

20 Bank Check Data Mining: Integrated Check Recognition Technologies

Nikolai Gorski

20.1 Introduction

Recently, one can observe new trends in bank check processing. Earlier, the customers (who are banks and other financial institutions) were mainly interested in automation of reading the check amount. Nowadays, there are more and more demands on deeper analysis of check content. They originated from general political and economical situation in the world: banks try to prevent money laundering, reduce losses from fraud checks, or even detect potential terrorists among their clients.

When processing a check, a bank is interested to read automatically as much information from the document as possible. Besides the check amount, this can include the date of check issue, the beneficiary name, the payer's address and signature, code line(s), etc. This poses new tasks for developers of document analysis systems. They should be able to process and understand a check as an integral document with many loosely structured information fields, some of which are mandatory and others are optional.

Many papers on bank-check processing have been published recently [2,4,6-15]. Most of them are devoted to amount recognition as the most actual task for check industry. In this paper we also touch it, but mainly concentrate attention on new-coming tasks and describe approaches to their solution by the example of A2iA CheckReader recognition system.

Section 20.2 outlines check processing task definitions and demands to their automation. Section 20.3 describes in details recognition technologies for one of the most important check fields: payee name (including results post-processing with a user defined dictionaries). Location, extraction and

understanding of this field on both cursive handwritten and machine-printed documents are discussed. Section 20.4 briefly presents recognition of other check fields, such as date, payee address and name, and a code-line. It also discusses experimental and exploitation results of the A2iA CheckReader system achieved on bank checks originated from different countries.

20.2 Challenges of Check Processing Industry

20.2.1 What to read?

In many countries, the bank check is one of the most popular ways of payment. The financial institutions receive daily hundreds of millions of paper checks and other payment documents, from which miscellaneous data should be read, converted into electronic form and send on for further analysis. Reading and keying these data to a great extent remains a manual job performed by human operators. Replacing them by automatic reading systems is the greatest challenge of check processing industry. An important question is what should be read from a bank check?

Traditionally, banks were interested in three pieces of information: amount to be paid, account to be debited with this amount and account to be credited.

In most countries, information of debit account is coded in a special field (so called *code line*) pre-printed on each check with magnetic ink - see Fig.20.1. In the USA, the use of code lines on checks was introduced in 1956, being the first step towards automated check processing. Special MICR scanners can read magnetic code lines with high accuracy, so the information they contain (e.g. payer account number, issue bank, etc) need not be manually keyed for a great majority of checks.

The credit account number is normally known from the payee – a person or an organisation who deposits the check at his bank. The payee is identified by his name indicated in the check (Fig.20.1.), while his account number is frequently present in a separate document - deposit ticket - associated with the check. The credit account number also comes in electronic form keyed by a clerk of payee's bank at the moment of depositing.

Thus, the first information to be automatically read from the check is the amount to be paid. Normally, it exists in two forms: literal (legal amount) and numeric (courtesy amount), both being subject to read and cross-validate each other for higher reliability. These amounts can be

handwritten or machine-printed, typical fraction of machine-printed variants being around 20% of the total number of items in the document flow.

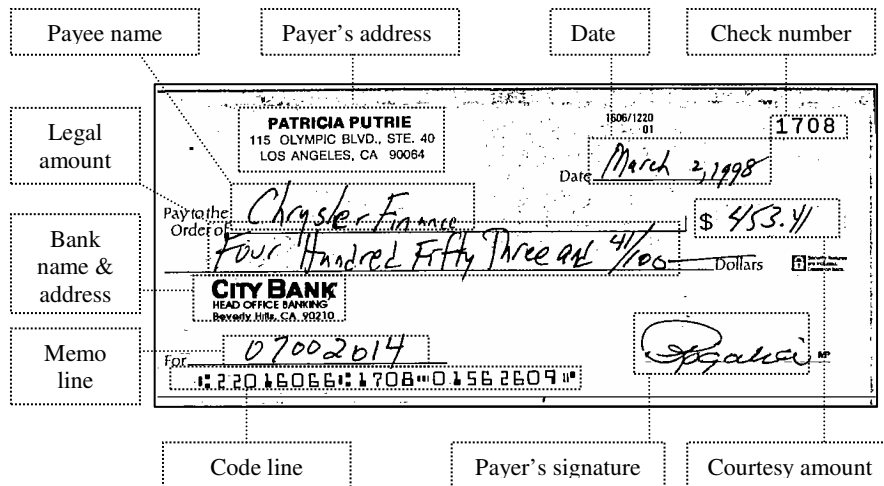


Fig. 20.1. Most important information fields of a US bank check.

Other important information fields of a check are the date and the payer's signature. In some countries (e.g. Canada, Ireland, Italy), a check becomes payable only from its date, which might be far in future from the current date. In these countries, reading the check date is a mandatory operation, which should be automated together with amount reading. As for the payer's signature, in practice, banks do verify it only on checks with relatively big amounts, because it is quite costly to accomplish this operation for all checks. Nevertheless, signature verification remains a good part of manual work, which needs to be automated.

New trends in check processing originate from realities of our anxious life. First, financial institutions are highly interested in further reducing the cost of check processing. Besides automation of traditional manual operations, this includes reducing of indirect losses, first of all from check frauds. According to the US National Check Fraud Center, check fraud and counterfeiting are among the fastest-growing problems affecting the financial system. They produce estimated annual losses of \$10 billion. To detect frauds, it is necessary to analyze many check fields and features in common, and this, in turn, demands an ability to read more information than it was several years ago.

Second, the governments increase efforts in preventing money laundering and reduction threats of terrorist attacks. New laws force banks to con-

trol more closely their transactions, thoroughly check their clients and trace who pays to whom and how much. The banks can receive lists of persons or companies under investigation whose names should be detected among payers or beneficiaries. For this, they have to read in checks such fields as payee name, payer's address, the name and address of the issuing bank. So, in this case information from a check should be analyzed in common with the external data like address books, name dictionaries, etc.

Third, massive automation of check processing raises its own problems, such as document image quality and usability. Digital copies of paper originals should be fully consistent and readable to serve as legal subjects of automatic reading; otherwise, the whole process could become useless. To prevent this, a check processing system should be able to perform a full diagnostics of input document flow to detect unusable information fields (or entire items) and warn its host about detected defects. The later is also important from the legal point of view. As some check fields are mandatory, their absence or poor readability makes the check an invalid document.

Thus, there are obvious needs to read more information fields than earlier and analyze these data in common, to provide high reliability of final decisions.

20.2.2 How to automate?

From above, one can see three different types of processes in automatic check reading: *recognition*, *detection* and *verification*.

The recognition process targets the content of an information field. Traditional check fields to be recognized are the courtesy- and legal amounts; in certain countries it is also the check date. New demands include recognition of the payee name, payer's name and address, the issue bank name. Sometimes, a code line also needs to be recognized, because not every check scanner is supplied with a MICR reader. This is true, for instance, for ATMs and cash machines, which are able to automatically deposit checks and dispense banknotes.

Automated recognition process should replace manual information keying. In reality, this is true only to a certain extent, because automatic reading systems still have lower reading abilities than human beings. However, it is possible to reduce a part of manual work by processing automatically those items (documents, fields, etc.), which the system reads with the same error level as an operator. Suppose, an operator keys items with 1% errors (a typical value for keying check amounts). Then the system, which reads 80% of items and makes among read items the same 1% errors, will do ex-

actly 80% of operator's job. So, two principal characteristics of a recognition process are the *Read rate* (READ) and the *Substitution rate* (SUBS):

$$\text{READ} = 100\% * (\# \text{ of automatically read items} / \text{total} \# \text{ of items})$$

$$\text{SUBS} = 100\% * (\# \text{ of incorrectly read items} / \# \text{ of read items})$$

Note, that substitution is related to the number of items read by the system, and not to the total number of items. When a human operator is replaced with an automatic reading system, it is the SUBS adjusted equal to that of an operator. Then the READ value reflects a labour economy. Sometimes, it is also called the "killing rate" of a system, as it shows a percentage of documents processed fully automatically and never seen by humans.

Another characteristic similar to the read rate, but measured only on the correctly read items, is the *Recognition rate* (REC):

$$\text{REC} = 100\% * (\# \text{ of correctly read items} / \text{total} \# \text{ of items})$$

READ and REC have a close meaning, but different domain of use: commercial people prefer to speak about READ (as it is connected with labour saving), while the scientific community more often uses REC, as it presents recognition ability of an algorithm or a system.

The detection process targets an item with a certain property or of a certain type (for example, checks of a particular payee). Typically, such items are very rare, their fraction being much lower than the total number of items in the flow. Traditional detection task is finding checks without payer's signature – these checks are not valid. New demands include detection of absence of any mandatory check fields; detection of payee-and/or payer names belonging to certain people; detection of fraud checks in the check flow, etc.

While recognition-type processes usually have well-organized but non-automated prototypes in check processing industry, detection processes are somewhat new and frequently do not have manual analogues. So, their implementation supposes creation of new industrial procedures, rather than replacing existing human operators by computers. As much work as possible should be performed automatically, and only a very small part demanded be done manually. In such a process, all items are analysed by an automatic system, which selects a set of suspicious items – potential detection targets. Only these items are subjects of human inspection. For example, 1% of checks can be selected as potentially invalid items. After manual verification, half of them can be found as valid; the other half being really invalid should be rejected as non-payable checks.

Efficiency of a detection process can be characterized by two complementary values: the *Detection rate* (DET) and the *Suspect rate* (SUSP):

$$\text{DET} = 100\% * (\# \text{ of detected target items} / \text{total} \# \text{ of target items})$$

$$\text{SUSP} = 100\% * (\# \text{ of suspicious items (potential targets)} / \text{total} \# \text{ of items})$$

The lower the suspect rate, the lower is the cost of the detection process, because the volume of a manual work decreases. The higher is detection rate, the higher is the revenue from the process implementation. The process is profitable when its revenue from detected targets is greater than its cost determined by the suspect rate. So, the process might be efficient even if it has not a very high detection rate, which can be achieved, for instance, by selective processing of items in the flow. For example, DET=50% of check frauds might bring higher savings than the cost of manual inspection of suspicious items at SUSP=1%.

The verification process is complementary to that of detection: while detection process finds “bad” items, verification targets at “good” ones. Again, the fraction of “bad” items is supposed to be tiny in the item flow. Verification is somehow similar to a quality control: all items are investigated, most of them are accepted, but some are rejected. An example of a traditional verification process is the payee name verification while check depositing. New demands include verification of payer’s signature and document image quality verification.

As detection, verification processes frequently do not have manual analogues and their implementation supposes creation of new automated processing where most of the work should be done automatically, and the rest needs manual processing. An automated system should analyse all items and pass only the good ones. Items rejected by the system are potential invalids – they are verified manually. Typical goal of a verification process is to pass only “good” items with a guaranteed (and very low) level of “bad” items among them. This needs total processing of every item in the flow, which might be very expensive. Totality of the process and demanded very low level of missed “bad” items are two important differences of a verification process from a detection one.

For example, all checks go to automatic image quality control. In this process 98% passes the control, and 2% are rejected as suspicious. After manual inspection, most of suspicious checks are also verified, however some documents with unreadable or corrupted information fields are considered non-payable.

Efficiency of a verification process can be measured by the same parameters as for detection one: the detection rate (DET) and the suspect rate

(SUSP). The cost of a verification process is proportional to the fraction of suspicious items (SUSP), which have to be processed manually. In check processing applications, verification processes are rarely profitable, because most of them are mandatory. For example, recently, US Check 21 standard has obliged verification of check image quality in all applications, which use digital copies of paper documents.

Table 20.1 summarizes properties of three process types described in this section.

Table 20.1. Summary of main automated processes in check processing applications.

Process	Typical tasks	Typical demanded rates	Automatic work (%)	Manual work (%)
Recognition	Amount recognition, date recognition, payee name recognition	READ = 50-80% SUBS = 1-3%	READ	100-READ
Detection	Check fraud detection, invalid check detection	DET = 50-90% SUSP < 2%	100-SUSP	SUSP
Verification	Image quality verification, signature verification	DET = 90-99% SUSP < 5%	100-SUSP	SUSP

20.3 Payee Name Recognition

Most of check recognition papers are devoted to recognition of courtesy- and legal amounts in check images [4,7,8,10,12-14]. We have also presented our technology in this domain in several papers [1,6,11]. Since that time the geography of check amount applications has been greatly enlarged (see Section 20.4.1), but the technology itself remained basically the same. So in this section we consider another technology that aims at recognition of the payee name field.

20.3.1 Field location and extraction

One of the most difficult tasks in bank check data analysis is the proper location of information fields. Despite numerous standards and regulations defining how a check should look like and what it should contain, bank clients have enough freedom to design their own checks with different layouts, miscellaneous pictures, exotic fonts, etc. Thus, a bank check is a typical loosely structured document: getting a check you can be sure that cer-

tain information fields are present in it, but you never know where these fields are really placed.

For example, in US business checks issued by corporate clients, the payee name can be found nearly anywhere in the left part of the check image (see Fig.20.2.). It can be placed on a separate text-line below- or above the legal amount, or present in a payee address block, or even sealed by a rubber stamp.

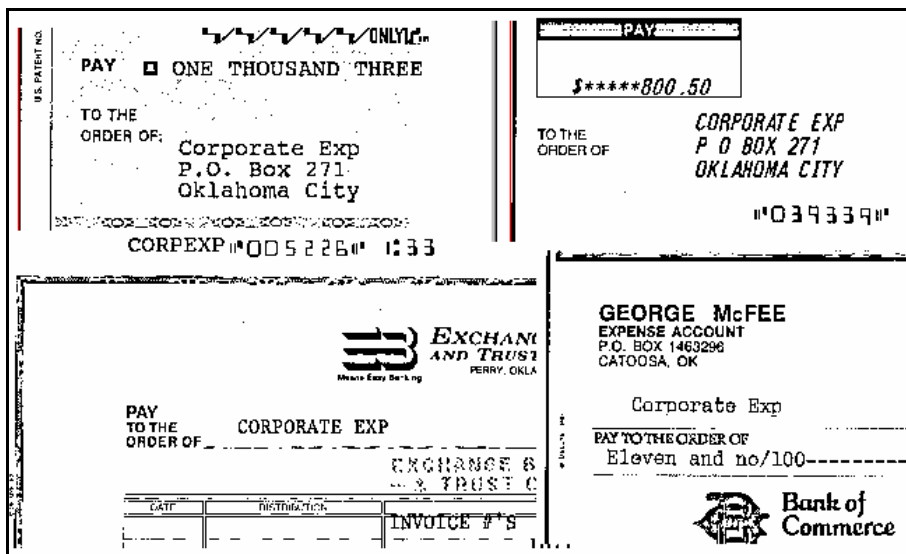


Fig. 20.2. Four examples of payee name positioning in business checks.

There are not many features, which help to locate a payee name. For business checks they are:

- Key-words “pay to the order of” or “to the order of” in the vicinity of the name.
- A paragraph of several left-aligned text-lines – possible payee address block that contains the payee name, or possible “pay to the order of” paragraph.
- Possible underline of the payee name text-line.
- Possible long text-lines (legal amount) above- or below the name.

We use all these features in the payee name location process. As soon as a business check comes into processing, all machine-printed text-lines in its left part are located and then their mutual positions are analysed to form the set of possible payee name candidate locations. Each location is characterised by the set of quantitative features that are used to train a neu-

ral network (NN) estimating the posterior probability of every candidate to be the true payee name. The candidate with the maximum probability is then taken for further analysis. In case the candidate location contains several text-lines, only the first one is considered a potential payee name. On an average, this enables to reach 90-99% correct locations of the payee name field. After cleaning the noise and side objects (e.g. underlines), the image of the payee name is sent for recognition.

In case of personal checks, payee name position is more stable than in business ones. Usually, the name is located opposite to the pre-printed currency sign (the courtesy amount marker) and underlined. It can also be preceded by the key-phrase “pay to the order of” – see Fig.20.1. More than 95% personal checks are handwritten, so the task is to find a handwritten text-line in the pre-defined location. In this case we extract the payee name zone by its position, and then remove all pre-printed objects of the check layout to keep only the handwritten text-line. The latter operation is quite delicate and difficult to adjust, because handwritten text is often very similar to other information fields. So there is a danger of both under- and over-cleaning of the payee name field. Fig.20.3. presents several name fields extracted from check images. Slant correction of the extracted images is done for handwritten fields at this step.

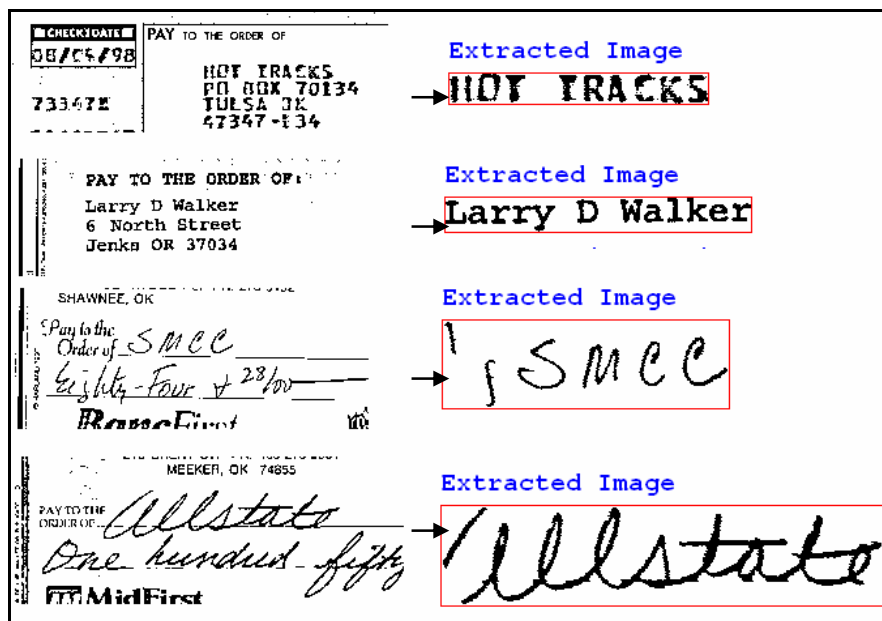


Fig. 20.3. Location and extraction of the payee name field.

20.3.2 Field recognition

Recognition technologies are different for handwritten and machine-printed fields. Machine-printed fields are recognized character-by-character, while handwritten fields (both hand-printed and cursive) are recognized on the word- or phrase basis.

20.3.2.1 Machine-printed field recognition

First, text-line is segmented into potential characters. Segmentation takes place “in parallel” with character recognition: every potential character image goes to an OCR, which estimates its score. Segmentations with higher scores are selected and form the preliminary character set. Then preliminary characters are split and/or merged depending on their sizes, scores and general properties of a text-line such as font regularity and font discontinuity. Finally, several character segmentation options are formed, each having its own probability (Fig.20.4.). For example, in the Fig.20.4 the first character may be presented by a single box (correctly containing an ‘H’) or by two boxes (erroneously containing two ‘I’s)

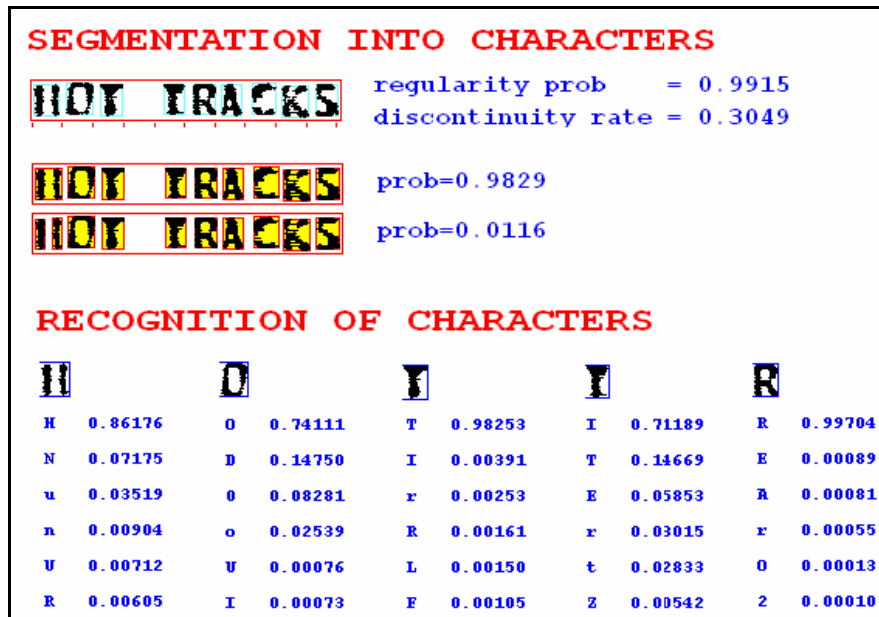


Fig. 20.4. Character segmentation and recognition of a machine-printed field.

Once characters are recognized, the payee name candidates are stochastically generated by combining character classes with their probabilities, probabilities of character segmentation options and word segmentation options (Fig.20.5.).

RECOGNITION OF OBJECTS			
HOT		IRACKS	
HOT	0.70393	IRACKS	0.46203
HDT	0.12869	TRACKS	0.19516
NOT	0.06594	IRAEKS	0.08058
VOT	0.03453	IRASKS	0.07104
NDT	0.01205	IRACES	0.06104
TIOT	0.00838	ERACKS	0.03797

Fig. 20.5. Word segmentation and recognition of a machine-printed field.

At the next step, payee name candidate list is generated from word candidates the same way as word candidates were generated from character recognition results. With this, raw recognition of the field is finished. Note, at this moment the correct candidate for the above example is only at second position in the list:

1. HOT IRACKS = 0.32520
 2. HOT TRACKS = 0.13726
 3. HDT IRACKS = 0.05914
-

Finally, the obtained raw list is filtered with a dictionary of possible payee names or with a dictionary of possible words, if they are available. The dictionary of words can be internal for the system, while the name dictionary is usually supplied by a customer. Of course, it can cover only a part of the set of possible names. Dictionary post-processing greatly improves recognition results, as it can be seen from Table 20.2. This topic is discussed also in Section 20.3.3.

20.3.2.2 Handwritten field recognition

Recognition of handwritten fields is based on Hidden Markov Models (HMMs). First, the extracted field image is segmented into graphemes which are supposed to be smaller entities than letters. On Fig.20.6 graphemes are displayed with alternate colours. Then graphemes are classified with a neural network to go from pixel representation to a feature representation, each grapheme being associated with a class probability vector. Obtained sequence of vectors is used to match with HMM models of dictionary words (phrases). Word models can be “static” trained in advance for the most frequent words like numerals, or “dynamic” built when necessary from letter models. The best-matching models form the list of possible answers, each answer being associated with its probability. In the example of Fig.20.6, the correct answer is obtained at first position in the list. Detailed description of this technology can be found in [1, 3].

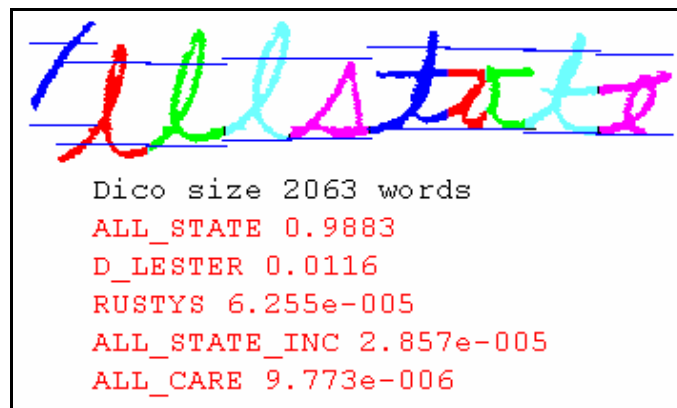


Fig. 20.6. Handwritten payee name recognition with a dictionary.

It is important, that word/phrase models should be build prior to the beginning of the recognition. Thus, unlike the machine-printed text, the handwritten text can be recognized only when the dictionary of possible words/phrases is provided to the recognizer. For the payee name recognition this means that the customer who wants to use the system should either supply a dictionary of possible names or allow the use of a generic dictionary of most frequent names, which of course could be incomplete.

Table 20.2 below demonstrates recognition results of the payee name field separately for machine-printed (business) checks and personal (handwritten) checks taken from a real document flow. The ceiling of recognition rate at 91% is mainly explained by field location problems and

partly by presence of over-lighted check images where the name is hardly can be read.

Table 20.2. Payee name recognition with- and without dictionary fully covering the set of recognized names.

SUBS(%) \ REC(%)	Machine-printed without dictionary (REC,%)	Machine-printed with dictionary (REC,%)	Handwritten with dictionary (REC,%)
1	-	90.1	70.2
2	17.8	91.0	74.9
5	60.8	91.1	79.5
10	87.9	-	82.2

20.3.3 Semantic processing

20.3.3.1 Alias problem

One of the difficulties in name recognition is so-called alias problem. Aliases are different forms of a word (in particular, a name), which have identical semantic meaning. Examples of possible name aliases are:

- Reduced or abbreviated form(s) of the prototype name (California Gas -> CA Gas)
- The prototype name with suffix or prefix words (AmEx -> AmEx Co).
- The prototype name with inserted/omitted non-important words (ATT -> AT and T).
- Misspelled name, which however can be identified with the prototype (Johannes->Johanes).
- Permuted words in the name (United States Treasury -> Treasury of the United States).

Typical information fields where aliases can be found are payee-, payer- or addressee- names in checks. Also, aliases are often appearing in dictionaries, which are used for post-processing recognition results. In this case, the name in the recognised document can be different from any its aliases in the dictionary. Presence of aliases leads to several problems:

- The correctly recognized name is rejected or gets a low score because the dictionary does not contain this name, having contained however its aliases. This reduces the recognition rate and increases rejections.

- The correctly recognized name is accepted, but considered as an error, because the recognition result does not match the desired truth name which is expressed by an alias.
- The correctly recognized name is rejected because the dictionary contains neither this name nor its aliases.

Two instruments named *alias detector* and *alias generator* have been developed to take into account presence of possible aliases in the name field and overcome the above problems.

20.3.3.2 Alias detector

The alias detector evaluates the probability that two input names are aliases. It is based on a neural network trained to distinguish aliases from non-aliases. Input features of this NN are the answers of seven heuristic *primitive detectors*, each of which is specialized in detecting aliases of a certain type:

- Primitive detectors 1 and 2 check possible word permutations in the name, as well as letter permutations.
- Primitive detectors 3 and 4 detect aliases originating from misspellings and typing errors.
- Primitive detector 5 and 6 find out and checks reduced or abbreviated words in the compared names.
- Primitive detector 7 compares names after removing non-important words, articles and suffix words

All primitive detectors are functions returning a score from 0 to 1. These scores are features for the NN-based alias detector. Training of this NN has been performed on the manually annotated data of 2000 dictionaries of real payee names as they were read and keyed by human operators.

From the annotated material, approximately 2,000,000 name pairs were generated, 20,000 (1%) being pairs of real aliases. All pairs of real aliases and 10% non-alias pairs were used for training the alias detector. Quality of the obtained detector is characterised by figures presented in Table 20.3.

Table 20.3. Detection of payee name aliases at different detection thresholds.

Detection threshold	Detected non-aliases (%)	False-detected non-aliases (%)	Detected aliases (%)	Missed aliases (%)
0.5	99.7	0.3	97.4	2.6
0.8	99.8	0.2	94.3	5.7
0.9	99.89	0.11	90.9	9.1
0.95	99.95	0.05	85.6	14.4

Thus, at the probability threshold 0.95, approximately 86% of aliases are successfully detected with an error rate 0.05%, i.e. 5 non-aliases is erroneously detected as aliases among 10,000 name pairs (in average). Such an error rate is negligible compare to the usual error rate of recognition engines, so alias probability level 0.95 was chosen as decision threshold of the alias detector.

Despite the fact that annotated material has been taken from data of a particular site, alias detector is not site-specific, as its all primitive detectors are not site-dependent. We successfully use it for any Anglo-Saxon material of personal- or company names. Some experiments even demonstrated that it is applicable also for names from non-English speaking countries, e.g., France.

20.3.3.3 Alias generator

The task of the alias generator is to produce the list of most probable aliases of a given name. The core of the alias generator is a name thesaurus. It consists of nests, each nest representing aliases of one name along with their frequencies. The thesaurus was filled with data from 20,000 name dictionaries containing about 700,000 real names as they appeared in the check flow. The thesaurus filling consisted of 4 steps:

1. Creation of the name frequency list (approximately 300,000 entries).
2. Primary thesaurus filling. Each name from 20,000 most frequent names is considered as a nest. With the alias detector, it is compared with all existing nests and either is added to the most similar nest, or forms a new nest. The most frequent names cover about 45% of name data.
3. Compression of nests. For each nest, most frequent names, words, and abbreviations are remembered. This information forms a canonical representation of a nest. Based on the canonical representation, the list of most frequent aliases is generated for each nest. This list is truncated when the cumulative frequency of its members covers 98% of all possible aliases. Typically, such a list contains 5 to 15 members.
4. Secondary thesaurus filling. Reminded names are compared with the lists of generated aliases in all nests and possibly added to the closest nest.

The filled thesaurus covers approximately 60% of names from processed dictionaries. After filling, it is used for alias generation. Given an input name, the generator determines the most probable thesaurus nest to which the input name belongs. If such a nest is found, the list of most

probable aliases is returned as a result. Optionally, this list can be truncated or enlarged depending on the desired number or frequency of generated aliases.

It should be noted, that the thesaurus is filled with data from a particular set of dictionaries. Fortunately, names presented in these dictionaries were of a great variability – input checks came from different states and different companies. So, hypothetically this thesaurus can be used not only for a particular site, but also for other similar applications. However, it is surely country-specific.

20.3.3.4 Implementation of alias recognition techniques

There are several ways to improve recognition results with the alias detector and the alias generator:

1. The number of recognition candidates in the output list is reduced by aggregating all aliases of each name. The idea behind this is to get a more “contrast” list where each name is represented by a single entry. This is important both for cursive- and machine printed name recognition, making the list shorter and candidate scores more objective.
2. The name dictionary is extended with the thesaurus and alias generator by the most frequent names absent in the dictionary and most frequent aliases of dictionary entries in a hope to cover more forms of recognized names. This is more important for recognition of cursive names, as it is sensitive to the set of input models.
3. A more objective result evaluation procedure compares recognition results with truth data taking into account alias presence.

Fig 20.7 demonstrates the influence of alias detector used for shrinking the output list. As can be seen, this essentially improves recognition rate in the domain of relatively low substitutions. A low recognition rate (~51%) is explained by the fact that nearly half of recognized names were not covered by the input dictionary provided by the customer in this application.

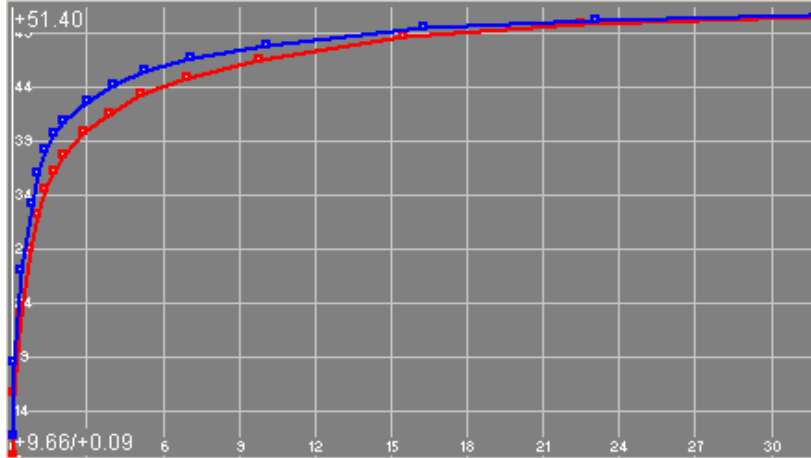


Fig. 20.7. Alias detector used for shrinking the output candidate list (upper curve) vs. the base recognition engine (lower curve). The X-axis represents recognition rate and Y-axis represents substitution rate.

After extending input dictionary with most frequent names obtained with the alias generator, further result improvement was obtained (Fig.20.8.).

The last experiment demonstrates efficiency of the alias detector in non-US application - payee name recognition on UK checks (Fig.20.9.). In this case the detector was used both for shrinking the output candidate list and objective evaluation of recognition results. Alias generator was not used in this case, as it is applicable only to US payee names.

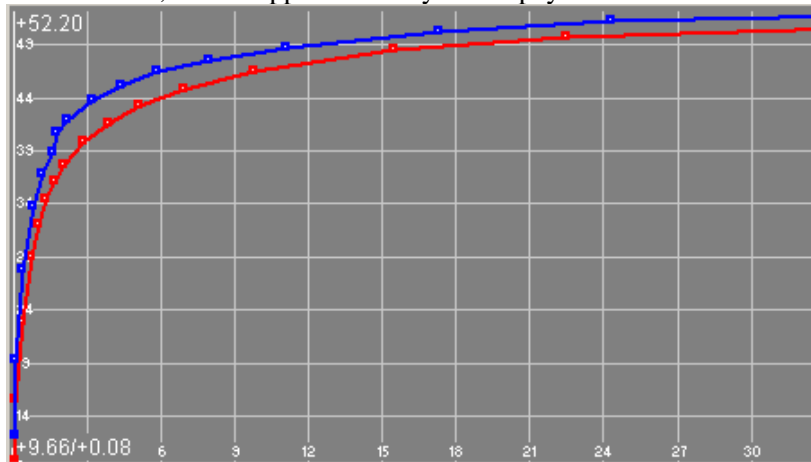


Fig. 20.8. Alias detector and alias generator used together (upper curve) vs. the base recognition engine (lower curve). X-axis represents recognition rate, Y-axis represents substitution rate.

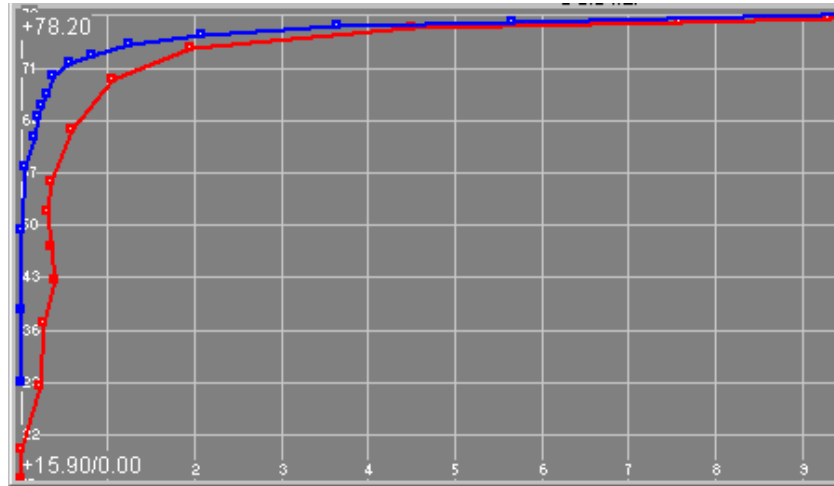


Fig. 20.9. Payee name recognition on UK checks with alias recognition technology (upper curve) and without it (lower curve). The X-axis represents recognition rate and Y-axis represents substitution rate.

4 Check Mining with A2iA CheckReader™

4.1 Amount recognition

A2iA CheckReader™ was primarily designed to replace human operators in automated payment systems, i.e. to read amounts of bank checks and associated documents [6,11]. Besides checks themselves, a real document flow contains other items, such as deposit tickets, debit- and credit forms, money orders, cash tickets, etc. As checks, these items also contain amounts to be recognized. Normally, all payment documents are scanned in a common stream and should be processed in a similar way. Thus, the main task of the system in this application is detection and recognition of amounts on all documents of the flow.

Since 1994, seventeen country-specific versions of the system have been developed, most of them providing recognition and cross-validation of courtesy amount (CA) and legal amount (LA). Recent developments included recognition of handwritten Chinese hieroglyphic amounts on Hong Kong checks [16] and cursive Italian LA frequently written without word

spaces. Table 20.4 summarizes amount recognition results at different substitution levels. In most of real applications, customers either chose substitution level between 1% and 1.5%, or use maximum recognition. Results of Table 20.4 should be considered with care, as they are site-specific. From site to site, the rates can easily vary within 5% margins mainly depending on the quality of input check images.

Table 20.4. Check amount recognition results of country-specific CheckReader versions

Country-specific version	Recognized LA language(s)	REC (%) at SUBS=1%	REC (%) at SUBS=2%	Maximum REC (%)
Australia	English	79	83	89
Belgium	n/a	91	93	94.5
Brazil	Portuguese	52	60	75
Canada	English and French	75	79	85
France	French	70	76	82
Germany	n/a	65	71	80
Hong Kong	English and Chinese	63	70	80
Ireland	English	81	86	89.5
Italy	Italian	65	72	80
Mexico	Spanish	55	59	71
Malaysia	English	57	62	72
New Caledonia	French	55	62	72
Portugal	Portuguese	68	74	85
Singapore	English	71	76	83
Thailand	n/a	42	50	63
UK	English	80	84	88
US	English	79	82	87

20.4.2 Recognition of other check fields

Gradually, CheckReader enlarges the variety of recognized field types. The demand of new field recognition usually originates from a single customer, and then the new functionality added to the system is spread among other customers and frequently extended to other countries. For example, check date recognition developed first for Canada is now available for six country versions, and payee name recognition – for three countries.

Technology of check field recognition is similar to that described in section 20.3. The first step consists of field location in the check image

with subsequent extraction of machine-printed or handwritten text-lines to be recognized. Further steps are different for fields with well-separated characters (machine- or hand-printed) and fields with touching characters (unconstrained handwriting or pure cursive).

Well-separated fields are segmented into characters that are recognized by OCRs. Word- and phrase options are then stochastically generated from character recognition results forming the raw candidate list of possible field contents. In case of available semantic or linguistic information, the raw list is post-processed or filtered with a dictionary to improve the result.

Unconstrained or cursive fields are segmented into graphemes and recognized with HMMs. In this case presence of linguistic or semantic information is mandatory; mainly it is used to prepare properly the set of models to which the recognized field is matched.

The concluding step is decision making. It is similar for all field types. It consists of evaluating the probability that the obtained recognition result is correct. Normally, a specially trained NN is used to fulfil this operation (see [5] for more detailed description of the technology). If the obtained decision probability is higher than a pre-defined threshold, the recognition result is accepted; otherwise, it is rejected. Table 20.5 summarizes CheckReader results for non-amount field recognition.

Table 20.5. Recognition results of CheckReader versions. Each cell presents values REC (%) / SUBS (%) for corresponding country version and field type.

Country	Date	Payee name (with dict.)	Payer address (without dict.)	Code line
Canada	60 / 3	-	-	94 / 1
France	39 / 3	71 / 1	max 55	98 / 1
Italy	42 / 3	-	-	73 / 1
Ireland	52 / 3	-	-	85 / 1
UK	55 / 3	78 / 1	-	96 / 1
US	65 / 3	82 / 1	max 58	85 / 1

20.4.3 Detection and verification tasks

Recently, CheckReader has acquired functionalities to accomplish new types of check processing tasks. They are: detection of invalid checks, check fraud detection, detection of names from “black lists” and name verification.

20.4.3.1 Invalid check detection

In every country, there are regulations specifying the set of features that a valid check should possess. Mainly they concern presence or filling of certain information fields in the check. For example, both courtesy- and legal amount should be indicated; payee name should be present as well as the payer's address; the check should be signed by the payer.

To answer a question whether a check is valid or not, CheckReader is trained to locate all mandatory fields in the check image and detect whether each of them is filled or remains empty. If at least one detector reports emptiness of a mandatory field, the check is considered suspicious and sent to the suspect basket. Suspicious checks are evaluated by a neural net, which is trained to distinguish invalid checks from valid ones. This network returns a check invalidity probability. When it is higher than a decision threshold, the check goes to manual inspection to confirm its invalidity. Table 20.7 presents invalid check detection results achieved by the CheckReader. Test data included 3000 valid- and several hundred invalid checks of different types for every country version. Most invalid checks for the test were selected while visual inspection of real documents and some were prepared manually.

Table 20.6. Invalid check detection by CheckReader versions.

Country	DET (%) at SUSP=1%	DET (%) at SUSP=2%	DET (%) at SUSP=5%
Canada	50	68	81
France	45	65	92
US	65	76	85

20.4.3.2 Detection of check frauds

Another important functionality is fraudulent checks detection. Check frauds become a serious problem causing sensible losses in payment systems, especially in US where many people do not have their own bank accounts and can get cash at the moment of check depositing.

To detect check frauds in the check flow, CheckReader has an ability to compare each input check with reference check(s) of the payer. Of course, every payer who wants to protect his checks from counterfeits should supply at least one valid check and all mandatory fields properly filled as a reference. The payer's account number indicated in the code-line identifies the reference.

Twelve various fraud detectors are used to compare a new incoming check with the existing reference(s). Each detector verifies a certain feature, which can indicate a possible fraud. In particular, detectors analyse geometry of check layout; positions of pre-printed markers, logos and keywords; positions of check information fields. Other detectors analyse payer’s handwriting (in case of a handwritten check). There are also detectors to find hidden cross-correlations between the content of the reference(s) and the investigated checks. Results of all detectors are integrated and fraud probability of the analysed check is evaluated. If it is higher than a given threshold, the check is considered a potential fraud and goes to manual verification.

Table 20.7 presents experimental results of fraud detection obtained on a data set of 5,780 documents. One third of the set were fraud checks manually simulated from real ones. Probably frauds were not enough sophisticated, as demonstrated detection rate is rather high.

Table 20.7. Fraud check detection by the CheckReader-US.

DET (%)	79.5	92.4	95.9	97.8
SUSP (%)	0.5	1	2	5

20.4.3.3 Name detection and verification

A number of check processing applications are concentrated around payee- and payer’s name analysis. Besides ordinary name recognition described in Section 20.2, they include detection and verification of special names. In both of these applications, the system is supplied with a dictionary of names, which should be found in the check.

Name detection is a search process, when the system is supplied with a so-called “black list” of wanted people (e.g. terrorists, criminals, etc.). Then every name recognized in the check is matched with names from the black list and if matching occurs, the check goes to a special basket of items to be manually inspected.

The difference from the ordinary name recognition is in the percentage of check names covered by the dictionary: in case of usual recognition dictionary coverage varies from 30 to 100%, while in detection tasks it is essentially lower than 1%. To adapt Check reader to this task we tuned decision-making module to detect demanded fraction of “black” names. Table 20.8 present detection results for payee- and payer names. Results for payer name are higher because this name is always machine-printed, while payee name is handwritten in 60-80% of checks

Table 20.8. “Black” name detection by the CheckReader-US.

Field type	DET (%) at SUSP=1%	DET (%) at SUSP=2%	DET (%) at SUSP=5%
Payee name	78	80.5	84
Payer name	92	93.5	94

Name verification is one of the ways to prevent check frauds. In this case, the bank issuing the check supplies it with a special record, duplicating the payee name. When such a check is deposited, its payee name is verified with the recorded name coming to the depositor by an alternative information channel. Mismatch of two names signals that the check is a potential fraud. Despite its seeming simplicity, this task is one of the most difficult, because (as every verification task) it demands analysis of every check and a very high acceptance rate. Therefore, both name location and recognition rates should be good enough; otherwise, too many checks would be rejected and the fraction of manual work becomes too big. Fortunately, all checks in this application are machine-printed.

To solve this task, CheckReader uses the payee recognition process described in Section 20.2, i.e. makes full name recognition without dictionary. Then, the obtained candidate list is compared with the recorded name, which plays the role of a dictionary with a single entry. A special decision module has been developed to reach demanded performances of the verification process. The achieved results are present in Table 20.9.

Table 20.9. Payee name verification by the CheckReader-US.

DET (%)	90	92	93.5	95
SUSP (%)	1	2	5	10

Conclusions

We have presented integrated technologies of bank check processing, which aim is to extract and use all available information from the check image. Three types of processes were considered: recognition (of the check amount, date, payee name, payer address, code-line), detection (of the signature, “black” payee names, frauds) and verification (of the payee name). Implementation of developed technologies in the A2iA CheckReader™ demonstrates that a recognition system is able reach the level of automation at 60-80% in recognition processes and 90-99% in detection and verification processes, thus greatly reducing manual work in banking industry.

References

1. Augustin E, Baret O, Price D, Knerr S (1998) Legal amount recognition on French bank checks using a neural network-hidden Markov model hybrid. In: S.-W. Lee, (ed) *Advances in Handwriting Recognition*. World Scientific, pp 81-90
2. *Automatic Bankcheck Processing*. (1997) Impedovo S, Wang P, Bunke H (eds). World Scientific
3. Dupre X, Augustin E (2004) Hidden markov models for couples of letters applied to handwriting recognition. In: *Proc. of the 17th International Conference on Pattern Recognition*. Cambridge, UK, pp 618-621
4. Dzuba G, Filatov A, Gershuny D, Kil I, Nikitin V (1997) Check amount recognition based on the cross validation of courtesy and legal amount fields. In: *Int. Journ. of Pattern Recognition and Artificial Intelligence*. 11(4) :639-655
5. Gorski N (1997) Optimizing error-reject trade-off in recognition systems. In: *Proc. of the 4-th Int. Conf. on Document Analysis and Recognition*. Ulm, Germany, pp 1092-1096
6. Gorski N, Anisimov V, Augustin E, Baret O, Maximov S (2001) Industrial bank check processing: A2iA Check Reader, *Int. Journ. of Document Analysis and Recognition*, (3)4 :196-206
7. Greco N, Impedovo D, Lucchese M, Salzo A, Sarcinella L (2003) Bank-check processing system: modifications due to the new European currency. In: *Proc. of the 7-th Int. Conf. on Document Analysis and Recognition*. Edinburgh, Scotland. pp 343-348
8. Guillevic D, Suen CY (1995) Cursive script recognition applied to the processing of bank checks. In: *Proc. of the 3-d Int. Conf. on Document Analysis and Recognition*. Montreal, Canada, pp 11-14
9. Kaufmann G, Bunke H (1999) Error localization and correction in check processing. In: S.-W. Lee (ed) *Advances in Handwriting Recognition*. World Scientific, pp 111-120
10. Kelland S, Wesolkovski S. (1999) A comparison of research and production architectures for check reading. In: *Proc. of the 5-th Int. Conf. on Document Analysis and Recognition*. Bangalore, India, pp 99-102
11. Knerr S, Anisimov V, Baret O, Gorski N, Price D, Simon J-C (1997) The A2iA Intercheque system: courtesy amount and legal amount recognition for French checks. *Int. Journ. of Pattern Recognition and Artificial Intelligence*, 11(4) :505-547
12. Lethelier E, Leroux M, Gilloux M (1995) An automatic reading system for handwritten numeral amounts on French checks. In: *Proc. of the 3-d Int. Conf. on Document Analysis and Recognition*. Montreal, Canada, pp 92-96
13. Di Lecce V, Dimauro A, Guerriero A, Impedovo S, Pirlo G, Salzo A (2000) A new hybrid approach for legal amount recognition. In: *Proc. of the 7th Int. Workshop on Frontiers in Handwriting Recognition*. Amsterdam, the Netherlands, pp 199-208

14. Oliveira L, Sabourin R, Bortolozzi F, Suen CY (2001) A modular system to recognize numerical amounts on Brazilian bank checks. In: Proc. of the 6-th Int. Conf. on Document Analysis and Recognition. Seattle, USA, pp 389-394
15. Shetty S, Shridhar M, Houle G, (2000) Background elimination in bank checks using greyscale morphology. In: Proc. of the 7th Int. Workshop on Frontiers in Handwriting Recognition. Amsterdam, the Netherlands, pp 83-91
16. Tang H, Augustin E, Suen CY, Baret O, Cheriet M (2004) Recognition of Unconstrained Legal Amounts Handwritten on Chinese Bank Checks. In: Proc. of the 17th International Conference on Pattern Recognition. Cambridge, UK, pp 610-613