.

[12] P. Wang, Character and Handwritten Recognition: Expanding Frontiers, World Scientific Publishers, Singapore, 1991.

[13] J.J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554.

[14] J. Allen, Natural Language Understanding, The Benjamin/Cummings Publishing Company, Inc., New York, 1995.

About the Author—YOUSEF AL-OHALI received his Bachelor and Master degrees in Computer Science from King Saud University (KSU), Riyadh, Saudi Arabia in 1990 and 1995. He is currently a Ph.D. candidate under the supervision of Dr. Suen and Dr. Cheriet at the Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, Montreal, Canada. His research interests include printed/handwritten character recognition, pattern recognition, and image processing.

About the Author—MOHAMED CHERIET received his B. Eng. degree in Computer Science from Université des Sciences et de Technologie d'Alger (Algiers) in 1984, and received his M.Sc. and Ph.D. degrees, also in Computer Science, from University of Pierre et Marie Curie (Paris VI) in 1985 and 1988, respectively. Dr. Cheriet was appointed Assistant Professor in 1992, Associate Professor in 1995, and Full Professor in 1998 in the Department of Automation Engineering, École de technologie superieure (ETS) of University of Quebec, in Montreal. Currently, he is the Director of LIVIA, the Laboratory for Imagery, Vision and Artificial Intelligence at ETS, and an active member of CENPARMI, the Centre for Pattern Recognition and Machine Intelligence. Prof. Cheriet's research focuses on image processing, pattern recognition, character recognition, text processing, documents analysis and recognition, and perception. He has published more than 90 technical papers in the field. He was a guest Editor of the International Journal of Pattern Recognition and Artificial Intelligence and the Machine, Perception, and Artificial Intelligence series books, published by World Scientific. He was the Co-Chair of the 11th and the 13th Vision Interface Conferences held respectively in Vancouver, in 1998 and in Montreal, in 2000. He is currently the General Co-Chair of the 8th International Workshop on Frontiers on Handwriting Recognition, to be held in Niagara-on-the-lake in 2002. Dr. Cheriet is a senior member of IEEE.

About the Author—CHING Y. SUEN received an M.Sc. (Eng.) degree from the University of Hong Kong and a Ph.D. degree from the University of British Columbia, Canada. In 1972, he joined the Department of Computer Science of Concordia University where he became Professor in 1979 and served as Chairman from 1980 to 1984, and as Associate Dean for Research of the Faculty of Engineering and Computer Science from 1993 to 1997. Currently he holds the distinguished Concordia Research Chair of Artificial Intelligence and Pattern Recognition, and is the Director of CENPARMI, the Centre for PR & MI. Prof. Suen is the author/editor of 11 books and more than 300 papers on subjects ranging from computer vision and handwriting recognition, to expert systems and computational linguistics. He is the founder and Editor-in-Chief of a journal and an Associate Editor of several journals related to pattern recognition. A Fellow of the IEEE, IAPR, and the Academy of Sciences of the Royal Society of Canada, he has served several professional societies as President, Vice-President, or Governor. He is also the founder and chair of several conference series including ICDAR, IWFHR, and VI. Currently he is the General Chair of the International Conference on Pattern Recognition to be held in Quebec City in August 2002. Dr. Suen is the recipient of several awards, including the ITAC/NSERC Award in 1992 and the Concordia "Research Fellow" award in 1998.