

A System for the Automated Reading of Check Amounts - Some Key Ideas

Guido Kaufmann and Horst Bunke

University of Bern, Institut für Informatik und angewandte Mathematik
Neubrückstrasse 10, CH-3012 Bern, Switzerland
{kaufmann,bunke}@iam.unibe.ch

Abstract. In this paper we present a reading system for amounts extracted from bank and postal checks. We focus our description on the legal amount recognition and the combination of the recognition results for the legal and the courtesy amounts. For these tasks we developed and applied several new techniques. In our work we deal with German check amounts. The automated reading of German legal amounts is a great challenge since the literal words the legal amount is constructed of are completely connected. Our system was tested on a database with real checks from the Swiss postal services.

1 Introduction

Automated reading of amounts extracted from postal or bank checks can be divided into three tasks: courtesy amount recognition, legal amount recognition, and subsequent combination of the results of the two recognizers. The combination results in either the recognized amount or a rejection of the check. Since digit and digit string recognition can be performed faster and with higher recognition rates than the reading of handwritten words, the courtesy amount recognizer has become dominant in check reading. The first check reading systems even rely on the recognized courtesy amount only, and ignore the legal amount. In later systems, some of which are still under development, legal amounts are also recognized. Their recognition results are used to verify the hypotheses generated by the courtesy amount recognizer, which helps improving the system's reliability and performance significantly. The recognition of the legal amount is either driven by the results of the courtesy amount recognizer [1], or performed independently [2]. For a good overview of the state-of-the-art in automatic bankcheck processing see the two special issues of the Int. Journal of Pattern Recognition and Artificial Intelligence [3] published on this subject.

This paper presents a system for reading handwritten bankcheck amounts. In the description of the main components we focus on some key ideas and new approaches that have been explored. In our work we are dealing with German amounts. Here recognition is very challenging due to the nature of writing German numbers by words. In contrast to many other languages, such as English and French, the literal words in German that form the legal amount are separated by very small gaps only, or they are completely connected by ligatures.

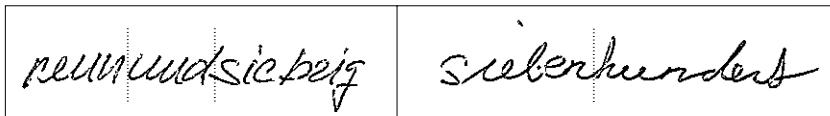


Fig. 1. Two samples of typical German numbers represented by words where the literal words are separated by very small gaps only (left: *neunundsiebzig* (seventy nine)), or completely connected by ligatures (right: *siebenhundert* (seven hundred)). With the dotted lines the hypothetical segmentation positions are marked.

These ligatures and the gaps are not different from those between the characters in a word (Fig. 1). Therefore a recognition approach for the legal amount based on its segmentation into literal words and their subsequent independent recognition is not possible. The legal amount has to be recognized as a whole which leads to a very large vocabulary.

The paper is structured as follows: Section 2 describes the recognition module for the legal amount. We applied a Hidden Markov Model (HMM) based recognition approach, since HMMs are well suited for reading a word without segmenting it into smaller components. For our work this approach was adapted for a segmentation-free recognition of the whole worded amount (consisting of several literal words) by hierarchically building models from simpler literal words. Our HMM-based word recognizer differs from other HMM-based word classifiers described in literature mainly in the type of features extracted from the input pattern. We propose to use the pure bitmap instead of a description of structural or topological elements [1,4,5]. In Section 3 the courtesy amount recognizer is presented, where a combined segmentation-based and segmentation-free recognition approach has been implemented. The combination of the recognition results for the courtesy amount and the legal amount is the topic of Section 4. We first describe a method for a systematic translation of recognized legal amounts into digit strings. The strategy of the subsequent combination is based on comparing the ranked lists of the single recognizers and searching for an agreement within the two lists. For checks where no agreement can be found, we propose a new approach to detecting, analyzing and correcting the potential errors of our legal- or courtesy amount recognizer. This means the mismatching subwords and digits are first localized and extracted from the input image. Then a second classifier is applied to those subwords that have been identified as potential errors. Experimental results are described in Section 5. Our system was tested on a database with 1500 real check amounts from Swiss postal services. As the focus of our work is on recognition rather than preprocessing and segmentation, these amounts were manually extracted from checks. Conclusions are presented in Section 6.

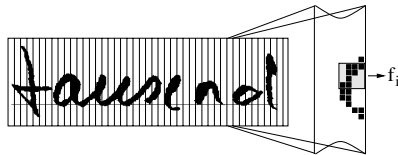


Fig. 2. In the feature extraction step a sliding window is further divided into squares (4×4 pixels) whose number of black pixels are counted.

2 Legal Amount Recognition

The process of legal amount recognition is generally divided into segmentation and recognition [2,7]: In the *segmentation* step the legal amount is split into its literal words; in most languages (English, French, Italian, German, for example) there are 20 to 30 such different literal words¹ the legal amounts are constructed of by concatenation [1,6,2]. In the *recognition* step the extracted words are classified. In earlier works in cursive script recognition the techniques applied for the recognition of handwritten words were based on further segmenting the words into individual characters and classifying them. Alternatively, in the last few years Hidden Markov Models (HMMs), widely used in the field of continuous speech recognition for a long time, have been qualifying as an excellent tool also for cursive script recognition. The words are recognized as a whole without any explicit segmentation into characters. Segmentation and recognition are performed jointly and therefore the extremely difficult and error prone segmentation task can be avoided [8]. Sequences of features are implicitly associated with the characters of a word by the HMM.

In the HMM-based legal amount recognizer presented in this section, four different tasks need to be solved, namely, preprocessing, feature extraction, training, and recognition. They will be described in the following subsections.

2.1 Preprocessing and Feature Extraction

In an HMM framework each pattern to be recognized has to be described by a sequence of feature vectors. A common approach to this kind of representation is to scan the image from left to right and to compute at regular positions a feature vector from a vertical strip, called *sliding window* [4] or *frame* [8]. In most recognition systems for cursive handwriting, high-level features, i.e. character fragments, such as lines, curves, or dots are used [1,4]. In contrast, we explore in our work the use of bitmap-based features, similar to an approach presented in [8]. I.e. the positions of the pixels set within a sliding window are the only information we extract. For that purpose the sliding window (in our work 216×8 pixels) is further split into small subwindows (4×4 pixels). For each subwindow the number of black pixels are counted (Fig. 2). This results in vectors with 108

¹ Analyzing the grammar of German numerals we found 26 different literal words.

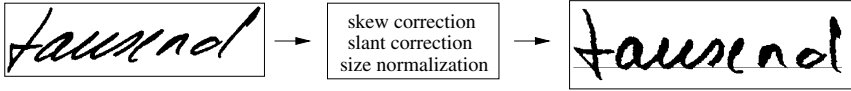


Fig. 3. The normalization steps in the preprocessing step of the legal amount

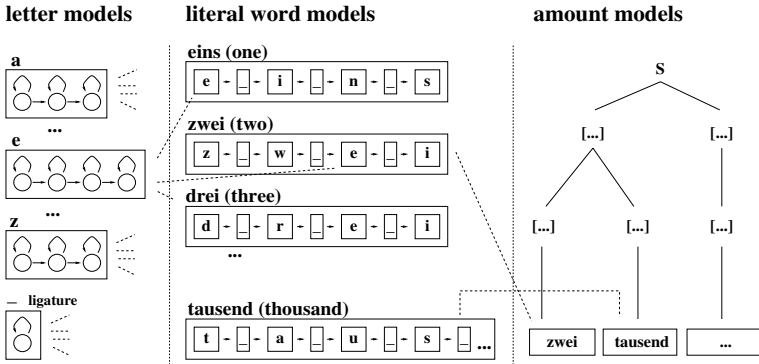


Fig. 4. Graphical illustration of the model building process. The dotted lines are references from the literal word HMMs to the letter HMMs, and from the (amount) word HMMs to the literal word HMMs, respectively.

feature values each. In our work these feature vectors are subsequently reduced by a principal axis transformation to dimension 18.

Working with this simple kind of features no sophisticated structural and topological analysis of the pattern has to be implemented. However, these features require a good normalization in the preprocessing step to eliminate writing variability as much as possible.

In the preprocessing step we apply several algorithms to reduce the effects of typical variations in handwriting, namely, the correction of skewness, slant and size differences (Fig. 3). The corrections are done by estimating the difference of the input word to a standard form and performing the inverse geometric transformation. For further details see [9].

2.2 HMM Model Structure, Training and Recognition

In order to model a legal amount by an HMM, we first define a (linear) HMM for each letter appearing in our words. Then a HMM for each of the literal words is built by concatenating the appropriate letter models. These HMMs are further combined in a hierarchical network, based on the grammar that describes the set of syntactically correct words. Each path in this network represents a valid legal amount. A graphical illustration of this model building process for the legal amounts is shown in Fig. 4. Notice that the letter HMMs are the smallest components and there exists only one HMM per letter regardless of the number

experiment 1	recognition rate	experiment 2	recognition rate
system [BIP] [1]	90.2%	system [BRS] [5]	98.4%
our system	96.1%	our system	98.8%

Table 1. Comparison of our system with two others HMM-based recognition systems based on structural features.

of occurrences of the letter in a literal word. The literal word models and the amount models are concatenations of these letter models, which are dynamically expanded during recognition.

The parameters of the letter HMMs are obtained in the training phase via the standard Baum-Welch algorithm [10]. The classification of a legal amount is performed by means of a modified Viterbi algorithm [10]. It consists in finding the best path in our HMM-network of all valid amounts. Due to the large vocabulary of all valid legal amounts – the vocabulary is further increased because there are very often more than one valid verbal representation for a numeral – an exhaustive search would be too slow. So we use a beam search strategy where paths are dynamically pruned during recognition. This classification algorithm will not necessarily deliver the best path, but only a suboptimal one. However, the search space can be kept reasonably small, allowing a fast search. Both training and recognition of our legal amount recognizer are based on the ISADORA system [5]. The ISADORA system provides flexible HMM-based pattern analysis tools and the possibility to build hierarchically structured models from simpler constituents, as described in the previous paragraph. Besides cursive script recognition, ISADORA has been applied in many other fields, for example, automatic speech recognition and the diagnosis of sensor data from moving vehicles.

In some preliminary tests we examined the quality of the features used in our system, comparing our recognizer with two different systems based on structural features [1,5]. In the first experiment (see Table 1) a database of 13'000 German literal words collected from 500 different writers was used. This database was divided into five disjoint sets, each consisting of 2'600 words. Each of these sets was once used as a test set, while with the others the HMMs were trained. On this database we compared our system with the one presented in [1].

In the second experiment we worked with a smaller database containing 3000 English words from 5 different writers. This database was collected by the authors of [5] from so-called “cooperative writers”. Each person wrote the 150 words of the vocabulary four times. The four data sets of this database were also cyclically used as test and training sets, similarly to the previous experiment.

In both experiments under exactly the same conditions (training/test sets) our system achieved similar or better results (see Table 1).

3 Courtesy Amount Recognition

For courtesy amount reading we adapted a system that was developed (in another project) for the recognition of digit strings. For a detailed description of this system see [11]. In our check reading system the digit recognition subsystem was trained with 50'000 digits collected from Swiss postal data. (These digits are not part of the database used in the experiments described in Section 5.) Working with this recognizer our courtesy amounts had to be restricted to amounts without cent portions. For the reading of Swiss postal checks this was a minor restriction since a cent part does not very often appear on Swiss postal checks.

4 Combination of Legal and Courtesy Amounts

In our system the courtesy and the legal amounts are recognized separately, without using any knowledge of each other. In the combination module the ranked lists of the recognized legal amount and courtesy amount are compared, considering the top L and C ranks of each recognizer, respectively, where L and C are system parameters. If the top ranks match, i.e., if there is an amount common to the top L ranks of the legal amount and the top C ranks of the courtesy amount recognizer, the check is accepted, with this common amount as its result, otherwise it is rejected. Due to the fact that typical confusions of digits generally do not correspond with typical confusions of words, this seems to be a quite promising approach. Since recognition rates for handwritten words are still rather low, though, this yields quite high rejection rates. In the following we describe an approach to lowering the rejection rate without increasing the error rate.

4.1 Amount Translation

Before we can compare the recognized legal and courtesy amounts in the combination step, the legal amounts have to be translated into digit strings. A problem that occurs in the processing of German amounts at this step is that the order of the numeral words does not need to correspond to the digit order. Therefore the translation cannot be done by simply replacing words from left to right by digits or blanks, as in English (for example *seventy nine*: seventy \rightarrow 7, nine \rightarrow 9). In German, in the legal amount *neunundsiebzig* (seventy nine, 79) the order of the numeral words that correspond to the digits 7 and 9 (i.e. *siebzig* and *neun*) is reversed. For this translation step we proposed in [12] a systematic and flexible approach, using the technique of syntax-directed translation (SDT). It can easily be adapted to other languages and is especially useful for languages where the ordering problem described above occurs, and many different word representations correspond to the same digit amount.

Our implementation of SDT delivers not only a translation of each recognized legal amount into its corresponding digit string, but also the information for each digit to which word of the legal amount it corresponds. For the example given

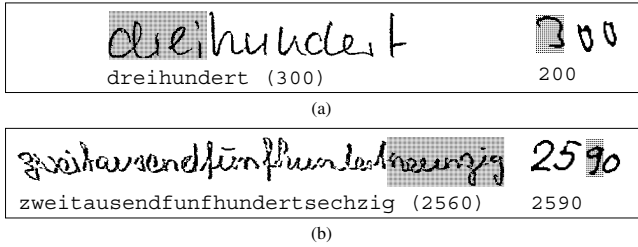


Fig. 5. Two examples of localized mismatching parts.

above, we not only get the digit string 79 as translation of the legal amount *neunundsiebzig* (seventy nine), but also the information that the first word *neun* (nine) corresponds to the second digit of the translated legal amount, i.e. 9, and the second word *siebzig* (seventy) to the first digit 7, respectively.

4.2 Localization of Mismatching Parts

In case of a disagreement between the courtesy amount and the translated legal amount this additional translation information is used to detect the cause of the missing correspondence. First, the two digit strings of the courtesy amount and translated legal amount are compared digit by digit. For each mismatching digit in the translated legal amount, the corresponding word in the legal amount is determined. The information needed for this step is provided by the SDT in the translation step. The start and end position of the corresponding word in the image are obtained from the HMM as a byproduct of the recognition process (for further details see [13]).

Two examples of located mismatching parts are shown in Fig 5. In example (a), the correct recognition result for the legal and courtesy amount recognizer is *dreihundert* and 300, respectively. However, the first digit in the courtesy amount was misrecognized as 2. As indicated in example (a), the system has correctly located the mismatching subunits. In example (b), the correct legal amount of *zweitausendfünfhundertneunzig* (2590) was misrecognized as *zweitausendfünfhundertsechzig* (2560). The mismatching subword *neunzig* (ninety) and the digit 9 are correctly detected and marked.

4.3 Correction of Recognition Errors

The marked words represent potential recognition errors. We call them *potential* errors because we are not completely sure that the disagreement was produced by a recognition error in the legal amount. Actually, there are three different causes that could have led to the disagreement. Either the subword was misclassified, or the corresponding digit, or both of them. Instead of rejecting a check when the legal and courtesy amount do not completely match, we use the images of mismatching subunits for a further post-processing where we try to correct the

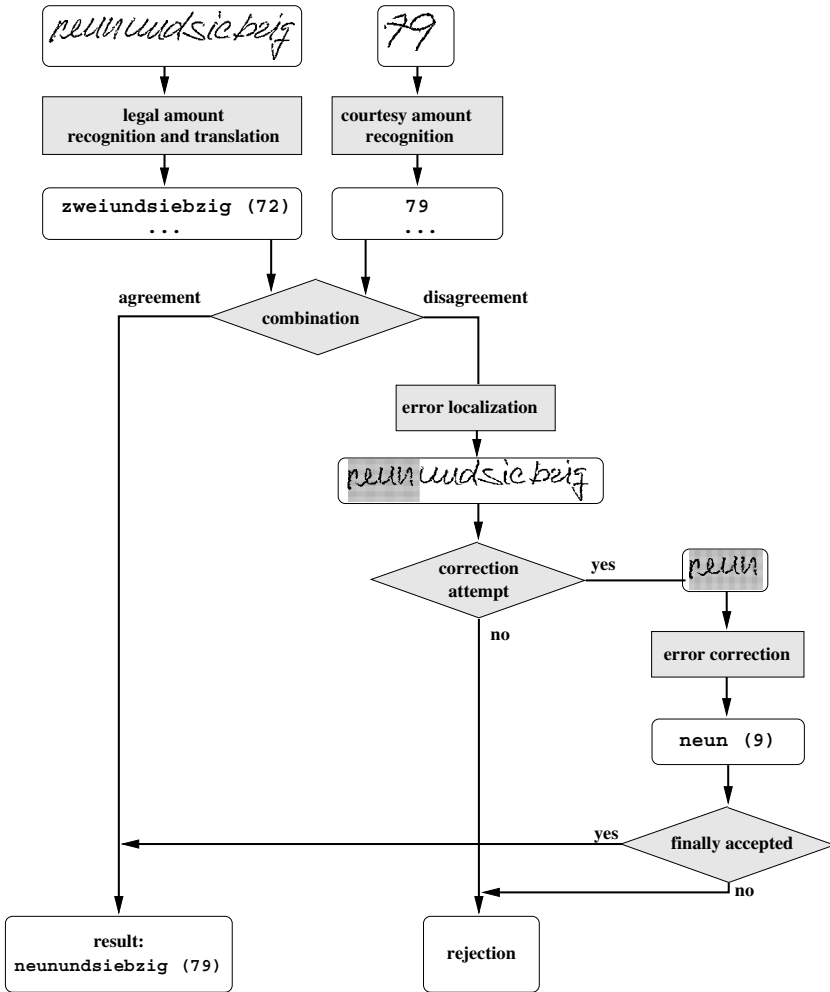


Fig. 6. The general system architecture enhanced by modules for error localization and correction. In the example the legal amount was recognized as *zwei*undsiebzig (seventy two) instead of *neun*undsiebzig (seventy nine). The courtesy amount was correctly recognized as 79. Therefore the first word *neun* (nine) recognized as *zwei* (two) is a potential error and analyzed by the error correction module.

potential errors. Since digit recognition is performed with higher reliability, we focus in our work primarily on the correction of mismatching subwords. In Fig. 6 our enhanced system architecture shown.

In the module *correction attempt* in Fig. 6 we analyze the recognition results once more, in order to make a decision whether it is worth trying to correct the detected potential errors or not. This means that we want to predict whether we can expect an agreement between the two amounts after a more sophisticated, second classification of the non-matching part of the legal amount. In other words, we try to identify those checks where correction attempts are likely to succeed. In particular, we eliminate and reject checks where several errors in the legal amount were localized, since the time needed for correction will increase proportionally to the number of errors detected. Furthermore, chances that all errors can be corrected decrease with each additional potential error. We also check the confidence of the corresponding digit delivered by the courtesy amount recognizer. If it has a low value the disagreement might be caused by an error of the digit recognizer. Therefore we probably won't be able to correct the error by further investigating the legal amount. Even worse, we risk causing an additional error, namely, changing the correct word such that it corresponds with the wrong digit.

If a legal amount is declared as a candidate for an error correction attempt, its marked subword is cut out of the image and inspected by a second, more sophisticated classifier. The second classifier deals only with a single word at a time, without any knowledge of the whole legal amount, the courtesy amount, or the expected result. If this re-estimation delivers another result for the subword, so that the newly generated legal amount now agrees with the courtesy amount, we assume that there was an error that has now been corrected. Therefore the check is finally accepted; otherwise it is rejected.

There are many causes that could have led to the first misclassification of the subword by the legal amount recognizer. In each processing step (preprocessing, feature extraction, classification) errors can happen. In our system, the second classifier starts with another preprocessing of the extracted subword. Actually, the considered subword has already been completely preprocessed in the context of the whole legal amount. However in a second round of preprocessing we might get some better results, for example, for the detection of the reference lines or the correction of skew and slant, since we do it on a local level. In Fig. 7 an example is given. For the legal amount *achtundvierzig* (forty eight) the lower baseline could not be placed properly, since the first subword *acht* (eight) was written skewed relative to the other two subwords *und* (and) and *vierzig* (forty). This problem might have caused the wrong classification of this subword as *zwei* (two). In the error localization step the word was declared as a potential error and the system decided to perform an error correction attempt. After another (local) preprocessing step with skewness correction the baseline was placed in a better position and the word was finally correctly recognized as *acht* (eight).

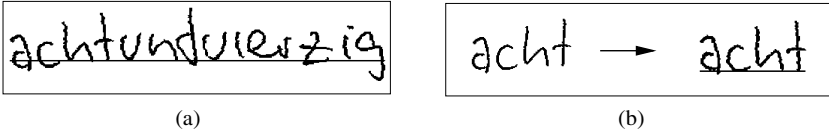


Fig. 7. The lower baseline for the legal amount was not correctly detected for all subwords (a) which might have led to the recognition error for the first subword *acht* (eight) that was recognized as *zwei* (two). After a local preprocessing (b) with another correction of skewness the baseline was detected properly and the subword was correctly recognized as *acht*.

rank	1	2	3	4	5	10
courtesy amount	79.3%	83.9%	85.7%	86.7%	87.1%	87.7%
legal amount	71.9%	81.3%	84.4%	86.5%	87.5%	88.8%

Table 2. Recognition rates (with zero rejection) of the legal and courtesy amount recognizers on the database used in this work.

5 Experimental Results

For our experiments a test database containing 1500 manually extracted legal and courtesy amounts from real Swiss postal checks was built. In the manual segmentation also the currency characters (for example, *Franken*, *Fr.* or *SFr.*), the cents portion (which is quite rare on Swiss Postal checks), and additional symbols, such as horizontal lines that delimit the amount, were removed.

On this database the individual recognizers for the legal and the courtesy amount achieved the recognition rates presented in Table 2.

We first tested our system with the simple combination strategy described at the beginning of Section 4, where the first L and C ranks are considered. In Fig. 8 (a) the rejection and the error rate for a system with $L = C$ is presented. For $L = C = 1$ (i.e. only the first ranks are considered), our system accepts 58.1% of all checks with an error rate of 0.2%. With more risk, expressed by higher values for L and C , the rejection rate drops while the error rate increases significantly.

If the values for L and C are different, the case where only the first rank of the courtesy amount ($C = 1$) and the best n ranks of the legal amount ($L = n$) are considered is of special interest (Fig. 8 (b)). Here the rejection rate decreases with a higher value for L similar to the previous case, but the error rate does not increase so quickly.

In the two experiments described in the following, we tested the influence of the correction method described in Sections 4.2 and 4.3 on the acceptance and error rates of our system. The difference between the two experiments is the number of ranks of the recognized legal amount (L) that were considered in the combination module. From the results of the courtesy amount recognizer we considered in both experiments only the top rank ($C = 1$).

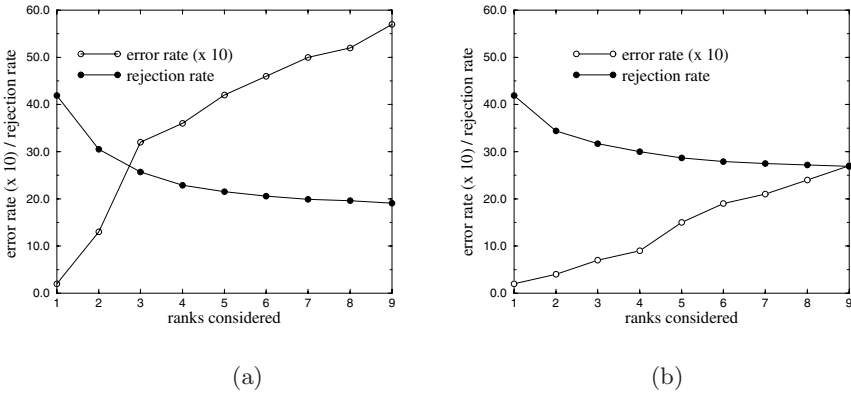


Fig. 8. Recognition rate with different system parameters for the number of ranks considered; (a) $L = C = x$, (b) $C = 1, L = x$.

In the first experiment, only the results on the top rank ($L = C = 1$) of each classifier were considered when combining the results (Table 3 (a)). Compared to the system without error correction the acceptance was improved by 7.3% without adding any additional error. In this experiment 251 out of 628 checks underwent correction; 108 of them (43 %) were accepted after reclassification of the localized potential error.

In the second experiment, the top 2 ranks of the legal amount recognizer and the top rank of the courtesy amount recognizer were considered in the combination (Table 3 (b)). With this combination strategy the error rate of the original system was 0.4%. With error correction the improvement in the acceptance rate was 3.2%. Also in this experiment no additional errors were produced, but the improvement was not as high as in the previous experiment.

6 Conclusions

In this paper we presented some key ideas of a check reading system that is currently under development. We focused on the reading of legal amounts and post-processing of the recognition results. For the legal amount recognition we proposed an approach based on Hidden Markov Models. In our system HMMs turned out as an excellent tool, since for German check amounts a segmentation of the legal amount into basic words and their subsequent recognition is very difficult, if not impossible. But also for other languages, where the gaps between subwords are also hard to detect, HMMs' holistic recognition approach is very suitable. In contrast to many other works in this field we didn't use structural features for the description of our words, but the pure bitmap. The bitmap features can be extracted very easily. However, a good normalization by means of preprocessing is required.

Experiment 1: L = C = 1					
	accepted	rejected	error rate	potential errors	
original system	58.1%	41.9%	0.2%	examined	corrected
with error corr.	65.4%	34.6%	0.2%	251	108

(a)

Experiment 2: L = 2 / C = 1					
	accepted	rejected	error rate	potential errors	
original system	65.6%	34.4%	0.4%	examined	corrected
with error corr.	68.8%	31.2%	0.4%	162	47

(b)

Table 3. Rejection and error rate of the system with the new error detection and correction module

For the subtask of translating the legal amount into a digit string, which is needed for the combination of the recognition results, we presented a new, systematic technique based on syntax-directed translation. This method is very flexible and can easily be changed and extended. In our system the translation provides not only the digit string as the translation of the legal amount, but also the corresponding subwords within the legal amount for each digit. This information can be used to localize potential recognition errors in the post-processing of mismatching recognition results.

We finally proposed a new approach for the analysis and correction of detected potential errors. Depending on the combination strategy the rejection rate of the system could be decreased by 17.4%, from 41.9% to 34.6%, or by 9.3%, from 34.4% to 31.2%, respectively, without increasing the error rate. To achieve these improvements only 251 and 162 subwords out of 1500 legal amounts with approximately three subwords each had to be re-estimated in the first and second experiment, respectively. Therefore the overall system throughput doesn't decrease significantly, although the second classifier has a more complex design, and works slower. The proposed architecture for error localization and correction is very general, and can be easily adapted to other check reading systems.

References

1. R. Bippus: 1-Dimensional and Pseudo 2-Dimensional HMMs for the Recognition of German Literal Amounts, *Proc. of the 4th Int. Conference on Document Analysis and Recognition*, Ulm, Germany (1997) 487–490 [188](#), [189](#), [190](#), [192](#)
2. C. Y. Suen, L. Lam, D. Guillevic, N. W. Strathy, M. Cheriet, J. N. Said and R. Fan: Bank Check Processing, *Int. Journal of Imaging Systems and Technology*, Vol. 7 (1996) 392–403 [188](#), [190](#)
3. S. Impedovo, P. S. P. Wang, H. Bunke (Eds.): Automatic Bankcheck Processing, Special Issue of *Int. Journal of Pattern Recognition and Artificial Intelligence*, Vol. 11, No. 4 and 5 (1997) [188](#)

4. T. Caesar, J. M. Gloger, E. Mandler: Preprocessing and Feature Extraction for a Handwriting Recognition System, *Proc. of the 2nd Int. Conf. on Document Analysis and Recognition*, Tsukuba, Japan (1993) 408–411 [189](#), [190](#)
5. H. Bunke, M. Roth, E. G. Schukat-Talamazzini: Off-line Cursive Handwriting Recognition Using Hidden Markov Models, *Pattern Recognition*, Vol. 28, No. 9 (1995) 1399–1413 [189](#), [192](#)
6. G. Dimauro, S. Impedovo, G. Pirlo and A. Salzo: Automatic Bankcheck Processing: A New Engineered System, in Reference 4, 465–504 [190](#)
7. S. Knerr, V. Anisimov, O. Baret, N. Gorski, D. Price and J. C. Simon: The A2iA Recognition System for Handwritten Checks, *Proc. of Int. Workshop on Document Analysis Systems*, Malvern, Pennsylvania (1996) 431–494 [190](#)
8. J. Makhoul and R. Schwartz and C. LaPre and C. Raphael and I. Bazzi, “Language-Independent and Segmentation-Free Techniques for Optical Character Recognition, *Proc. of Int. Workshop on Document Analysis Systems*, Malvern, Pennsylvania USA (1996) 99–114 [190](#)
9. G. Kaufmann, H. Bunke, T. M. Ha: Recognition of Cursively Handwritten Words Using a Combined Normalization / Perturbation Approach, *Proc. of the 5th Int. Workshop On Frontiers in Handwriting*, University of Essex, England (1996) 17–22 [191](#)
10. L. R. Rabiner, B. H. Juang: An Introduction to Hidden Markov Models, *IEEE ASSP Magazine* (1986) 4–16 [192](#)
11. Thien M. Ha, M. Zimmermann, H. Bunke: Off-line Handwritten Numeral String Recognition by Combining Segmentation-based and Segmentation-free Methods, *Pattern Recognition*, Vol. 31, No. 3 (1997) 257–272 [193](#)
12. G. Kaufmann, H. Bunke: Amount Translation and Error Localization in Check Processing Using Syntax-Directed Translation, *Proc. of Int. Conf. on Pattern Recognition*, Brisbane, Australia (1998) 1530–1534 [193](#)
13. G. Kaufmann, H. Bunke: Error Localization and Correction in Check Processing, *Proc. of Int. Conf. on Frontiers in Handwriting Recognition VI*, Taeduck, Korea (1998) 77–87 [194](#)