Table of Contents

Reading and Learning - from Document to Knowledge

How Postal Address Readers are Made Adaptive

Klaus Kreuzer, Udo Miletzki, Hartmut Schäfer, Marc-Peter Schambach, Matthias Schulte-Austum

Siemens Dematic AG, Konstanz, Germany,
Klaus.Kreuzer@siemens.com, Udo.Miletzki@siemens.com,
Hartmut.Schaefer@siemens.com, Marc-Peter.Schambach@siemens.com,
Matthias.Schulte-Austum@siemens.com,
WWW home page: http://www.siemens-dematic.de

Abstract. In the following chapter we describe how a postal address reader is made adaptive. As the postal address reader is a hughe application we concentrated at some of the adaptive technologies are used to adapt an postal address reader to a certain task. In particular it is the adaptation of the region of interest (ROI) detection and classification, the incremental adaptation of the single character classification, the adaptation of a hidden markov recognizer for hand written words and how to improve the address dictionary of the reader. All described techniques gets used for all postal address reading applications, which are parcel, flat, letter and in-house mail sorting.

1 Introduction

The address reader from Siemens Dematic AG has been designed with the goal to virtually read all kind of addresses in any country of the world. However, to fulfill this ambitious target, a set of tasks had to be solved: for example, the location of the address block within a complex and changing text-image-graphics layout has to be found, which is a difficult task, which needs statistical modeling; within the address block, the structure and the style of handwriting show a great variety, which can only be coped with by introducing adaptive models; the address logics and the dictionary structures need to be adaptive, and so forth. Indeed, the address reader needs to be highly adaptable in many respects. This section describes the solutions for the individual components as well as for the system as a whole and it also shows, how this goal, which is a precondition for world wide business, has been reached on the whole.

Of course, a highly adaptable address reader needs a reasonable amount of adaptation effort for each address type. As the adaptation costs must be spent regardless of the customers need for just one address reader or hundreds of them, it is especially highly expensive for small countries with small number of readers. Thus one of the main goals of Siemens Dematic AG is, and will be, to make the adaptation process cheaper by automation, ultimately resulting in a self-learning system, able to improve itself by continuously learning from daily live mail. During Adaptive Read, Siemens Dematic AG has made a big

step forward towards this goal. An overview of the techniques to automate the adaptation process will be presented in this chapter.

Besides the cost reduction of the adaptation process, the postal address reading market requests a bunch of new features such as "forwarding" and "revenue protection". "Forwarding" means to detect as early as possible obsolete addresses from recipients whose address has changed, and to send the mail directly to the new address. "Revenue protection" means to detect mail pieces with missing or wrong franking and to treat these mail pieces according to certain given rules. For both features, the address reader needs to be able to read much more than just the standard target address block: It is also necessary to read the full senders address including names and nicknames and to determine the mail class. To do this, the address reader has to deal with different types of franking (stamps, barcodes), it needs to recognize a huge amount of different remarks that may occur at a mail piece surface. For this purpose, during the Adaptive Read project phase, the address reader was extended by a set of new recognition algorithms and a much more flexible interpretation and control algorithms. At the end, the result is a highly adaptable reader, which is able to perform "multi-Lingual" reading, "revenue protection" and "forwarding", and many more. Some of the most remarkable goals that have been reached during Adaptive Read will also be described in the following sections.

The state-of-the-art SIEMENS DEMATIC Reader Systems typically are optimised backstage once before delivery using a set of well-related adaptive algorithms and a set of well defined learning data, extracted from the daily mail stream. Various optimisation methods have been developed so far for the individual steps of reading, the so-called pre-processing, the recognition, and the post-processing. They matured over the decades and have been substantially refined during first *READ* R&D programme, funded by BMBF in 1997 – 1999.

Yet, what they lacked, was their adaptiveness on-site, i.e. the learning algorithms directly implanted in the product itself, so that they would be able to learn from weaknesses and errors all over their life cycle. The reason for the late introduction of this improvement was the heavy structural alteration of the system that was necessary, and the big effort associated with it. Therefore this major improvement step was reserved to *Adaptive READ*, the results of which are explained here in detail.

Within this project, the generic reading system as a whole was analysed and investigated in respect of adaptive-ness. Wherever constant parameters or fixed rules for decision processes were found, they were changed into variables and made available to learning algorithms, which optimise them during the learning phase in terms of error rate, and recognition rate, and thus contributed to the general learning aptitude of the reader product.

Following the main stream of data processing, first, new methods for adaptive object finding and segmentation were investigated: Objects are no longer exclusively target addresses, as described above, but also senders addresses, which are often necessary to read in case of automatic forwarding, stamps, indicia, logos etc., in short, all iconic objects that contain information relevant to the

postal mail processing. Learning stamps and indicia means being able to control postage charges and thus support the postal services to protect their revenue. Learning Icons may help identifying the sender, if the senders address is not written explicitly on the envelope. Objects are basically every kind of blocks on the mail surface except advertisement up to now.

In future, maybe also mailing of un-addressed mail will make it necessary to recognize the type of advertisement and to use the reader system inversely as a printer printing the appropriate address on the un-addressed mail according to individual profiles of the postal end customers. Thus every customer gets only his special selection of advertisement and will not be bothered with unwanted ads. This is another opportunity to come with the results of Adaptive READ. As one can see, the scope of how and why preprocessing is made adaptive, is fairly large.

Once the relevant objects are detected, we have to differentiate between image blocks containing text and image blocks containing other symbols, graphics or pictures. All text blocks have to be converted into a string of ascii-codes or unicodes or equivalent other codes for the characters inside, in other words, these image blocks have to be converted into their meaning by use of OCR-classifiers. These statistically based classifiers typically belong to the core method for the general conversion from pixels to meaning, and contain several hundred thousand s of free parameters, which were defined in an off-line learning process during development phase. This procedure works quite well for a while, until the set of new fonts and character shapes prevails. Then an update will be urgently necessary too late to be optimal, and resembling a typical two-threshold or bang-bang control: the degradation of the system must take place before an action will be taken.

But with the aid of Adaptive READ, the Stepwise Improvement Classification method, or short: the SIC method, has been developed for on-site applications, a new approach, which has the advantage to do adaptation in small steps, character by character, and can be interrupted at any time, learning new shapes character by character, instead of the time consuming, complex and expensive direct method, which needs to generate moment matrices and subsequent matrix inversions, thus learning all characters samples at once. The section for character classifier adaptation shows in detail, how character recognition can be made adaptive on-site.

The above mentioned method is applicable, wherever text can be properly segmented into its elements, the character. But if this is not the case, which mainly is true for script words as well as for non-segmentable machine printed words, then another method is necessary, which can recognize words or phrases like *Stratford upon Avon* as a whole. This works quite well with more or less hand printed characters or neatly written script.

However, state-of-the art word recognizers work with models of fixed structure, no matter how simple or complex a character structure is. This handicap gets evident especially when trying to read natural handwritings. Therefore to overcome this drawback, an extended Hidden Markov Model based method,

called HMAM, was developed which is able to model words automatically using adaptive model structures for the individual characters, the number of states used depending on their complexity. By this way, word recognition is made more attentive to idiosyncrasies of human writers, making them readable.

Another important adaptation field lies in the Dictionary correlation: Even if all characters and words within an address have been recognized perfectly, the result will be rejected, if it cannot be verified as a unique and valid address by use of a dictionary. Dictionaries are usually generated according to a rule base and are periodically updated. However, human writers will not always obey to these rules and write addresses with certain deviations. Here a new dictionary learning method is introduced, which is able to find from daily mail stream the most frequent deviations, the favorites. In case of uniqueness, they can be accepted and lead to a successful read result. This is one way how address interpretation is made adaptive.

Last but not least, it is not enough to make all the components of the system adaptive. They must also communicate with each other concerning overall parameters and they must be adapted and tuned as an ensemble, otherwise the components may be better, but the system performance nevertheless goes down. Therefore, a method for optimization of a cascade of components of a complex system has been developed to ensure best possible system performance. This is the way, how the total system is made adaptive.

The readers adaptive-ness is proven by operative applications throughout Europe, in Australia, New Zeeland, USA and even Cyprus with a Greek alphabet, Dubai with an Arabic alphabet and in several countries in the east of Europe with a Cyrillic alphabet. These non-Latin alphabets typically need to be combined with a Latin alphabet, because in these countries, Latin is used as well, especially for international business mail. Thus the reader gets multi-Lingual, a process to which Adaptive Read project contributed an important part.

Although the postal address reader from Siemens Dematic AG is already highly adaptive and the adaptation process can be automated to some degree, Siemens Dematic AG will continue to make the adaptation of the postal address reader more comprehensive, cheaper and faster, ultimately resulting in a self-learning system. This will remain a challenge for the next years.

2 How preprocessing is made adaptive

2.1 Abstract

One of the most challenging tasks in modern document analysis on complex postal images is the detection and identification of all postal relevant regions. Such relevant regions (Regions of Interest - ROIs) are for example text areas like the receiver address block or the sender address block or indicia objects like stamps or meter impressions.

To identify those regions correctly one has to consider different type of mail-piece layouts with its inherent relation ships between the regions. To cope with

this virtual infinite amout of different layouts a adaptive technique has been developed which allows the semi-automatic training of the region evaluation system. This technique is described in this article.

2.2 Block generation

A preliminary step for block evaluation is the text block generation. Inside the image processing component which delivers the regions of interest (in the following named as Region of interest function or ROI) there is a text block generation component which delivers all text blocks.

After the generation of text blocks these blocks are evaluated by generating features for each of these blocks. A simple approach is to use a rule system to select the correct target address area. For example "take right block before take the left", "take lower block instead of the higher block" or "the block in the top left corner is the receiver block" are some rules which have been quite good for the layout of standard mail envelopes.

Those standard rule result are only correct to a certain extend. The amount of effort extending and maintaining such block evaluation rules increases with the amount of different mailpiece layouts. Some existing mailpiece layouts even have no obvious rules.

2.3 Generating statistical data

To overcome the problem of adapting rules to a infinite amount of mailpiece layout a basic statistical instrument for evaluating blocks has been developed. Large samples of test images have been recorded (>10000 images) and the block generation system has calculated the address block positions and features over this data. The statistical data for each of the used text block features is transformed into a feature diagram where the probalities for each of this features are documented.

Basic text block features are:

1. position on the mail piece on a 8x8 grid
2. block is left, centered, or right aligned.
3. ratio of width to height
4. block is handwritten, or typewritten, has constant spacing or not
5. number of lines inside the block
6. has a lower right neighbor

The features are separated by brackets and inside the brackets the feature probabilities are separated by a comma.

2.4 Block evaluation

With the trained block evaluation statistics histogram it is possible to calculate a probability for each text block to be a receiver address block and hence with

the comparision of all candidates to get an optimum in the maximization of the read rate and a minimization of error rate.

The block evaluation starts with the calculation of the basic features for all text blocks generated by the block generation. Those features give a vector, which is classified using the statistic histogram. The resulting confidence is compared with a given minimum threshold. If the confidence is higher, the block is selected and returned as one candidate of the (usually target) address block. If the confidence is below this threshold the text block will be rejected.

The resulting probabilities of the remaining text blocks are compared then the block with the highest probality will be the final address block and will be send to further OCR processing steps.

To handle multiple type of layout there was the need to discriminate standard layouts and complex layouts. For standard layouts the probality of an address block located in the top 2/8 of a mailpiece is very low and in our current configuration set intentionally to 0, so that text blocks which reside in this area will get 0 probality and therefore getting rejected. The reason for that setting is the reduction of sender errors, because sender address blocks reside often in the top area of the mailpiece.

If there are existing two blocks and the correct one is in the middle or to the right of the sender address, in the very most cases the correct one gets the higher probability and will be selected. But it may happen that a very bad or weak (in binary image) written text block is not generated as a text block and the only text block generated is the return address. For these cases we need the ability to reject a single block only due to the typical sender address position.

You can look in the configuration of the ROI where some histo*.cfg files exist. For standard flats and machine style written text blocks the file "histo_flat_standard.cfg" is applied. Look in in the top two matrixes of 8x8 value. The first matrix is for blocks in horizontal positions and the second matrix is for blocks in vertical orientation. Relevant for the selection of the position histogram is the orientation of the analyzed text block. The mailpiece is divided into a 8x8 grid and the center of the analyzed address block is used to go inside this grid and determine the probability. For horizontal blocks the top 2/8 of the mailpiece and the left 1/8 of the mailpiece are zones where all detected blocks are rejected. For vertical block this behaviour is a bit different.

For following types of mailpieces separate configurations are being used:

- Letter
- Flat
- Parcel
- Hand written
- Machine print
- Newspapers

2.5 Training the system

With this method of using statistical data it is possible to train a block evaluation system semi-automatic. A typical training or tuning of such a system follows the following steps:

1. Capture a typical representative sample of images (mailpieces with different layouts)
2. Run a standard OCR engine over these images
3. Generate statistical data over the block features of the mailpieces
4. With the aid of reference data select the images with correct read results
5. Transform this statistical data into a loadable histogram
6. Run again the OCR engine with a block evaluation controlled by this new histogram

The tuning process may include some iterations by changing the histogram and running again the OCR engine up to the system performs well.

2.6 Conclusion

The invention of this adaptive block evaluation has speeded up the adoption of the OCR engine to new layouts. It has also improved the read and error rate in most applications. One great drawback of this system is the not always obvious decision of the block evaluation. Compared to strictly rule based system, this block evaluation may decide in some rare examples against some obvious rules! A typical rule is "take right text block before the left block". In most cases the statistical block evaluation will decide according this rule but if for example the summarized feature probability of the two blocks are very close together another feature may overrule this rule. One solution for this problem was the handling of different layout models with different statistical block evaluation engines.

3 How character recognition is made adaptive

In a certain sense character recognition is already adaptive by definition. Then how could it be "made adaptive"?

3.1 What does it mean "Character Recognition is Made Adaptive"?

Character recognition is the process that takes as input a character image and gives as output the recognition of it, i.e. the character meaning. The process has a certain amount of free parameters for this purpose. These parameters need to be tuned in order to fulfill the performance requirements for the process. The tuning *is* the adaptation to meet these requirements. In this sense character recognition is already adaptive.

Some common requirements are a high recognition rate, a low error rate, a low consumption of computer resources, i.e. processing time and memory, a set of classes that are to be recognized.

So, what is meant in this context by "character recognition is made adaptive"?

The tuning/adaptation of the parameters of the recognition process is itself a process. It takes as input character images from a learning set where the meaning of an image is known and gives as output adjusted parameters for the recognition process. When does this process takes place?

In case the learning set is available it can take place before the recognition process starts, i.e. offline learning. The parameters are adjusted once and the recognition process can use them unchanged.

Character recognition is "made adaptive" means in this context that an adaptation process has to be made a part of the recognition process, i.e. online learning. The parameters can be adjusted during the recognition phase.

3.2 What is to be done to be "Made Adaptive"?

Both the adaptation and the recognition process need to be redesigned. The adaptation process now must also comply to the requirements of the recognition process and the recognition process must also provide the input of the adaptation process.

The input of the adaptation process are character images with their correct meaning. The recognition process gets images and returns a meaning for each image. But is it the correct meaning? There has to be another process in order to provide the correct meaning or to decide if the meaning of the recognition process is trustworthy enough to be learned/adapted.

Such a process could use context information, dictionaries or even human expertise (video coding).

If there are time and memory constraints for the recognition process then they must be considered by the adaptation process. It will not be possible any more to collect and hold all the statistics of all classes from the learning set and add the new learning element to approximate the optimal parameters for this enlarged learning set. What is needed in this case is a quick and cheap way of learning.

3.3 An Answer to Quick and Cheap Learning: SIC

Quick and cheap in this context means moderate CPU and memory requirements. The idea is to take the existing parameters and alter them appropriately by considering the new image with its meaning as if it had been part of the learning set. J. Schürmann described this idea as *Recursive Learning* [8].

This incremental way of learning is implemented in the SIC-method (Stepwise Iterated Classifier). In most applications the SIC-method is quick enough to run during the recognition process and is moderate enough considering the memory resources.

The SIC-method can enhance the recognition rate and can reduce the error rate on the learning set but is not as good on unlearned test sets as the

direct method. There are some parameters of the method which have to be chosen carefully to avoid overlearning and unlearning of classes. It depends on the frequencies of the different classes within the learning set.

3.4 Applying SIC to Complex Structured Classifiers

The parameters for the recognition process can be put ensemble in a simple way to a plain classifier. In the initial adaptation process the learning set is arranged to guarantee that all classes are well distributed over the learning set to comply with the requirements of the SIC-method. Plain classifiers adapted or iterated by this method show good results.

Performance requirements lead to the introduction of more complex structured classifiers.

A tree structure e.g. is able to handle larger sets of classes faster. In each node of the tree the decision is made if the presented image belongs to a certain subset of classes and for this subset to which subsubset and so on until the class is decided (divide and rule).

A net structure is useful for high recognition rates when time restrictions are not so strong and the number of classes to recognize is not too high. The decision is derived from the decisions of a set of classifiers with two classes.

Applying the SIC-method to these complex structured classifiers did not result in the expected improvements at first. There were some urgent questions to be answered. Is the SIC-method appropriate for this kind of classifiers? Is there a bug in the implementation? Or what?

3.5 A New Approach Applying SIC to Complex Structured Classifiers

Tree and net classifiers are a composition of plain classifiers. To apply the SIC-method to these complex structured classifiers you have to apply it to every plain classifier of the composition. If each plain classifier improves the whole classifier should improve. There is no reason why the SIC-method should not work as well for the complex structured classifiers as for the plain classifiers.

Software is suspected to be buggy more or less. Perhaps there is a problem with the implementation? The core of the algorithm proved to be working fine for plain classifiers. The code to decompose a complex classifier into plain classifiers and the recomposition after the iteration is easy to verify and showed no hint of a problem.

So what is the reason for not showing improvements when applying the SIC-method to complex structured classifiers?

The idea came that since the SIC-method is sensitive to the arrangement of the elements of the learning set there has to be a special arrangement for every plain classifier of the classifier composition.

To verify this idea the software had to be redesigned.

3.6 Software Redesign Needed

Since the SIC-method depends on the frequencies of the different classes within the learning set there has to be a function that distributes the different classes over the whole learning set.

This function was called before the decomposition of the complex structured classifiers. Now it has to be called for every plain classifier of the classifier composition. But this is not enough. It had to be made possible to choose the way it should do the rearrangement.

Think e.g. of a tree classifier for digits. A node has to decide between two subsets of digits . Let `l` be the left subset the digits {`0 1`} and let `r` be the right subset {`2 3`}. Finally let the learning set be sorted which is not good for the SIC-method {`0 0 1 1 2 2 3 3`}. The classifier has to learn these elements as `l` and `r`. Therefore they have to be renamed. To keep track from which meaning they were renamed we use subscripts as in $\mathtt{l}_0$.

One can think of two ways to rearrange the learning set for this node. Rename the digits first which gives {$\mathtt{l}_0$ $\mathtt{l}_0$ $\mathtt{l}_1$ $\mathtt{l}_1$ $\mathtt{r}_2$ $\mathtt{r}_2$ $\mathtt{r}_3$ $\mathtt{r}_3$} and then arrange `l` and `r` elements to be well distributed which gives {$\mathtt{l}_0$ $\mathtt{r}_2$ $\mathtt{l}_0$ $\mathtt{r}_2$ $\mathtt{l}_1$ $\mathtt{r}_3$ $\mathtt{l}_1$ $\mathtt{r}_3$}. One observes that in the second half there are no elements which came from {`0 2`}.

The other possibility is to distribute the elements which gives {`0 1 2 3 0 1 2 3`}, rename them {$\mathtt{l}_0$ $\mathtt{l}_1$ $\mathtt{r}_2$ $\mathtt{r}_3$ $\mathtt{l}_0$ $\mathtt{l}_1$ $\mathtt{r}_2$ $\mathtt{r}_3$} and finally rearrange them {$\mathtt{l}_0$ $\mathtt{r}_2$ $\mathtt{l}_1$ $\mathtt{r}_3$ $\mathtt{l}_0$ $\mathtt{r}_2$ $\mathtt{l}_1$ $\mathtt{r}_3$}. Here we find every class in both halfs of the learning set sequence.

It turned out that the second variant showed better results.

3.7 Summary and Outlook

It has been shown that to make character recognition adaptive there is the need for a quick and cheap adaptation method. The SIC-method is suitable for this task and can be applied to simple and complex structured classifiers if one takes care of arranging the learning set elements well.

Further investigations are necessary on how this restriction can be considered during the recognition process. It cannot be expected that the incoming images are distributed well enough to be used directly. The process that decides what elements are trustworthy enough to be adapted could also take care of this restriction.

The new approach of the SIC-method is an important stepping-stone in order to make character recognition adaptive.

4 How word recognition is made more flexible

When building a system for cursive handwriting recognition, an important step is finding the right *model* that best describes cursive script. Many recognition systems are based on Hidden Markov Models (HMMs). In these systems, much attention has been paid on training of HMM *parameters* to gain best recognition

performance. To improve the flexibility of word recognition, the adaptation of the underlying HMM *structure* has been added to the training of the system (Fig. 1). Modelling the structure is the first step in building a recognition system and is normally done manually. In the system presented, this task has been automated.

One decision when modeling script concerns the question of how many writing variants, or *"allographs"* of each letter have to be considered to get a model representative for all occurring writing styles. Mostly, these decisions are made manually, based on assumptions that are made about the writing. This way, upper- and lowercase letters are distinguished often, as well as hand block and cursive writing styles. But a good model has to consider those variants that really occur, especially if there is no detailed knowledge about writing styles. Another decision is about the number of HMM states that model each variant. It mirrors the complexity of the letters.

It is useful to determine the writing variants and complexities automatically, especially in postal automation systems, where recognition systems specific to different countries with different writing styles and even alphabets have to be developed. The HMM structure is determined by analyzing the same sets of training data which are already available for setting the recognition system parameters.

4.1 Recognition of cursive script with HMMs

The script recognition system is based on linear left-to-right HMMs, with a semi-continuous, tied-mixture probability modeling structure [1]. The script model is defined by a set of *graphemes* (letters, numbers and special characters). Different writing variants of a grapheme – *allographs* – are combined in a *multipath letter model* with multiple, parallel state paths (Fig. 2).

The system is applied in postal automation systems to recognition tasks in cursive script, hard-to-segment hand block and machine print, and Arabic script recognition. For testing the algorithms, eight configurations from different countries have been selected: Canada, Germany, the USA, and the United Arabic

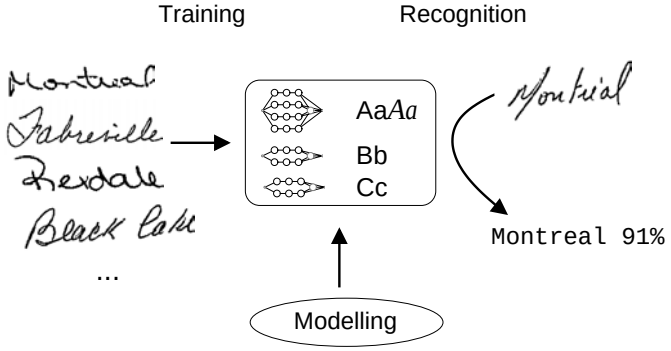

Fig. 1. Flexible word recognition by modeling of the HMM structure.

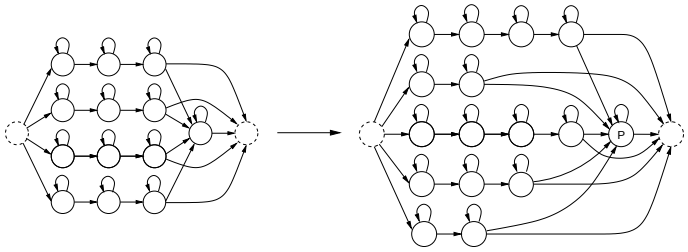

Fig. 2. Improving the flexibility of HMM model structures

Emirates. Word recognition tests have been performed with dictionaries of sizes of about 150 words for country, city and street names, and about 40000 words for ZIP codes.

4.2 Model length adaptation

Trying to find the optimal state number for each of the letter HMMs, one is faced with the following problems: Under the condition that no letter boundaries are labeled in the training data, every change in model length influences the other models during training. Adapting the number of states of each model separately also results in high computational costs. An algorithm has been developed that is able to adapt all models at once in one iteration, so only a few iterations are necessary [3].

Algorithm The model length adaptation is done iteratively, and all letter models are optimized at the same time. The starting point is a trained standard system with default path lengths. In every iteration, different model alternatives are generated, all models are trained, and finally the best model is chosen. The following steps are performed:

- In case of multipath models, assign fixed writing variants to the training data and separate the models into singlepath models.
- Expand each allograph model by adding parallel state paths with ± 1 states (Fig. 3). The new state paths do not represent writing variants but realize alternative model lengths.
- Perform parameter training (*Baum-Welsh* algorithm [4]) with the expanded models.
- Select the "best" path, representing the new model.
- Retrain the HMM parameters, including a new vector quantizer, without fixed assignment of writing variants.

Two criteria control the termination of iterations. The first criterion is that no more improvement in likelihood of training data takes place. To avoid being stuck in suboptimal configurations, a second criterion has been added, allowing

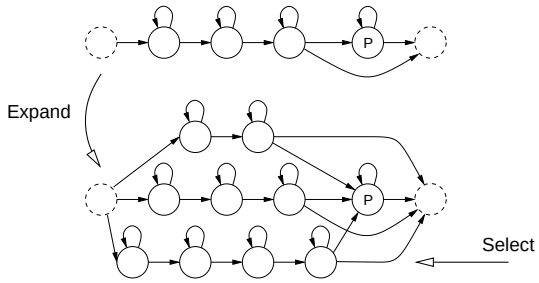

Fig. 3. Learning the model length

to stop only when additionally no further change in model topology occurs. Better recognition results have been achieved this way.

Path selection criteria The choice of the "best" path during iterations has great impact on the algorithm performance. A two-step mechanism has been developed. First, the state path which has been selected most frequently during training is chosen; this is indicated simply by the transition probabilities to the first states of each path. Then, after retraining the paths selected this way, the average likelihood for the new letter models is calculated. Only in cases when this likelihood is improved, the new model length is kept. The second step is important to gain better recognition performance.

Results By applying the algorithm on the tested systems, the average length of model state paths is increased. As the modeling gets more fine-grained and better adapted to the data, the recognition performance is improved by about 2.1 %. Analysing the relation of performance on training and test data, an increase of the generalization capabilities of the system can also be seen: Results on independent test data converge to those that are achieved on training data.

4.3 Allograph adaptation

To adapt the HMM script model automatically to the best number of writing variants, a similar, iterative algorithm has been developed.

Starting from a trained system with default model, the allograph clustering algorithm iteratively adds and removes allograph models. In every iteration, the following tasks are performed:

- Select allograph state paths for modification. Different strategies have been developed and tested. They are described in detail further down.
- Modify the selected allographs. Remove a state path by deleting or mixing similar paths, or add one by modifying a selected, existing path.
- Retrain the HMM, including codebook calculation.

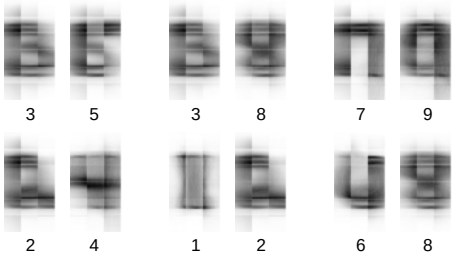

Fig. 4. Visualization of HMM-Parameters for model pairs with smallest (top) and highest distance (bottom) in a recognition system for German ZIP codes

Iterations are terminated when a maximum or minimum size of the allograph model is reached, or when a given maximum of iterations has been performed. The final script model is selected by Bayesian model selection criteria.

In the ongoing section, the different model selection strategies (used within the iterations) are presented and discussed. Then, the final model evaluation method is presented. For practical use, a variation of the algorithm will be developed, and finally, results are shown.

Strategies for selection and modification Four different strategies for allograph selection and modification have been developed. A subset of three is used for adding and another subset of three for removing allograph models.

Model similarity – Determining the writing variants in script samples means clustering the training data. For clustering, it's necessary to define a distance between objects, in this case between allograph HMMs. The distance proposed is calculated for left-to-right allograph models, but it can also be applied to multipath letter models. The calculation method is called *Statistical Dynamic Time Warping* and is described in [2]. It is possible to calculate distances between models even if they have differing numbers of states. Similar allographs within a single letter model represent the same data and are therefore candidates for merging. Using the proposed distance measure, the grapheme model with the pair of paths closest to each other is chosen and the respective paths are joined.

The quality of the distance measure proposed is examined by visual inspection. A method for visualization has been developed and is presented in [3]. Fig. 4 shows some examples of letter models with smallest or highest distance; the results correspond quite well to human perception of similarity. Additionally, it could be shown that the recognition system also confuses similar models more frequently.

Model quality – To test whether it is useful to model allographs, an independent measure for model quality has been developed. For semicontinuous HMM systems, an easy measure is the entropy of emission weights. No properties of the classes themselves are considered, but for well separable classes, the value

indicates how well defined the model states are. The letter model quality is defined by averaging the entropy of all states. The worst model, indicated by the highest allograph entropy, is chosen to have an additional writing variant. As initialization, the best allograph, i.e. the one with lowest entropy, is doubled, and emission weights are shifted randomly.

Model likelihood – The effect of grapheme modeling within the recognition system can be given by model likelihood. This is the contribution of a particular letter model to training data likelihood, and it can be calculated from the Viterbi path, where likelihood is assigned to each model state. To remove an allograph, the one with the overall worst contribution in likelihood is chosen. Model likelihood also can be used to add allographs: the grapheme with the average worst likelihood of all allographs is chosen, and the allograph with the highest likelihood variance is doubled.

Amount of represented data – A pragmatic approach to improving the recognition system selects model detailing by frequency in training data. Graphemes that appear more often have higher influence on overall performance, and are modelled with more allographs. The allograph frequency is defined by the product of absolute number of the grapheme in training data and the transition probability to the first state in the allograph state path. Allographs with low frequency are removed; those with high frequency are doubled.

Discussion of selection strategies For each of the eight configurations tested (see section 4.1), experiments for all six selection and modification strategies have been performed [5]. The inspection of the images of the final grapheme models gives interesting insights into the properties of the different selection strategies. When adding allographs, selection by entropy gives the best visual impression. Adding allographs by frequency results in best recognition performance, but the modeling is non-intuitive. The best mixture of performance and good visual impression is given by selection by likelihood. Regarding the iterations where allographs are removed, the best visual impression combined with good recognition performance is obtained by removing by distance.

Evaluation of models: Bayesian selection criteria After adaptation iterations, the best script model has to be selected. Generally, when trying to find the "right" model, the danger of over-adaptation to the available data exists. In our case, more allograph models could result in worse generalization to test data. Bayesian model selection criteria are a general approach to deal with this subject. Different selection criteria like the *Akaike Information Criterion*, the *Bayesian Information Criterion* and *Cheeseman-Stutz Approximations* [6] have been applied. They are differing slightly in the terms penalizing high model complexity. Fig. 5 shows that the criteria predict quite well the generalization capability of the models.

Fixed complexity However, none of the criteria could really be applied to final model selection, because generated models have been too big regarding time and

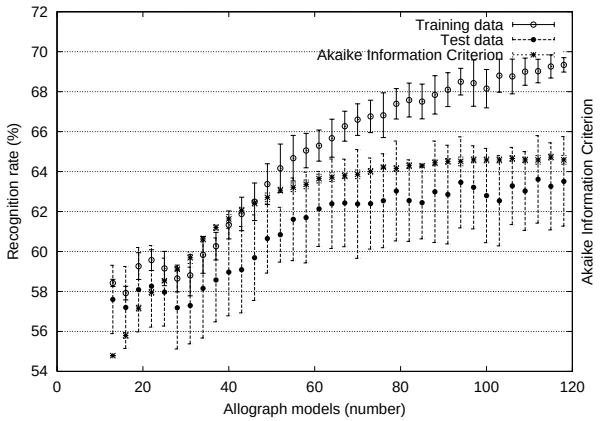

Fig. 5. Saturation of test recognition rate when continously increasing model complexity. The *AIC* predicts the effect quite well

Table 1. Word recognition rates (forced recognition) for different adaptation methods

Project	Baseline	Length only	Allographs	Combination
Arab Emirates	76.9 %	85.5 %	86.4 %	86.4 %
Canada (address)	93.4 %	93.4 %	95.4 %	95.5 %
Germany (address)	92.8 %	93.3 %	94.5 %	94.7 %
Germany (ZIP)	69.1 %	72.1 %	78.3 %	76.0 %
USA (address)	81.2 %	82.7 %	84,6 %	84.4 %
USA (ZIP)	60.8 %	62.9 %	70.7 %	71.4 %
CEDAR (cities)	84.2 %	84.2 %	85.1 %	86.3 %
CEDAR (ZIP)	58.2 %	59.5 %	68.3 %	71.7 %
Average	77.1 %	79.2 %	82.9 %	83.4 %

memory constraints of real world applications. Therefore, the adaptation process had to work with a given, fixed overall model complexity, allowing only zero-sum changes between models. An adaptation algorithm with alternating steps for adding and removing allographs has been developed. The selection criterion was given by a ranked list of likelihood, distance and frequency, to take full advantage of all different selection strategies.

Experimental results Recognition rates are shown in Table 1. Eight recognition systems have been adapted and tested with independent data. With adaptation of model length only, an improvement of 2.1 % could be obtained, while allograph adaptation alone gave an improvement of 5.8 %. Both methods have been combined by serial execution, which resulted in an overall improvement of 6.3 %. Thus, the approaches proved to be "orthogonal" (i.e. independent of each other, concerning their effects), as the different degrees of freedom in the model images (Fig. 2) also suggest.

4.4 Application to practice

The developed algorithms for adaptation of HMM model structure are used for fine-tuning word recognition systems for new and existing applications in postal automation. By determining the model structures automatically, engineering efforts for model selection could be reduced. Because improvements in recognition can be obtained keeping the overall model complexity constant, computational costs increase only in offline training, not during the crucial recognition phase. Thus, the adaptation system can be used for real world applications.

5 How address interpretation is made adaptive

Address reading systems need information on the content and syntax of addresses in order to be able to extract the required information such as town, zip code, first and last name, etc. The permissible content of individual address elements is described by means of a dictionary (list of permissible strings) which, according to the prior art, is built up from present information sources such as, e.g. from a postal dictionary or from a list of employees of a company. However, the application domain changes with time so that the dictionary created at the beginning no longer completely includes all existing contents. It is especially when a reading system is used for mail distribution within a company, that the change in the set of words is considerable: employees leave the company, new employees are added, employees change their department or last names due to marriage, etc. Thus, entries are missing in the dictionary and there are entries which are no longer valid. The more the set of words currently used deviates from the lexicon, the more the recognition performance of the reading system drops.

Previously, these changes had to be manually transferred into the dictionaries at certain time intervals so that postal organisation have to spend a lot of time and money to gain and add new addresses to the dictionary and to detect obsolet adresses.

In the following a system will be described that is able to add words or word groups to the dictionary and detect not longer used words or word groups in the dictionary in order to remove them from the dictionary. First section will give an overview over the system and its possible variants. The second part will describe a concrete system with examples.

5.1 Overview

It is the object of the described system to automatically form and/or automatically update a dictionary for reading addresses. This is based on the concept of temporarily storing the results of the current reading processes, to evaluate them and to use them for automatically building up or updating a dictionary. During the temporary storage, the respective address is marked to indicate whether it has been read successfully or whether it has been rejected because it could not

be found in the existing dictionary. If a dictionary is to be newly created or if new addressees are to be entered in the existing dictionary, the rejected reading results that could not be found in the current dictionary are utilized.

The dictionaries are directed graphs where each node contains a word of a certain type (first name, last name, street, city etc.). A path through this graph represents a valid address. In address written at letter a valid address can be sapced apart by unknown words as well as some words contained in the dictionary may be skipped by the writer.

Automatic building up of a dictionary or, respectively, automatic updating a dictionary due to new addressees or changes in the addressees is possible by forming classes of words or word groups which have a fixed minimum measure of similarity with respect to one another, and including at least some representatives in the dictionary of the associated address areas in order to add it to the correct place in the dictionary.

An important step in processing is to form classes of not correlated words or word groups in order to add only the correct written word or word group and not those words or word groups with different wrong characters in it. To do this it is advantageous to create a list of all words or word groups of not correlated reading results which are sorted in accordance with their frequency. Beginning with the most frequent word or word group, the factor of similarity with all remaining words or word groups is determined and entered in a similarity list. All words or word groups in the similarity list having a similarity factor above a fixed threshold are then allocated as class to the current word or word group. After that, the words or word groups of the class formed are removed from the frequency list. In order to find the correct written word or word group in the class a representative is determined. The representatives of the respective class of words or word groups of the reading results temporarily stored and rejected can be formed by the shortest or most frequent word or word groups.

To recognize words or word groups in the dictionary which must be changed or removed, it is advantageous to statistically analyze the addresses read unambiguously, this means words or word groups that have been correlated to the dictionary. If there is an abrupt change in the frequency of correlation of words and/or word groups from the dictionary beyond a particular threshold and if it persists for a predetermined time, these words or word groups are removed from the dictionary.

To avoid irrelevant words of the reading results from being included in the dictionary, they can be determined by comparison with words stored in a special file for irrelevant words. It is also of advantage in this connection not to include short words of less than p letters and without fullstop as irrelevant in the dictionary.

To perform the address interpretation in as detailed as manner as possible with the aid of the dictionaries, it could advantageous to include, in addition to the representatives, also the words and/or word groups of the associated classes with the similarity factors and frequencies.

In a further advantageous embodiment, words or word groups belonging together. This will result in phrases of words or word groups which will help to add them to the dictionary. These phrases having n words which are mutually spaced apart by m words. They can be determined in that the addresses are searched with windows having a width of n + m words starting with the respective individual word determined for the dictionary. Once the further n + 1 individual words with the gaps of m words between them have been determined, this word group and its frequencies are included in the corresponding group dictionary.

It is also advantageous to determine the similarity factor for words and word groups by means of the Levenshtein method (see [9]).

In order to avoid errors while updating the dictionary, it can be advantageous to categorize, and to have confirmed, the dictionary updatings found by the system at a manual coding station by a human operator or to compare the new entries into the dictionary additionally, before they are taken into the corresponding category, with the contents of a file in which characteristic generally applicable names or at least strings related to the respective category (first name, last name, department) are stored.

5.2 Examplary Embodiement

In the text which follows, the system will be explained in greater detail in an exemplary embodiment and referring to drawings. The aim is to determine previously unknown last names (n = 1) or pairs of unknown first and last names (n = 2) or last and/or first and last names and department names of employees of a company and/or corresponding, no longer valid names or name combinations, and to perform dictionary changes.

The word proposals are automatically generated from the recognition results calculated for each pattern of an item, which is the corresponding snippet of an image or features derived from the image snippet, by the reading system in daily operation. The recognition results for each pattern of an item comprise different geometric objects (layout objects) such as text blocks, lines, words and characters and their relations to one another, that is to say which lines belong to which text block, which words are located in which lines etc.

For each individual character pattern, the reading system generates a list of possible character meanings. In addition, the reading system calculates for each layout object its position in the pattern of an item and its geometric dimensions. To update or even learn dictionary entries, the set of items processed is separated into two subsets, into the set of items read automatically (but not necessarily correctly) by the reading system and the set of rejected items. The set of items read automatically is used for determining dictionary entries which are no longer valid; from the set of rejected items, new dictionary entries are derived.

The exemplary system consists of five modules: a monitor process (figure 6), processing of the recognition results (preprocessing, figure 7), two dictionary generation methods (figures 9 and 9) and a proposal administrator (figure 10).

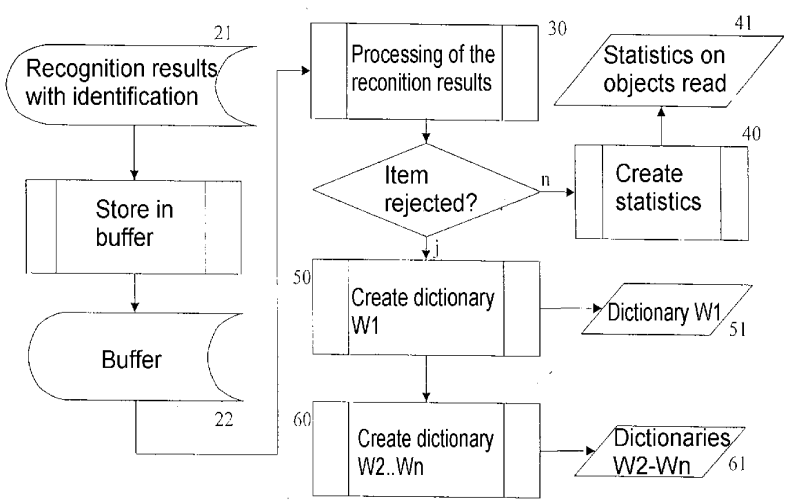

Fig. 6. flow structure of a monitor process for monitoring and controlling the updating of the dictionary

Monitor Process The monitoring process according to figure 6 monitors and controls the dictionary learning. The recognition results 21 for each pattern of an item, together with an identification for "read successfully (correlated)" or "rejected", are transferred from the reader to the monitor. Additional information on the type of item (letter, large letter, in-house mail form) and other features relating to the individual objects of the recognition results such as ROI (Region of Interest), line and word hypotheses, disassembly alternatives and character recognition results can also be transferred. These recognition results are stored in a buffer 22 in the monitor until a sufficiently large amount of data has accumulated (e.g. after 20 000 items or after one week of operation).

In the simplest case, only the first alternative of the character recognition results together with the best segmenting path is stored in a buffer. In a more complex case the complete segement graph or a sub graph with several alternative character recognition results could be uses. In the described system we implemented the simplest case. For example, the content could look as follows:

```
=====================================================================
<Recognition Results>                <Identification >
:...
1017921 PMD 55                       recognized
MR. ALFRED C SCHMIDI
EXCCU1LVE DIRCC1OR, 0PCRA1IONS
DCVC1OPMENT
```

```
MyComp, INC
1 MyStreet

MyCity, 12345

PO11Y O/BRIEN                          rejected, not in the
                                       dictionary

MANAGER, COMMUNITY AFFAIRS
MyComp INC
1 MyStreet
MyCity, 12345

PO1LY OBRIEN                           rejected, not in the
                                       dictionary

MANAGER, COMMUNITY AFFAIRS
MyComp, INC
1 MyStreet
MyCity, 12345

MS ME1INDA DUCKSWORTH                  recognized
MyComp, INC
MAI1 CODE 63-33
1 MyStreet
MyCity, 12345

*********AURO**MIXED AADC 460          Rejected, not in the
                                       dictionary

MIKO SCHWARTZ
O AND T 26-00
1 MyStreetMyCity, 12345

...
-----------------------------------------------------------------------
```

If sufficient results are available, the rejected recognition results are transferred to a processing unit 30 and forwarded to the two subprocesses for dictionary training for single words 50 and word groups 60. In the case of a successful automatic recognition, the results are transferred to a statistics module 40. When all items have been processed, the word and word group lists 41 of the statistics module and of the dictionary training processes 51, 61 are collected and presented to an operator for confirmation by means of a suitable graphical user interface.

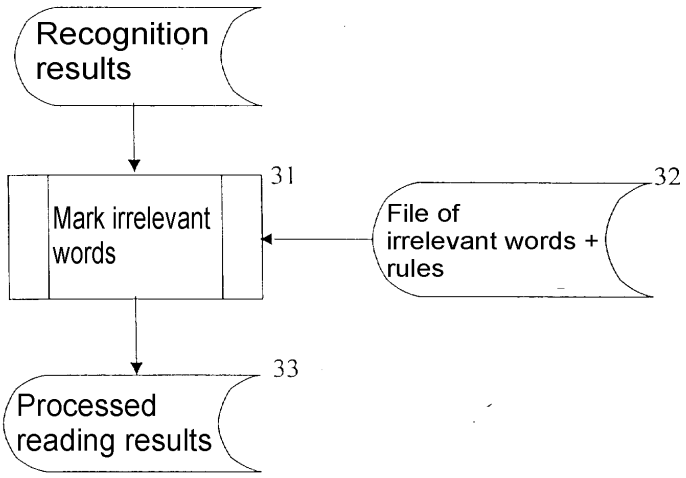

Fig. 7. flow structure for determining and marking a irrelevant words

Marking irrelevant Words In the processing unit 30, irrelevant words in the rejected recognition results are identified which are not taken into consideration in the subsequent text analysis (compare figure 7). These words are marked as not relevant but are not deleted since the word neighborhood is of importance for the subsequent building up of the dictionary.

In the method step marking irrelevant words 31, short words are marked from the set of word hypotheses, for example those words which are less than 4 letters long and, at the same time, do not have a fullstop, and those, less than 50% of whose characters are alphanumeric. Furthermore, those words are marked which are contained in a special file 32 which contains frequent but irrelevant words for this application. In the application of in-house mail distribution, for example, this special lexicon can contain the company name, city name, street name, post box designation etc. The results of the processing are written back into a buffer 33.

After the preprocessing, the results look as follows:

```
-----------------------------------------------------------------
<title MR> <first-name ALFRED> <last-name SCHMID>
<role EXECUTIVE DIRECTOR OPERATIONS>

PO11Y O/BRIEN
MANAGER, COMMUNITY AFFAIRS
<irrelevant MyComp, INC>
```

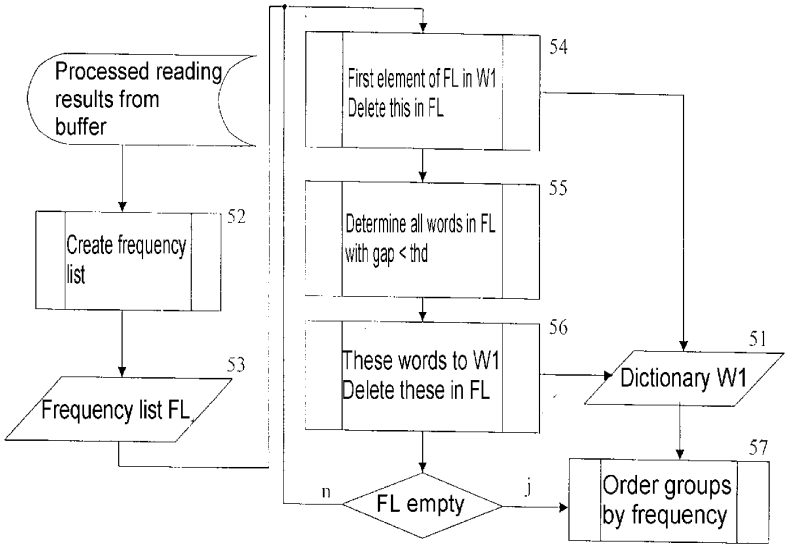

Fig. 8. flow structure for determining previously unknown single words (n = 1) (last names)

```
<irrelevant 1 MyStreet>
<irrelevant MyCity> <irrelevant 12345>

PO1LY OBRIEN
MANAGER, COMMUNITY AFFAIRS
<irrelevant MyComp, INC>
<irrelevant 1 MyStreet>
<irrelevant MyCity> <irrelevant 12345>

<title MS> <first-name MELINDA> <last-name DUCKSWORTH>

<non-alpha ********AURO**MIXED> AADC <short 460>
MIKO SCHWARTZ
<short O> <short AND> <short T> 26-00
<irrelevant MyComp, INC>
<irrelevant 1 MyStreet>
<irrelevant MyCity> <irrelevant 12345>

...
-----------------------------------------------------------------------
```

Determining previously unknown Words According to figure 8, from the processed rejected recognition results, a frequency list FL 53 of all words occurring there is created in first step 52, sorted in accordance with descending frequency and stored in a buffer. For the above example, the frequency list FL 53 could look as follows:

```
================================================================
...
AFFAIRS                    37
MANAGER                    37
COMMUNITY                  37
OBRIEN                     20
O/BRIEN                    17
SCHWARTZ                   15
MIKO                       12
POLLY                      10
PO11Y                       8
PAULA                       8
PO1LY                       5
MIKO                        3
...
----------------------------------------------------------------
```

From this list, a dictionary W1 of relevant words 51 is built up step by step. For each word in the frequency list FL 53, the distance d to all words in this frequency list is determined. One method for measuring the distance between two strings is the Levenshtein method which calculates the minimum distance between two strings referred to 3 cost categories, at the cost of replacing one character, an insertion and a deletion operation. In addition to the string, other features of the recognition result, for example the character alternatives, the segmentation alternatives, etc., can be used for calculating d.

The first word in the frequency list FL 53 (the currently most frequent one) is included in the dictionary W1 51 and deleted 54 from the frequency list FL 53. All words from the frequency list FL 53 having a distance of less than a predetermined threshold thd are allocated 55, 56 to the current word in the dictionary W1 51 with their frequency. At the same time, these words are deleted in the frequency list FL 53. The iteration stops when the frequency list FL 53 is empty. This forms word classes which do not exceed a distance d between each other or, respectively, do not drop below a corresponding similarity factor.

When all words have been processed, the dictionary W1 51 consists of a set of word classes. The shortest word of a word class is called the representative of the group. Each word class contains words which are similar to each other, with the associated frequencies and distances from the class representative. The representatives of word classes in the dictionary W1 51, and thus also the word classes, are sorted 57 in accordance with descending frequency. The frequency of a word class is composed of the frequency of the representative and the frequencies of the elements of the word class. Word classes with a frequency which

drops below a particular threshold are deleted from the dictionary W1 51. In consequence, the following dictionary W1 51 is formed from the above list:

```
================================================================
<Word class>            <Frequency>             <Distance>
...
AFFAIRS                 37
MANAGER                 37
COMMUNITY               37
OBRIEN                  37
        O/BRIEN         17                      (d = 1)
POLLY                   23
        PO11Y            8                      (d = 2)
        PO11Y            5                      (d = 1)
SCHWARTZ                15
MIKO                    15
        MIKO             3                      (d = 1)
PAULA                    8


...
================================================================
```

The formation of representatives can be supported with further knowledge depending on the application. Thus, a word can be mapped either onto a number or onto an alpha sequence by using OCR replacement tables which define interchangeable pairs of characters such as 1 - L, 0 - 0, 2 - Z, 6 - G etc. If, in addition, alternative sets for word classes to be learnt are known, for example nicknames for first names such as Paula - Polly, Thomas - Tom, etc., this replacement can also be performed. Both steps can be applied to the dictionary W1 51 which leads to a further blending of word classes.

Finally, all words occurring in the dictionary W1 51 are marked in the recognition results and supplemented by their representative. In the text which follows these words will be called W1 words.

At the top of the dictionary W1 51, the most frequent, previously unknown word forms are located and the word classes contain spelling variants thereof. Thus, in the application of in-house mail distribution, previously unknown first and second names and parts of departmental designations will be in the dictionary W1 51. In addition, their word classes contain spelling variants or variants which have arisen due to the characteristics of the reading system.

Finding Previously Unknown Word Groups Starting with the representatives of the word classes in the dictionary W1 51 which are marked as such in the recognition results, word groups of length 2 to n are determined in the next step according to figure 9 in that the neighborhoods of W1 words of the recognition results 62 are examined. For each W1 word, the right-hand neighborhood

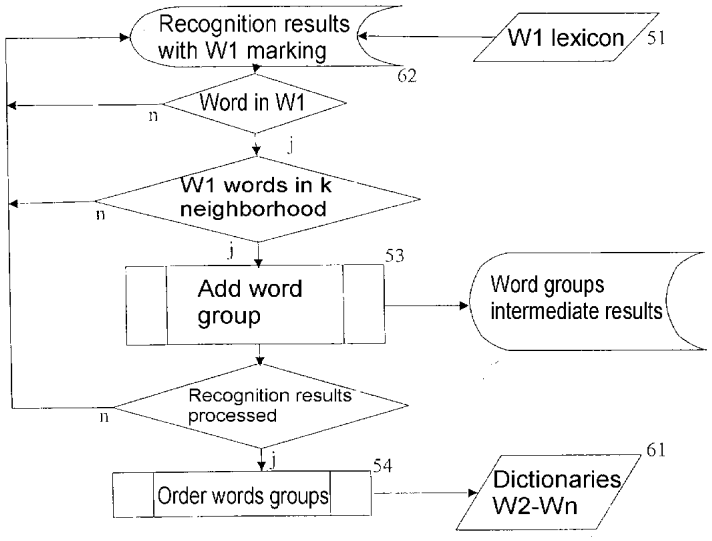

Fig. 9. flow structure for determining previously unknown word groups starting with the single words

is searched in a window of width k <= n to see whether it contains further W1 words. The so created groups of words are stored in lists we call dictionary. n-1 initially empty dictionaries are set up in a buffer and filled step by step. An n-tuple is then included in a word group buffer 53 when n W1 words have been found and there are fewer than m further non-W1 words between these n. As in the case of the dictionary W1 51, the frequency of occurrence of the individual word groups of length n is stored here, too.

The choice of the values of m and n depends on the actual application. For values of n >4, no further significant frequent entries can be expected in the application of reading addresses. m = 0 means that all n W1 words follow one another directly. In the case of pairs of first and last names, however, in particular, a second name can occasionally interrupt the direct succession, just as segmentation errors of the automatic reader can generate supposed word hypotheses and thus prevent a direct succession. In consequence, m = 1 and n = 3 are suitable values for the application described. In this step, in consequence, n 1 dictionaries Wn 61 containing frequent word sequences with the frequencies for pairs, triplets etc. up to n-tuple are generated from the word group buffer. In each dictionary Wn 61, the frequencies of the n-tuples are included with the frequencies of the W1 words of the n-tuples to calculate a dimension. Each dictionary Wn 61 is sorted in accordance with descending dimensions so that the most significant word groups are again at the beginning of each dictionary Wn 54.

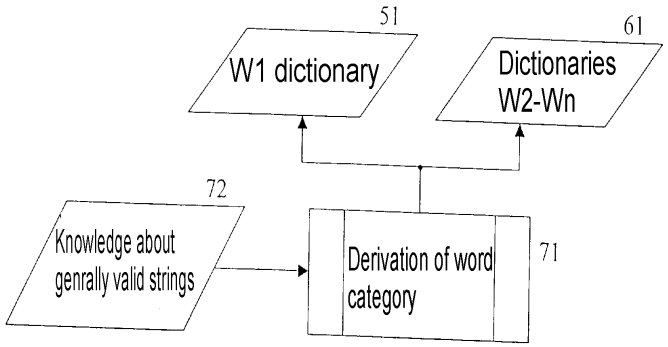

Fig. 10. flow structure for updating the dictionaries, taking into consideration the word categories

For the above example, the dictionary W2 looks as follows:

```
W2
=================================================================
COMMUNITY AFFAIRS              37
MANAGER COMMUNITY              37
POLLY OBRIEN                   23
MIKO SCHWARTZ                  15
PAUL OBRIEN                     8
=================================================================
```

The dictionary W3 has 3 entries provided that the name POLLY OBRIEN always occurs in combination with the designation MANAGER COMMUNITY AFFAIRS and that a line break is allowed in an n-tuple.:

```
W3
=================================================================
MANAGER COMMUNITY AFFAIRS      37
POLLY OBRIEN MANAGER           23
OBRIEN MANAGER COMMUNITY       23
=================================================================
```

Group Dictionary Update As described, the word proposals of the dictionaries Wn 61 (W2, W3, etc.) are now presented to an operator for validation

according to figure 10. Knowledge about the word units 72 to be learnt makes it possible at this point to categorize 71 entries in the dictionaries W1, W2, ... Wn 51, 61 semantically. Thus, in this application, entries can be allocated to the semantic class <Name>by looking at generally applicable lists of first names. This similarly applies to the semantic class <Department>which can be derived from keywords such as Department. Naturally, this process can also be carried out automatically without an operator by comparison with the entries of these lists.

Detecting obsolet words or word groups For addresses successfully correlated, the address elements required for this have been found in the dictionary and are identified as such in the recognition results. If, for example, last name and first name have been successfully read in the application of the in-house mail distribution, these results are registered in statistics; in particular, the frequency of the extracted words, pairs, generally of in-tuples over defined time intervals td, e.g. for a week, are stored and it is possible to take into consideration the type of item. As a result, a distribution of the address elements to be extracted for a sequence of time intervals is obtained:

```
================================================================
Time 1

MELINDA DUCKSWORTH       123
ALFRED SCHMID                          67
...

Time 2
MELINDA DUCKSWORTH         1
ALFRED SCHMID                          85
...

Time 3
MELINDA DUCKSWORTH         2
ALFRED SCHMID                          72
...
================================================================
```

From the distribution thus found, it is possible to derive whether dictionary entries are to be deleted: the entries are inserted into a list for removal from the dictionary if their frequency abruptly decreases from tdi to tdi+1 and stays at this level in successive time intervals tdi+k (e.g. k = 4). Thus, the person MELINDA DUCKSWORTH in the above example is deleted from the dictionary. This sequence can also be additionally conducted via a confirmation process.

5.3 Conclusion

The described system worked well for the in-house mail sorting application and national mail sorting as well. It was able to detect missing entries. Besides the correctly identified missing entries the system also indicated words or word groups as missing which in fakt have been of a wrong content. This happended for example for "postage paid". Next steps will be to refine the single processing steps in order to have it ready for daily use.

6 Summary and economical benefit

6.1 Technology advances

In this chapter it was shown, how the recognition methods were generalized and how the individual components of a postal address reader as well as the total system was made adaptive as a whole.

It was shown, how finding of all kinds of regions of interest were improved by statistical learning methods, namely by an adaptive histogram classifier. It was also shown, how stamps, logos, and other templates can be recognized and easily updated by use of learning algorithms.

It was also shown, how character recognition can be adapted to new fonts and writing styles in the target system by means of a new method, called Stepwise Improvement Classifier-, or SIC-method. This method enables the system to be self-adaptive, which is important for learning new fonts and handwriting idiosyncrasies at once.

Another important contribution to better adaptive-ness was described in detail: the HMM method with adaptive model structure, which proved to be a major break though in improvement of recognition of handwritten words. While in state-of the-art systems, only the parameters of models of fixed structure can be adapted, in the new method developed here, both the parameters, and the model structure itself will be modeled by the word learning set according to the complexity of the reading task. As a result, in an average over several country applications, an improvement of >6 % read rate was reached, which is significant.

Last but not least, the Learning Dictionary method developed under adaptive Read lead to the opportunity to find automatically obsolete data entries and new valid ones, which can be offered in two ways:

As an address data base improvement service and/or as a read improvement, since an updated address database is coupled with higher read performance.

6.2 Economic potential

Despite periodically recurring Cassandra calls, wanting to make us believe that physical mail will decrease or even vanish and will be finally substituted by electronical data transfer, especially by email, this seems fairly unlikely such as the paperless office since there is no way for substitution of mailing physical

goods. Therefore the next decade forecasts of the experts from UPU and other institutions dealing with postal future tell another story:

In average, physical mail will have an annual increase of 2 % in industrial countries, and even 4 % in developing countries such as China. This seems rather plausible when thinking of the recent development of e-purchasing and the heavy increase of flats and parcels caused by it.

So, mail in total will rather increase moderately, however, mail mix and postal logistics will change dramatically due to new market tendencies. This in turn provokes new and challenging requirements to the postal reading and sorting technology. Thanks to the advances in automatic pattern recognition and machine learning gained in the project *Adaptive READ*, Siemens Dematic is well armed for these future challenges, as there are:

- Omni –Mail format processing, including letters, flats, parcels, bundles, newspapers, or short: standard and non-standard mail.
- Omni – Image Object Finding, which means every information block, which is postally relevant like target address, senders address, stamps, labels, stickers, logos, endorsement lines for forwarding, etc., or short: every information block except advertisement.
- Omni – Country Application, which has been practised now for more than thirty countries, and will be potentially applicable for all UPU members.
- Omni – Alphabet Reading, adding to global Latin all the other local alphabets which are relevant in a global postal network as well, like Arabic, Cyrillic, Hangul, Chinese, Hindi etc.

Adaptive Read also brought an essential contribution impetus to new business through new functionalities such as Automatic Forwarding also called redirection of mail and revenue protection, and maybe in future also business reply mail.

Thanks to Adaptive Read project, also a set of new business ideas are realizable, such as web-based Reading/Coding services: The customer contracts with Siemens Dematic a service level agreement and uploads the images of his daily mail, and depending on his chosen service level he gets returned the read results of 100 % of the images or less within seconds or hours. Thus smaller private postal offices, which cannot afford large reading sorting equipment, can also benefit from this high sophisticated technology.

In the same way large postal services can benefit from an reject analysis and improvement service: A representative sub-sampled set of mail images is uploaded to the Reading & Coding web-center and the analysis results and improvement proposals are returned.

Another very promising service is the automatic address database clean up service, using address interpretation and learning dictionary methods to search out obsolete data entries as well as new valid aliases for improving efficiency of the reader system.

References

1. A. Kaltenmeier, T. Caesar, J. Gloger, and E. Mandler. Sophisticated topology of hidden Markov models for cursive script recognition. In *Proc. of the 2nd Int. Conf. on document analysis and recognition*, pages 139–142, Tsukuba Science City, Japan, Oct. 1993. IEEE Computer Society Press.
2. C. Bahlmann and H. Burkhardt. Measuring HMM similarity with the Bayes probability of error. In *Proc. of the 6th Int. Conf. on document analysis and recognition*, pages 406–411, Seattle, WA, Sept. 2001. IEEE Computer Society Press.
3. M.-P. Schambach. Model length adaptation of an HMM based cursive word recognition system. In *Proc. of the 7th Int. Conf. on Document Analysis and Recognition*, Edinburgh, Scotland, August 2003.
4. L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, Februar 1989.
5. M.-P. Schambach. Determination of the number of writing variants with an HMM based cursive word recognition system. In *Proc. of the 7th Int. Conf. on Document Analysis and Recognition*, Edinburgh, Scotland, August 2003.
6. P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. MIT Press, 1996.
7. M.-P. Schambach. *Automatische Modellierung gebundener Handschrift in einem HMM-basierten Erkennungssystem.* Proposed Dissertation, Universität Ulm, 2003.
8. Jürgen Schürmann. Pattern Classification A Unified View of Statistical and Neural Approaches, Chapter 6.14 *Recursive Learning* pages 165–182. ISBN 0471135348, Wiley-Interscience, March 1996.
9. K. Okuda, E. Tanaka and T. Kasai. A Method for the Correction of Garbled Words, based on the Levenshtein Metric IEEE Transactions on Computers, Vol. c-25, No. 2, February 1976