Final Report

October, 2004
VolkswagenStiftung Project Reference: I/77 863

OCoRrect: Cyrillic and Latin OCR Correction using Large Electronic Dictionaries and Sentence Context

Project leaders

Prof. Dr. Klaus U. Schulz
Centrum für Informations- und Sprachverarbeitung (CIS)
University of Munich

Dr. Stoyan Mihov
Central Laboratory for Parallel Processing
Bulgarian Academy of Sciences

Project Duration: March, 2002 — August, 2004

1 Summary of Scientific Results

In the project we covered a wide range of theoretical and practical issues of OCR postcorrection, both on the level of general alphabets and specifically for Cyrillic-Latin alphabets. A survey of the topics investigated in the project that supports orientation is given in the table below.

On the theory side, the most important achievement of the project are new and extremely fast methods for approximate search in large dictionaries. We developed a new correction methodology based on the concepts of Levenshtein automata and universal Levenshtein automata. Additionally we proposed the use of a combined – forwards and backwards – dictionary traversal, which provided significantly better efficiency compared to other methods. The results have been published in the leading journals of the field – [SM02,MS04].

Another theoretical achievement is a new method for dictionary rewriting using subsequential transducers. We developed an efficient method for constructing, given a rewrite dictionary a subsequential transducer that accepts a text as input and outputs the intended rewriting result under the so-called "leftmost longest match" replacement with skips. The main application in the project context are error-dictionaries that may be used to automatically correct typical errors in a very efficient way. The resulting rewriting mechanism is very efficient since it is linear in time in respect to the text size. This result is submitted to the JNLE [MS04a].

From the practical point of view the most important achievements are the preparation and evaluation of a representative OCR corpus, the large series of experiments with different correction strategies and the realization of a flexible and effective software system for OCR correction.

We created a Bulgarian OCR corpus (2304 documents) and a German OCR Corpus (349 documents). Both corpora follow a predefined structure and have only real life documents with a wide variety of styles, layouts, formatting, fonts, font sizes and printing quality in order to be representative. The analysis of the OCR errors in the corpora showed a new error class in mixed alphabet documents – the wrong replacement of letters in one alphabet with letters similar in shape in the other alphabet. This error class was not considered before in the literature and revealed to be very frequent in mixed Cyrillic-Latin texts. We reported the problem and a method for its correction in [MKRSS]. A further publication, to be submitted to a major international conference, is in preparation.

We made many large-scale experiments for correction with different correction dictionaries. These experiments showed how the use of domain specific and crawled "Web Dictionaries" significantly improve the correction results [SRSM03a,MKRSS04].

Another topic of interest was the combination of different ranking strategies based on word and collocation frequencies, weighted edit distance etc. for achieving better correction results. We implemented an optimization technique that automatically finds the optimal linear combination of the rankings and optimizes the resulting correction [SRSM03b].

And finally we developed a very flexible architecture for a software OCR postcorrection system. We constructed a pipeline where the data presented in an uniform XML format is processed by a pipe of specific tools. Initially the XML data is derived from the OCR-ed text and afterward on each step the data is enriched with additional elements like correction candidates and various rankings by each of the tools. At the end the data is evaluated and the corresponding correction result is given as output.

	General languages/alphabets	Latin Cyrillic
Theory	• Fast approximate search in large dictionaries [SM02, MS04] • Optimization of correction by rankings combination [SRSM03b] • Efficient text rewriting [MS04a]	• Corpus collection and analysis [Corpus] • Empirical study [Corpus]
Practice	• OCR Correction system [SRSM03a, SRSM03b, MKRSS04] • Combination of dictionaries [SRSM03b] • Combination of ranking methods [SRSM03a] • Sentence context with collocation frequencies	• Error analysis [MKRSS04] • Special correction techniques for mixed alphabets [MKRSS04]

2 Scientific Results

We present the main project achievements, closely following the description of the corresponding tasks given in the original project plan.

Further development of word context based correction method

In general the problem of word context correction can be considered as the searching and ranking of "similar correct" words in respect to the mistaken word. We can assume that the set of all correct words is given in a dictionary. Similar means within a given Levenshtein distance. What we need are methods for efficient filtering of correction candidates – extracting the set of similar words from a dictionary.

During the period we extensively analyzed and evaluated different methods for finding best correction candidates from a large electronic dictionary. The following main topics have been studied:

- Levenshtein automata with additional primitive edit operations suitable for OCR Correction.

  Those additional operations include symbol merges and symbol splits. We have shown how to extend the notion of the Levenshtein Automaton in order to reflect those additional primitive operations. We have presented the results in [SM01,SM02].

- Methods with optimal efficiency for finding correction candidates.

  We have extensively studied various methods for fast retrieval of the correction candidates. Those methods include dictionary segmentation with similarity keys, combined

straight and reverse dictionary traverse, lookup in dictionaries with garbled entries etc. We have presented the result of this study in [MS04].

- Universal Levenshtein Automata.

  We have introduced the notion of "Universal Levenshtein Automaton". The Universal Levenshtein Automaton of degree k may be used to decide for arbitrary words if their edit distance is less or equal to k. The transition labels of those automata are characteristic vectors in respect to a given pattern. This technique has additionally simplified the construction for finding correction candidates by improving the efficiency. We present the result in [MS04].

- Flensburg filter.

  We developed a new method for alignment of large parallel texts. This method is suitable for alignment of the OCR text with the "ground truth" data (ie.e, the correct original document) required for the analysis of the OCR errors and the correction evaluation. The method is based on so-called Flensburg filter and is reported in unpublished work.

From a practical point of view we need corresponding software tools, which implement the above listed methods. The Dictionary Application Tool presented in a later subsection implements the Levenshtein dictionary filtering method and the OCR alignment tool implements the Flensburg filter.

Theory for weighted Levenshtein automata. The initial plan to use weighted Levenshtein automata could not provide applicable results. The problem is that the determinization of a weighted Levenshtein automata led to an exponential blow up of the number of states. This made this method too expensive for finding the best correction candidates based on different weights of the primitive edit operations.

As a better alternative our team developed a new method for efficiently retrieving the best correction candidate by a very limited traversal of the dictionary automaton presented in [MS04]. This method combined with an additional ranking procedure delivered a very efficient method for deriving the candidates list.

Ordering correction candidates according their word frequencies. The method described in the previous paragraph has been supplemented with a procedure for assigning the word frequencies of the correction candidates. The final weight is given as a linear combination of the word frequency and the weighted edit distance. This method concludes our efforts for developing an universal, efficient and flexible technique for retrieving exclusively the best correction candidate on the base of primitive edition weights and word frequencies. The following section presents more details about the realization of the candidate ordering procedure.

Extension of the Bulgarian, Russian, German and English electronic dictionaries with OCR aiding data

Initially the idea presented in the project plan was to extend the dictionaries with OCR aiding data like word frequencies and other rankings. This additional OCR correction aiding data has to be derived by OCR error analysis on a OCR Corpus. We intended to use specific dictionary structures in order to provide the additional information. During the project our

team has made a number of experiments regarding construction of electronic dictionaries suitable for OCR postcorrection. The main results of this study are summarized below:

- Using of domain specific dictionaries

  We have shown that application of domain specific dictionaries for the OCR correction can significantly improve the correction quality. We have experimented by dynamical building of domain specific dictionaries using the Internet as texts source. This study is presented in [SRSM03a].

- OCR error analysis

  Based on the comparison of the original text with the OCR retrieved text the typical recognition errors have been categorized. Beside the useful categorization of the most common OCR error on the word level, this study revealed that a significant amount of OCR errors refer to the meta word level.

- Combining the result of distinct OCR systems.

  Our team has experiment with comparison of the OCR results retrieved from different OCR systems. We have shown that this technique can e.g. significantly reduce the "false friends" failure rate. The problem occurs when a wrongly recognized word appears in the dictionary. This study is presented in [SRSM03b].

As a consequence, and in contrast to the original project idea, we decided to present the correction aiding data *outside* the dictionaries. This makes the system much more flexible in respect to the addition of distinct dictionaries and the details of the combination of ranking scores used in the pipeline.

Representative corpora collection. The corpus collection revealed to demand huge efforts. In the framework of the project two large size OCR corpora have been created: Bulgarian OCR corpus (2304 documents) and German OCR Corpus (349 documents). Both corpora have followed the following methodology:

- Collecting real life documents from libraries, publishing houses, enterprises and organizations.
- Selection of documents using different languages and alphabets – Cyrillic, Latin and mixed alphabets in one document.
- Selecting wide variety of different styles, layouts, formatting, fonts, font sizes and printing quality.
- Obtaining permissions for the use of the provided documents for scientific purposes.
- Scanning with 600dpi optical resolution and 256 grades of gray – this way of scanning guarantees that the resulting image will be close enough to the original.
- Detailed description of each collected document.
- Providing the "ground true data" i.e. the original text for the documents.

The report [Corpus] gives a detailed overview about the collected corpora and a careful OCR error analysis. In addition we have enriched the description of each document in the corpus with the following information:

Font Family	All Errors	Latin-Cyrillic	Capital-Lower
Arial	3,8%	18,73%	6,74%
Courier	10,66%	13,03%	0%
Times	3,95%	2,66%	2,81%
Times Italic	12,5%	21,96%	13,63%
Universum	14,97%	32,72%	23,04%
TypeWriter	6,11%	3,39%	6,77%
Font Family	Letter-Digit	Punctuations	Other symbols
Arial	2,64%	0,60%	5,24%
Courier	6,36%	0%	0,66%
Times	1,94%	1,55%	5,56%
Times Italic	0%	1,13%	1,13%
Universum	17,01%	0,71%	5,31%
TypeWriter	1,01%	2,30%	3,07%

Table 1: Statistics of OCR errors on the Bulgarian Corpus.

- Font characteristics
- Percentage of words with OCR errors
- Classification of each of the OCR errors into one of the following categories:
  - split of letters
  - merge of letters
  - substitution of letters
  - deletion of space (merge of words)
  - insertion of space (split of words)
  - lower case – upper case substitution
  - letter with digit substitution
  - letter with special symbol substitution
  - Cyrillic with Latin letter substitution
  - punctuation substitution

OCR error analysis. Table 1 presents the analysis of the different types of errors in respect to the font type and percentage of OCR errors.

Based on the comparison of the original text with the OCR retrieved text the typical recognition errors have been categorized. Beside the useful categorization of the most common OCR error on the word level, this study revealed that a significant amount of OCR errors refer to the meta word level.

In our experiments we found a new source of OCR errors for documents which contain *both* Latin *and* Cyrillic letters. The problem is that there are a number of Latin-Cyrillic letter pairs which have the very same graphical representation. For example the Latin letter *p* has the same graphical representation as the Cyrillic letter *r*. There are approximately 12 such pairs depending on the font. In many cases whole Cyrillic words are recognized as Latin strings. For example the Bulgarian word *paca* is often mistaken as a Latin string.

Although graphically the recognized Latin string looks exactly the same like the Cyrillic word, the resulting encoding is entirely different. Because of that we have problems applying electronic processing tasks like indexing, searching, etc. on the OCR documents. As a result we had a significantly higher error rate for the OCR on documents with mixed Latin-Cyrillic alphabets.

As clearly shown the largest class of problems arises because of the Cyrillic with Latin letter substitution especially in the "Universum" font type. This font is very commonly in magazines, newspapers and some books in Cyrillic. We have paid special attention in order to be able to correct this error type. For handling the Latin-Cyrillic similarity problem we extended our correction method by treating those symbol pairs as equivalent symbols. This means that words like *paca* are considered as written in Cyrillic as well as in Latin. We look up the Cyrillic encoding in the Bulgarian dictionary and the corresponding Latin encoding in the English dictionary.

Word frequencies analysis. We have analyzed 33 million words Bulgarian and English Corpus for retrieving the relative word frequencies of the dictionary words. To realize this we implemented a very efficient tool presented in the next section. For German we used a 1.5 terabyte Web corpus to get the word frequencies.

Analysis of recognition error risk. Our studies revealed that the recognition error risk of a given word exclusively depends on the symbol sequence in the word. In that sense the recognition error risk for a given word can be retrieved using the possibilities of the primitive edit operations for the symbols of the word which we calculate from a corpus with the OCR error processing tool.

Construction of Bulgarian, Russian, German and English OCR dictionaries and Bulgarian-Russian-German-English consolidated OCR Dictionary. As mentioned above initially we intended to build specific dictionaries extended with OCR correction aiding data. In order to have a very flexible architecture we designed our system to use any number of dictionaries in the OCR correction pipeline implemented with the Dictionary Application tool. Afterward we are able to add any number of additional data like weighted similarity, frequencies, collocation etc. by applying the Candidate Ranking Tools.

Test series for the probabilities of symbol dependent recognition errors

Collection of font samples of representative documents and analysis of the font samples for symbol dependent recognition errors. As described in the previous subsection we collected and scanned representative documents from very different sources printed with different fonts. Based on this data the OCR error processing tool produces the statistics of the individual OCR errors appearing in the text. This is realized by the OCR error processing tool which takes as input aligned OCR-ed and original texts and produces the symbol confusion matrix.

Sentence context based OCR correction

Statistical analysis of representative corpora for word collocation. Our method for ranking the correction candidates provides the ability to combine the collocation frequency information in the ranking procedure. In order to realize this we implemented the

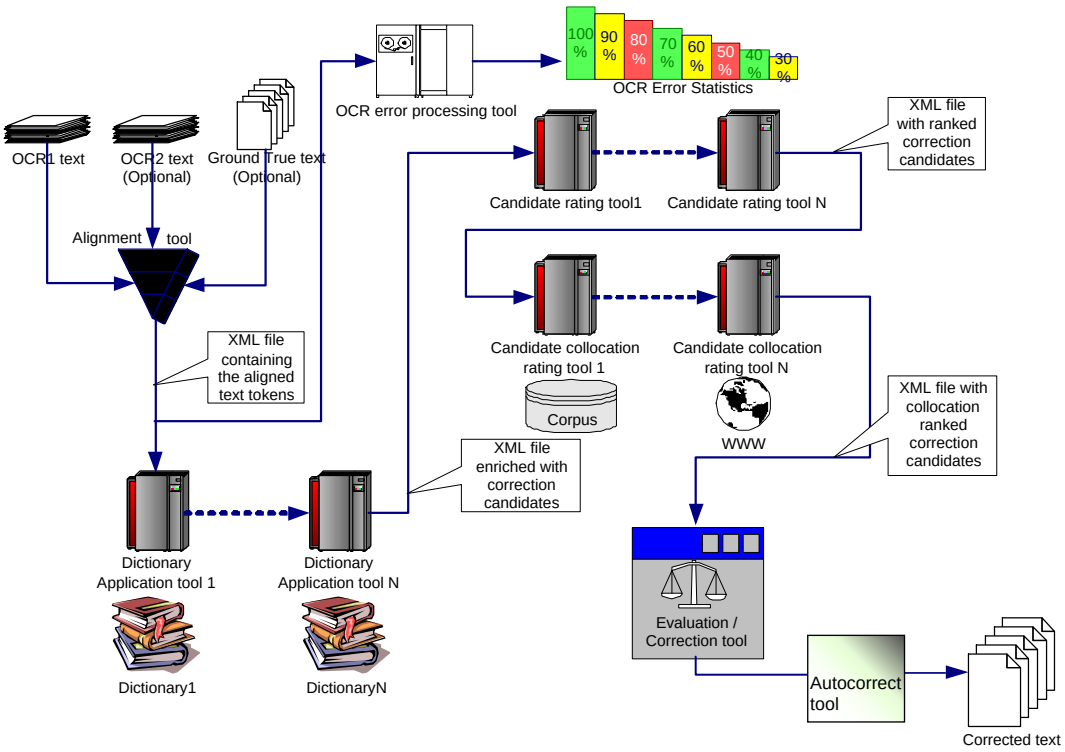

Figure 1: The OCR correction pipeline.

Candidate Collocation Rating tool described in the next section. The tool derives the frequencies of the word trigram consisting of the correction candidate in the middle and the left and right neighboring word. The tool can be applied using either a given large scale text corpus or the Web for frequency evaluation.

2.1 Comparison of the experimental results with industrial systems

In [SRSM03a,SRSM03B,MKRSS04] we have given the results of our correction after applying it to two leading industrial OCR systems. The result showed in all tests improvements in the recognition precision. For some of the tests the improvements were dramatic.

2.2 Implementation

The team has developed a very flexible architecture for OCR Correction system. The main idea is to construct a correction pipeline, where the data presented in an uniform XML format is processed by a pipe of specific tools. Initially the XML data is derived from the OCR-ed text and afterward on each step the data is enriched with additional elements by each of the tools. At the end the data is evaluated and the corresponding correction result is given as output. The diagram below presents the scheme of the OCR correction pipeline.

For the purposes of the OCR Correction pipeline we developed a special XML format. Below the XML DTD is given:

```
<!ELEMENT    SCF              (record+)>
```

```
<!ELEMENT   record          (cands)>
<!ATTLIST   record
  wOrig     CDATA           #REQUIRED
  wOCR      CDATA           #REQUIRED
  inLex     (true|false)    #REQUIRED
  addr      NMTOKEN         #REQUIRED>
<!ELEMENT   cands           (cand*)>
<!ELEMENT   cand            EMPTY>
<!ATTLIST   cand
  vCand     CDATA           #REQUIRED
  score0    NMTOKEN         #REQUIRED
  score1    NMTOKEN         #REQUIRED>
```

In the DTD `wOrig` corresponds to the original token and `wOCR` is the OCR recognized token. `inLex` is a flag showing whether `wOrig` is contained in the dictionary. `addr` is the start position of `wOCR` in the text file provided by the OCR-engine. For every `wOrig`-token there is a set of correction candidates. The set might also be empty. A correction candidate consists of scores. A score is always in the range [0; 1] where 1 is the "good" end of the scale and 0 the "bad" one. A (confusion) probability perfectly corresponds with a score. Scores can be derived from different things like edit distance or weighted Levenshtein distance or frequency information. Generally scores can be combined. A combined score is used to rank different candidates and also used to rank the `wOrig`-tokens (according to the confidence of the post correction).

OCR alignment tool

The first process in the pipeline is done by the token Alignment Tool. It takes as input the text from the OCR engine(s) and the original text in parallel. The tool aligns the texts token by token and produces the first initial XML file containing only the aligned tokens. Initially we encountered many difficulties for this initial task because of the many different types of errors which could appear in a OCR text. We tried the standard methods using the dynamic programming framework first. Unfortunately they could hardly be applied to large texts because of the quadratical time complexity of those algorithms. We could not limit the number of errors in the documents because they depend on the size.

In order to create a robust alignment tool which works sufficiently fast on large documents we developed a new method based on Flensburg filters as mentioned in the previous section. The idea is that we give penalty points for each error but for each correct alignment a part of the penalty is remitted.

The tool is programmed in Java and is Unicode compliant. It has been extended to use a table of graphically equivalent letters for solving the Latin-Cyrillic similarity problem described in the previous section.

After applying the tool the resulting XML file looks like:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
 <record wOrig="(a)" wOCR="(a)" addr="0" />
 <record wOrig="Средна" wOCR="Средна" addr="4" />
 <record wOrig="геометрична" wOCR="геометрична" addr="11" />
 <record wOrig="стойност" wOCR="стойност" addr="23" />
 <record wOrig="за" wOCR="за" addr="32" />
 <record wOrig="период" wOCR="период" addr="35" />
 <record wOrig="от" wOCR="от" addr="42" />
 <record wOrig="два" wOCR="дВа" addr="45" />
 <record wOrig="месеца" wOCR="месеца" addr="49" />
 <record wOrig="при" wOCR="при" addr="56" />
 <record wOrig="взимането" wOCR="Взимането" addr="60" />
 <record wOrig="най-малко" wOCR="най-мал!<o" addr="70" />
 <record wOrig="на" wOCR="на" addr="81" />
 <record wOrig="две" wOCR="дбе" addr="84" />
 <record wOrig="проби" wOCR="проби" addr="88" />
...
```

OCR error processing tool

This tool gets as input the XML file with the aligned tokens and produces as a result detailed statistics about all individual OCR errors which occurred in the file. The idea is that during the training phase this tool will produce the statistics which used afterward for calculating the individual OCR error probabilities.

This tool is programmed in C++. It takes as parameters the alphabet, the table of graphically equivalent letters, the XML file with the aligned tokens. The result is a table of containing the counts of all primitive OCR errors and statistical data about the text.

Dictionary application tool

This is one of the central tools in our pipeline. It takes as input the XML file with the aligned tokens and produces in its output the XML file containing a list of correction candidates for each of the OCR tokens. This list is retrieved from a large scale dictionary using the technique presented in [MS04]. Since there are tens of candidates for each token we have to be able to produce this lists as fast as possible. With the new method this step requires a couple millisecond per token.

The tool takes as input the dictionary, its alphabet, the table of graphically equivalent letters and the XML file with the tokens. The result is the XML file enriched with lists of correction candidates. The tool is programmed in C++ for providing maximal efficiency. Below an extract from the resulting XML file after the application of the tool is given:

```
...
<record addr="84" wOCR="дбе" wOrig="две" inLex="false">
 <cands>
  <cand vCand="абе" />
  <cand vCand="бе" />
  <cand vCand="две" />
  <cand vCand="де" />
  <cand vCand="дуе" />
  <cand vCand="обе" />
 </cands>
</record>
<record addr="88" wOCR="проби" wOrig="проби" inLex="true">
 <cands>
  <cand vCand="дроби" />
  <cand vCand="поби" />
  <cand vCand="пороби" />
  <cand vCand="преби" />
  <cand vCand="проба" />
  <cand vCand="проби" />
  <cand vCand="пробив" />
  <cand vCand="пробие" />

 </cands>
</record>
...
```

Candidate ranking tools

Those are the tools responsible for the ranking of the correction candidates based on various criteria. Each of the tools gets as input the XML file with the tokens and correction candidates. The tool is assigning a given score to each of the correction candidates on the basis of a given criteria. We have developed candidate ranking tools for the following criteria:

- inverted normalized Levenshtein distance;
- normalized word frequency;
- OCR error probability (based on the statistics provided by the OCR error processing tool).
- Normalized collocation frequency (based on the frequency of occurring the given candidate in the context of the neighboring tokens derived either from a Web search engine or from a indexed text corpus);
- OCR engines combination score.

All the candidate ranking tools are implemented in Java. They take as arguments the XML file with the tokens and their candidates and depending on the criteria either a frequency dictionary, OCR error statistics or collocation indexes. The result is the XML file

enriched with scoring information for each of the candidates. Below a fragment of such a XML result file is presented:

```
...
<record addr="84" wOCR="дбе" wOrig="две" inLex="false">
 <cands>
  <cand vCand="абе" score0="3.4700000000000002E-6" score1="0.8333333333333334" />
  <cand vCand="бе" score0="0.0010039408" score1="0.8" />
  <cand vCand="две" score0="2.098604E-4" score1="0.8333333333333334" />
  <cand vCand="де" score0="4.33228E-5" score1="0.8" />
  <cand vCand="дуе" score0="2.1879999999999997E-7" score1="0.8333333333333334" />
  <cand vCand="обе" score0="2.02E-7" score1="0.8333333333333334" />
 </cands>
</record>
<record addr="88" wOCR="проби" wOrig="проби" inLex="true">
 <cands>
  <cand vCand="дроби" score0="1.348E-7" score1="0.9" />
  <cand vCand="поби" score0="6.719999999999999E-8" score1="0.8888888888888888" />
  <cand vCand="пороби" score0="1.348E-7" score1="0.9090909090909091" />
  <cand vCand="преби" score0="1.2632E-6" score1="0.9" />
  <cand vCand="проба" score0="1.4485999999999998E-5" score1="0.9" />
  <cand vCand="проби" score0="1.71136E-5" score1="1.0" />
  <cand vCand="пробив" score0="2.9308E-6" score1="0.9090909090909091" />
  <cand vCand="пробие" score0="2.4592E-6" score1="0.9090909090909091" />
 </cands>
</record>
...
```

Evaluation / correction tool

This is the final tool in our pipeline. It takes as input the initial OCR text file and the XML file with the aligned tokens, lists of correction candidates and their scorings. There are two modes of operation. In the first (training) mode the XML file has to contain the OCR tokens and the corresponding original text tokens. In this phase the tool is optimizing the weights of each scoring and the threshold for correction in order to achieve maximal preciseness. In the second phase, using the parameters derived by the training, the tool is producing a corrected text where OCR errors are corrected. When the original text is available, the tool can be used for evaluating the results.

The correction tool is programmed in Java and has a graphical user interface. A screen shot of the tool is given on Figure 2.

Autocorrection tool

As seen on Table 1 a number of OCR errors are in punctuations (very often a comma is recognized as a dot or vice versa), in letter / digit or OCR errors on other non-letter symbols. Such kind of errors can not be corrected using the dictionary based approach. For couping with some of those errors we used another strategy as a final step in the OCR correction pipeline

Modern text processing packages provide the option for automatic correction. The idea is that the strings which are often mistaken and can not be wrongly interpreted are corrected automatically. For example some of the word-processors are correcting *teh* to *the*

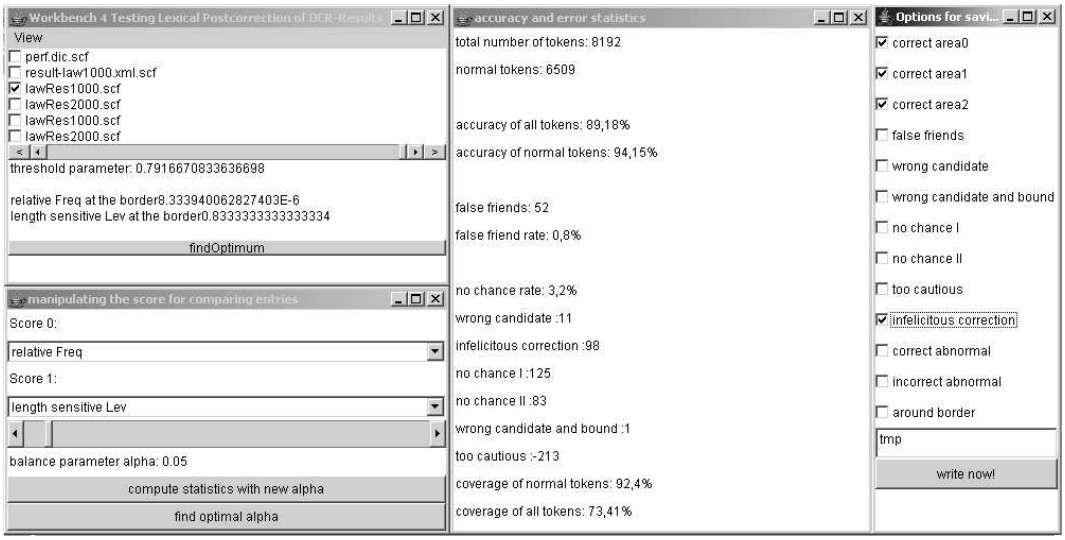

Figure 2: The evaluation / correction tool.

automatically during typing. The correction is done by simply rewriting each string from a predefined dictionary with its corresponding correction.

For example each occurrence of the string '`. which`' could to be automatically replaced by the string '`, which`'. Many other common OCR mistakes outside the words can be corrected by applying some fixed set of rewriting rules presented as a rewrite dictionary. The rewrite dictionary is a finite list of pairs of strings, representing a source string (the "original") and its substitute. In our case the tool processes the whole text as one string of characters (without any tokenization) and rewrites it by replacing all occurrences of originals by their substitutes.

We have developed a new, very efficient method for implementing rewrite dictionaries. The result is reported in [MS04a]. In our approach we construct a directly a subsequential transducer for representing the rewrite dictionary. Let us consider the rewrite dictionary

$$(a \mapsto 1), (ab \mapsto 2), (abcc \mapsto 3), (babc \mapsto 4), (c \mapsto 5).$$

Then the corresponding subsequential transducer is given on Figure 3. The new algorithm does not use backward steps and no memory device is needed: in our approach we compute, given the rewrite dictionary D, a sequential finite state transducer $\mathcal{T}$ that always produces the desired transformation result when applied to an input text t.

Our autocorrect tool is implemented using this technology. The tool is programmed in C and rewrites the text with a speed of 1,2 MB/s (disk I/O operations included).

3 Publications and Presentations

3.1 Publications

Here we present the publications in respect to the project with their abstracts. The three most important publications are [MS04], [SM02], [SRSM03a].

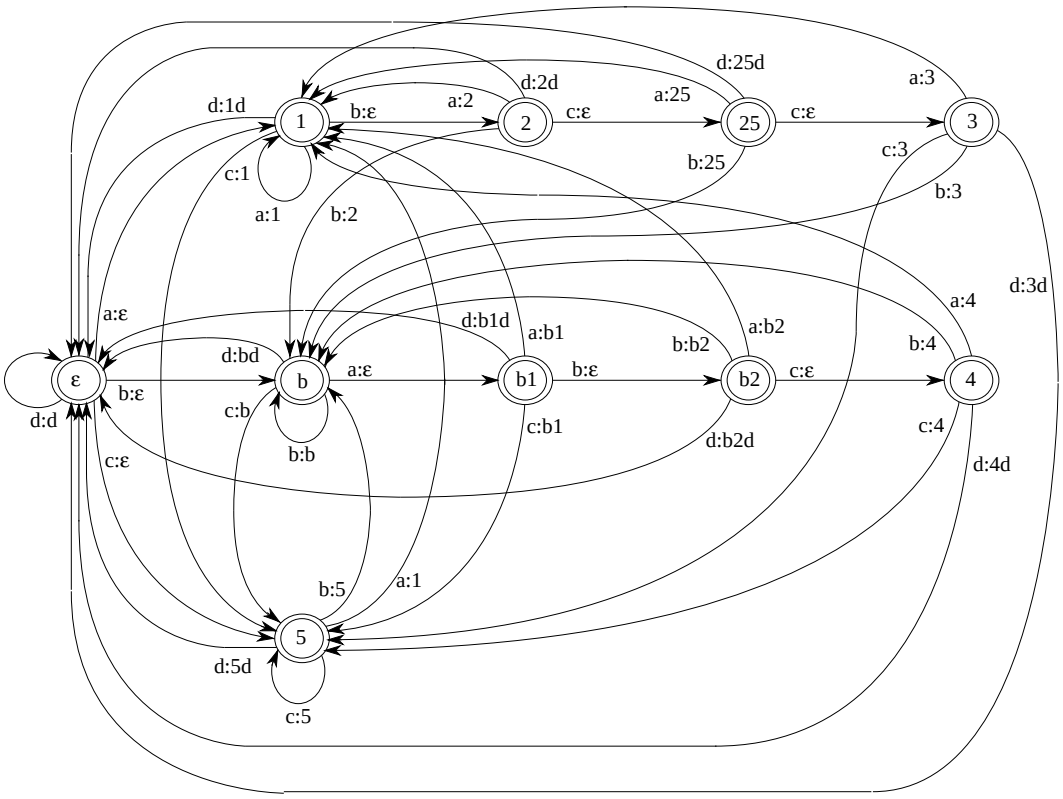

Figure 3: The sequential transducer for rewriting dictionary.

[SM02] Klaus U. Schulz, Stoyan Mihov, *Fast string correction with Levenshtein automata*, IJDAR 5 (2002) 1, 67-85

The Levenshtein-distance between two words is the minimal number of insertions, deletions or substitutions that are needed to transform one word into the other. Levenshtein-automata of degree n for a word W are defined as finite state automata that recognize the set of all words V where the Levenshtein-distance between V and W does not exceed n. We show how to compute, for any fixed bound n and any input word W, a deterministic Levenshtein-automaton of degree n for W in time linear in the length of W. Given an electronic dictionary that is implemented in the form of a trie or a finite state automaton, the Levenshtein-automaton for W can be used to control search in the lexicon in such a way that exactly the lexical words V are generated where the Levenshtein-distance between V and W does not exceed the given bound. This leads to a very fast method for correcting corrupted input words of unrestricted text using large electronic dictionaries. We then introduce a second method that avoids the explicit computation of Levenshtein-automata and leads to even improved efficiency. Evaluation results are given that address both variants.

[SM01] Klaus U. Schulz and S. Mihov, *Fast String Correction with Levenshtein-Automata*. CIS-Bericht-01-127, Centrum fur Informations- und Sprachverarbeitung, Universitat Munchen, 2001.

This paper (67 Pages) is an extension of the previously mentioned paper where transpositions, as well as merges and splits, are treated as additional edit operations. All technical details concerning design and computation of appropriate Levenshtein-automata are explained, evaluation results are given.

[SRSM03a] Christian Strohmaier, Christoph Ringlstetter, Klaus U. Schulz, Stoyan Mihov, *Lexical Postcorrection of OCRResults: The Web as a Dynamic Secondary Dictionary?* Proceedings of ICDAR 2003.

Postcorrection of OCRresults for text documents is usually based on electronic dictionaries. When scanning texts from a specific thematic area, conventional dictionaries often miss a considerable number of tokens. Furthermore, if word frequencies are stored with the entries, these frequencies will not properly reflect the frequencies found in the given thematic area. Correction adequacy suffers from these two shortcomings. We report on a series of experiments where we compare (1) the use of fixed, static large scale dictionaries (including proper names and abbreviations) with (2) the use of dynamic dictionaries retrieved via an automated analysis of the vocabulary of web pages from a given domain, and (3) the use of mixed dictionaries. Our experiments, which address English and German document collections from a variety of fields, show that dynamic dictionaries of the above mentioned form can improve the coverage for the given thematic area in a significant way and help to improve the quality of lexical postcorrection methods.

[SRSM03b] Christian Strohmaier, Christoph Ringlstetter, Klaus U. Schulz, Stoyan Mihov, *A visual and interactive tool for optimizing lexical postcorrection of OCR results*, Proceedings of "DIAR-03: Workshop on Document Image Analysis and Retrieval" (In conjunction with IEEE CVPR'03) Madison, Wisconsin, June 21, 2003.)

Systems for postcorrection of OCRresults can be fine tuned and adapted to new recognition tasks in many respects. One issue is the selection and adaption of a suitable background dictionary. Another issue is the choice of a correction model, which includes, among other decisions, the selection of an appropriate distance measure for

strings and the choice of a scoring function for ranking distinct correction alternatives. When combining the results obtained from distinct OCR engines, further parameters have to be fixed. Due to all these degrees of freedom, adaption and fine tuning of systems for lexical postcorrection is a difficult process. Here we describe a visual and interactive tool that semiautomates the generation of ground truth data, partially automates adjustment of parameters, yields active support for error analysis and thus helps to find correction strategies that lead to high accuracy with realistic effort.

[MS04] Stoyan Mihov, Klaus U. Schulz *Fast approximate search in large dictionaries*, Journal of Computational Linguistics Vol. 30(4), December 2004. (In print)

The need to correct garbled strings arises in many areas of natural language processing. If a dictionary is available that covers all possible input tokens, a natural set of candidates for correcting an erroneous input P is the set of all words in the dictionary for which the Levenshtein distance to P does not exceed a given (small) bound k. In this paper we describe methods for efficiently selecting such candidate sets. After introducing as a starting point a basic correction method based on the concept of a "universal Levenshtein automaton", we show how two filtering methods known from the field of approximate text search can be used to improve the basic procedure in a significant way. The first method, which uses standard dictionaries plus dictionaries with reversed words, leads to very short correction times for most classes of input strings. Our evaluation results demonstrate that correction times for fixed distance bounds depend on the expected number of correction candidates, which decreases for longer input words. Similarly the choice of an optimal filtering method depends on the length of the input words.

[MKRSS04] Stoyan Mihov, Svetla Koeva, Christoph Ringlstetter, Klaus U. Schulz and Christian Strohmaier *Precise and Efficient Text Correction using Levenshtein Automata, dynamic WEB Dictionaries and optimal correction*, Proceedings of the Workshop on International Proofing Tools and Language Technologies, Patras, Greece, 2004.

Despite of the high quality of commercial tools for optical character recognition (OCR) the number of OCR-errors in scanned documents remains intolerable for many applications. We describe an approach to lexical postcorrection of OCR-results developed in our groups at the universities of Munich and Sofia in the framework of two research projects. Some characteristic features are the following:
(1) On the dictionary side, very large dictionaries for languages such as German, Bulgarian, English, Russian etc. are enriched with special dictionaries for proper names, geographic names and acronyms. For postcorrection of texts in a specific thematic area we also compute "dynamic" dictionaries via analysis of web pages that fit the given thematic area.
(2) Given a joint background dictionary for postcorrection, we have developed very fast methods for selecting a suitable set of correction candidates for a garbled word of the OCR output text.
(3) In a second step, correction candidates are ranked. Our ranking mechanism is based on a number of parameters that determine the influence of features of correction suggestions such as word frequency, edit-distance and others. A complex tool has been developed for optimizing these parameters on the basis of ground truth data.
Our evaluation results cover a variety of corpora and show that postcorrection improves the quality even for scanned texts with a very small number of OCR-errors.

[MS04a] Stoyan Mihov, Klaus U. Schulz, *Efficient Dictionary-Based Text Rewriting using Sequential Transducers*, Submitted to Journal of Natural Language Engineering.

Problems in the area of text and document processing can often be described as *text rewriting tasks*: given an input text, produce a new text by applying some fixed set of rewriting rules. In its simplest form, a rewriting rule is given by a pair of strings, representing a source string (the "original") and its substitute. By a rewrite dictionary, we mean a finite list of such pairs; dictionary-based text rewriting means to replace in an input text occurrences of originals by their substitutes. We present an efficient method for constructing, given a rewrite dictionary D, a subsequential transducer $\mathcal{T}$ that accepts a text t as input and outputs the intended rewriting result under the so-called "leftmost longest match" replacement with skips, t'. The time needed to compute the transducer is linear in the size of the input dictionary. Given the transducer, any text t of length $|t|$ is rewritten in time $O(|t| + |t'|)$, where t' denotes the resulting output text. Hence the resulting rewriting mechanism is very efficient. As a second advantage, using standard tools, the transducer can be directly composed with other transducers to efficiently solve more complex rewriting tasks in a single processing step.

[Corpus] Klaus Schulz, Christoph Ringlstetter, Claudia Gehrcke, Annette Gotschareck, Stoyan Mihov, Veselka Dojchinova, Vanja Nakova, Kristina Kalpakchieva, Ognjan Gerasimov, *The SOFIA-MUNICH-Corpus: A repository of scanned documents for evaluating OCR-software and techniques for postcorrection of OCR-results*. We intend to submit a revised version of the paper to an international conference (ICDAR 2005).

The evaluation of OCR-software and techniques for postcorrection of OCR-results is difficult since only a small number of corpora are freely available that are appropriate for this task. In general, these corpora do not cover special languages and alphabets. In this paper we describe the Sofia-Munich-Corpus of scanned paper documents designed for the above-mentioned tasks at Sofia University/Bulgarian Academy of Science and at Munich University. The corpus was developed in the framework of a two-years project funded by VolkswagenStiftung. Since the project had a special focus on problems for OCR caused by a mixed Cyrilic-Latin alphabet, the major part of the corpus (2306 files) consists of Bulgarian documents. A smaller German subcorpus has been added. We describe the structure of the corpus and add technical details about file formats and other kind of useful meta-information. We also summarize the typical problems and errors that were observed when applying modern industrial OCR technology to convert the documents to electronic textual form. As a special feature that drastically simplifies all kinds of evaluations, perfect reconstructions of the original texts (ground truth data) have been prepared for many documents. We plan to make the SOFIA-MUNICH-Corpus freely available, together with an updated version of the present paper.

3.2 Thesis

In Sofia 2 Master's Thesis have been successfully completed and one more is in preparation.

- Vanja Nakova: *Rule based OCR Error Correction*, Sofia University, Faculty of Slavic Studies.
- Kristina Kalpakchieva: *Rule based grammar correction in OCR Documents*, Sofia University, Faculty of Slavic Studies.
- Ivan Pejkov: *Direct Construction of a Bimachine for Rewriting Rule*, Sofia University, Faculty of Mathematics and Informatics. (In Preparation)

In Munich, two Master's Thesis at Ludwig-Maximilians-University (LMU), a Diploma Thesis at Technical University (TUM), and one Ph.D. thesis at LMU have been successfully completed.

- Christoph Ringlstetter: *OCR-Korrektur und Bestimmung von Levenshtein-Gewichten*, Master's Thesis LMU,
- Yulia Palchaninava: *Kontextuelle Verfahren bei der OCR-Nachkorrektur*, Master's Thesis, LMU,
- Florian Schwarz: *Development and Evaluation of Different Algorithms for Inexact Matching.* Diploma Thesis, TUM,
- Christian M. Strohmaier: *Methoden der lexikalischen Nachkorrektur OCR-erfasster Dokumente.*

3.3 Conference and workshop presentations

- Presentation of [SRSM03a] at the Seventh International conference on Document Analysis and Recognition – Edinburgh, August 2003.
- Presentation of [SRSM03b] at the Workshop on Document Image Analysis and Retrieval (DIAR-03) – Madison, Wisconsin, June 2003.
- Presentation of [MKRSS04] at the Workshop on International Proofing Tools and Language Technology – Patras, June 2004.

3.4 Project web site

A Web site dedicated to the OCoRrect project was created at `http://lml.bas.bg/ocorrect`. The web site contains all major information about the project:

- Project Goals and Objectives;
- Project team members;
- Project events;
- Project results;
- Contacts.

The main purpose of the site is to present the project achievements to the wide public.

3.5 Internal meetings

- The first OCoRrect project meeting has been held in Sofia on 11-13 September 2002.
- The second OCoRrect project meeting has been held in Munich on 3-5 November 2002.
- The third OCoRrect project meeting has been held in Munich on 16-20 July 2003.
- The fourth OCoRrect project meeting has been held in Sofia on 8-12 October 2003.
- The fifth OCoRrect project meeting has been held in Munich on 18-22 February 2004.
- The sixth OCoRrect project meeting has been held in Sofia on 4-8 August 2004.

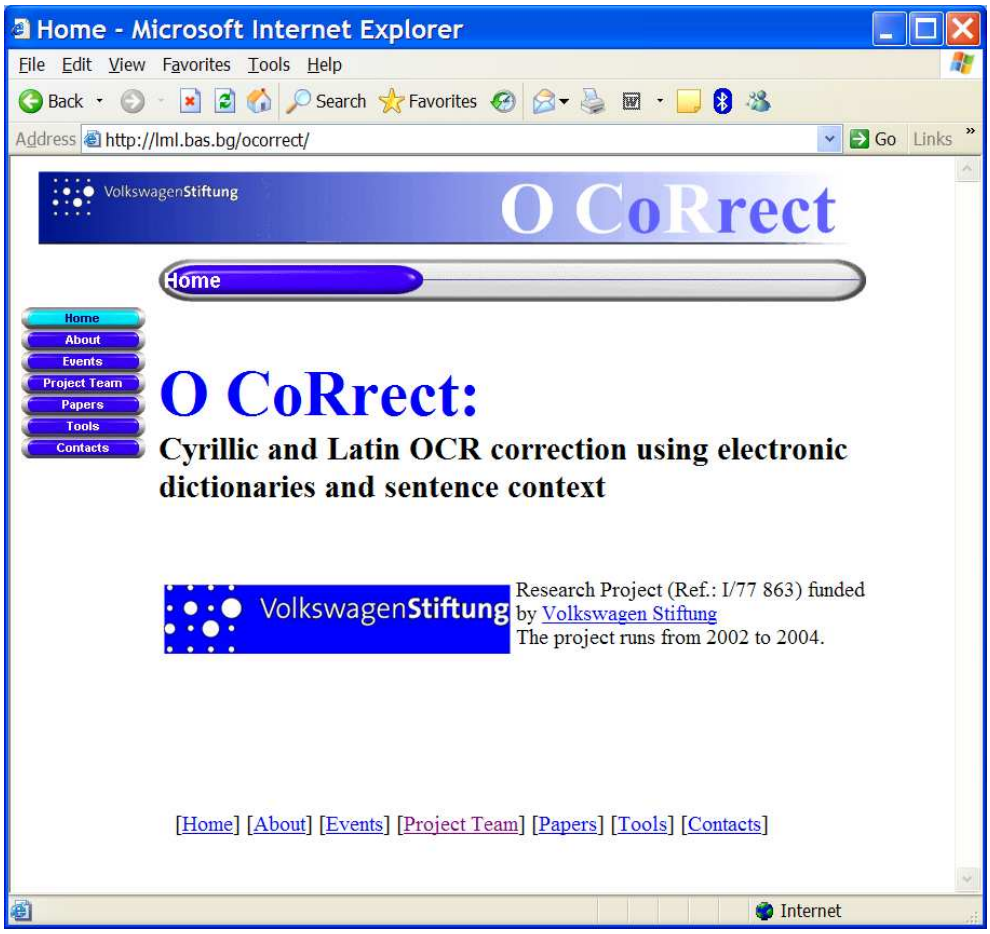

Figure 4: The Project web site.

4 Cooperation

Most of the work in the project was mainly done by our two teams. Some of the results have been applied commercially by the AREXERA Gmbh. We are in close contact to a small Suisse enterprise that is interested in using the techniques for correcting OCR-results developed in our groups.

5 Own Evaluation of Results

We have got our theoretical results accepted in two leading journals, and a new journal publication has been submitted. Three papers have been presented at the leading international conferences and workshops.

The method for fast approximate dictionary lookup developed in the project framework provides the best known efficiency – in order of magnitude better than the other methods. Our text rewriting method with a rewriting dictionary provides optimal assymptotic complexity as well.

On the practical side we have created a large representative OCR corpus for Bulgarian and German OCR texts. A very flexible and efficient implementation of the OCR correction pipeline has been realized. We enriched the OCR correction system with the sentence context evaluation tool and a tool for combining the results of two OCR engines.

We see two important points to be considered in future work. First, the influence of the sentence context ranking to the correction result should be studied in more depth. It is desirable to complete a series of experiments in order to clarify the dependence on the domain and language used and how much improvements it yields. The second point is the further exploration of correction based on the output of two different OCR engines. This idea was not considered in the beginning of the project, but our experiments showed that this approach can handle successfully one of the worst OCR problems – the false friends. Further development and evaluation of this technique would be very valuable for the whole OCR field.

Based on the above we consider the work on the project as very successful and satisfactory. We would like to thank VolkswagenStiftung again for the kind support.