handwriting OCR, medical documents
language model, tagger
parser, Polish

Maciej PIASECKI*

MULTILEVEL CORRECTION OF OCR OF MEDICAL TEXTS

In the paper the idea of the multilevel correction of the results handwriting OCR of medical texts is investigated. The correction is performed according to different levels of linguistic knowledge. Three types of models, namely: the n-gram Language Models of word form and base form sequences, the morpho-syntactic model based on a tagger and the model of correction by parsing are presented and their results are compared. The parsing model is based on the combination of a deterministic Czech parser adapted for Polish and the Structured Language Model based on lexicalised, binary parsing trees produced in the left-to-right manner. Contrary to the initial expectations, the best result of correction from 82% of the word level classifier to 92.98% of the overall accuracy was achieved with the help of a n-gram Language Models. The more rich description of language expressions in a model, the worse results were obtained. This result is in large extent caused by the specific characteristics of the processed medical documents.

1. INTRODUCTION

Contemporary medical documents are mostly created in electronic format, but still many handwritten documents are stored in archives. They can comprise a very valuable source of statistical knowledge as only they are converted to electronic forms. Unfortunately, medical handwritten documents are known from the low quality of their writing style. However, they are written according to some stable schemes, come from a limited domain and their author and the place of the origin are very often known. These factors make their structure and content predictable in some extent. OCR is typically divided into two phases: recognition of words by a *word classifier* working on the level of letter and word images, and recognition of text — sequences of words. Word classifiers utilizes stochastic models of letter sequences and dictionaries of known word forms, but only as the prediction of sequences of words is added, the quality of the whole process can reach the practical level.

The prediction of a word form on the basis of preceding ones (or surrounding) is typically performed with the help of a *Language Model* (LM). LM is a stochastic model describing probabilities of word form sequences. However, in the case of Polish, the number of different possible word forms (about 1.7 million) and the free word order seem to make collecting the amount of data needed for the proper probability estimation impossible in general. On the level of morphology, Polish word forms encode morpho-syntactic properties

* Institute of Applied Informatics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, Wrocław, Poland, maciej.piasecki@pwr.wroc.pl

that constraint possible syntactic structures. Also the meaning of a lexeme represented by a word form in a text constraints the semantic relations with other elements of the text. Thus, our hypothesis is that the linguistic knowledge of different levels can be exploited in order to predict word forms in text and to improve the quality of OCR of handwriting.

Obviously, the idea is not known. Typically, stochastic LMs are used during post-processing, e.g. [6],[7]. In the case of a limited vocabulary, grammars are used for prediction [7] (but our system can be classified as an almost open vocabulary system). In [6], a LM enriched with the information concerning Parts of Speech is introduced. However, the level of syntax is mostly limited to the stochastic modelling of sequences of "syntactic word classes"[1]. An attempt to the application of a parser based on stochastic context-free grammars in the OCR correction is presented in [18], but for a small training corpus.

Our goal was to investigate how the linguistic knowledge of different levels can influence the *correction by prediction* of the handwriting OCR for medical documents.

For the needs of the experiments a corpus (called KorMedIIS — the Polish acronym for ``The Medical Corpus of the Institute of Applied Informatics'') [12] of electronic medical texts has been collected from the database of some hospital for which the prototype OCR system is being constructed. The collected texts belong to several categories, but only *epicrisises* were used during our experiments. Epicrisises are short descriptions of a patient stay in hospital. An epicrisis is written when a patient is discharged from the hospital. A typical epicrisis includes larger passages of text, consists of several sentences (or phrases), reports some details of the patient stay and treatment, and copies often after the other documents. KorMedIIS includes presently 1 373 741 words in 15 961 epicrises and 1 334 590 words in texts of the other types. The complete lexicon derived from the whole corpus contained more than 34 000 words. The corpus was divided into the *Training Corpus* consisting of 12 691 epicrisises (1 006 146 words) and the *Testing Corpus* containing the remaining 3 600 epicrisises (367 595 words).

The collected electronic texts come from the last few years (since the introduction of an integrated computer system) but can be treated as representative for the older hand-written texts of the same type of the given hospital, e.g. they posses the identical structure, and were created during similar procedures of treatment.

2. MULTILEVEL PREDICTION AND CORRECTION

Prediction of a word form sequence can be based on linguistic knowledge concerning different levels of the natural language description:

- *statistical properties* of word forms in text — on the basis of a corpus a LM can be constructed, the model can be enriched with morphological information concerning origins of word forms — the construction of such LM is discussed in Sec. 3.
- *morpho-syntactic characteristics* of word forms — morpho-syntactic descriptions of word forms produced by the morphological analyser [17] can be used in constraining the possibility of a word form occurrence in some position in the text, see Sec. 4,
- and *syntactic structures* of the given language — the possible syntactic structures of the given language define the possible combinations of word

forms — an application of syntactic analysis to OCR correction is presented in Sec. 5.

The subsequent levels deliver knowledge of increasing complexity that results in the increasing complexity of the automatic analysis performed. Thus we assumed the model of gradual improvement by using the subsequent types of linguistic knowledge. The input to the processing is a list of possible *candidates*, i.e. possible recognitions, produced for subsequent word positions in a text. The list is produced by the word level classifier (i.e. of the graphical level) [14] and includes several candidates for a position. By the subsequent rejection of the least scored candidates we want to decrease the complexity of processing by tools on the higher levels. However, before putting all prediction models together, we wanted to analyse first the possible influence of different models, when applied separately — the experiments are presented in Sec. 3–5.

3. STATISTICAL LANGUAGE MODEL

First, we constructed a simple LM based on probabilities of *tri-grams*, i.e. sequences of three word forms [11]. The probability of a word form w_i in a text can be approximated by the *Maximum Likelihood Estimation* (MLE) [8]:

$$P(w_i|w_{i-2},w_{i-1}) = P(w_{i-2}, w_{i-1}, w_i)/ P(w_{i-2},w_{i-1}) = c(w_{i-2}, w_{i-1}, w_i)/c(w_{i-2},w_{i-1}) \qquad (1)$$

where $P(w_{i-2}, w_{i-1}, w_i)$ is the probability of a sequence of word forms, and $c(w_{i-2}, w_{i-1}, w_i)$ is the number of occurrences of the sequence in the training corpus.

For a sentence of n words there are k^n (here $=10^n$) different combinations of candidates. In order to efficiently search for the best combination of candidates we treat candidates as states in a *stochastic Markov process*. Because candidates are possible word forms, the probabilities of state transitions can be estimated on the basis of tri-grams.

As we are interested in the most consistent sequence of candidates from the linguistic point of view we want to look for a sequence of candidates maximising the probability of the whole path across the candidates for subsequent positions. We calculate the best maximal path by the algorithm called *Global Word Consistency* [11]. According to the algorithm, sets of candidates for subsequent positions in text are modelled as *HMM trellises* [8] and the search procedure follows the general scheme of the *Viterbi algorithm* (however, modified in order to calculate the maximal path, not maximal subsequent states).

When constructing tri-grams from the open vocabulary corpus, one immediately encounters the problem of data sparseness. We collected only 10^{-7}% of the possible word tri-grams ($365\,288/32\,302^3$) from the Training Corpus. We tested several methods of probability *smoothing*, surprisingly obtaining the best result with the simple Laplace smoothing [11]. The best result of the tri-gram, word form LM was 92.82% of the overall accuracy in recognition and 62.4% of the error reduction in relation to 85.9% — the accuracy of the word classifier alone and 96.6% — the maximal possible accuracy for the 10 candidates produced for a position by the word classifier.

The accuracy achieved by the word tri-gram LM is surprisingly good in relation to the theoretically unrestricted language of the corpus. However we observed a decreased accuracy of the model when applied to texts other than epicrisises. This suggests that the

LM built on word tri-grams lacks generality and even a small difference between the training and test corpora can be significant for the accuracy. Moreover, the coverage of tri-grams is very low for the open vocabulary. The coverage can be increased by grouping words into classes. In the case of an inflective language like Polish, the natural word form classes arise from the morphological features of word forms. Many word forms are derived from the same *base form*, e.g. up to 14 for a noun and 119 for a verb (including participles, gerunds etc.). We collected tri-grams of base forms from the corpora automatically disambiguated by the TaKIPI *tagger* [10]. A tagger is a program that chooses for each word form which is ambiguous among many possible morphological descriptions, the description that is appropriate for the given context. The best sequences of candidates were calculated by a modified algorithm of the *Global Base Forms Consistency*:

1. Each candidate is exchanged on the list with all its possible base forms — very often more positions are added to the lists of candidates.
2. The scheme of the Global Word Consistency algorithm is applied to the lists of candidate base forms; the probabilities are calculated as smoothed MLE: $P(b_i|b_{i-2},b_{i-1})$, where b_i is a base form (often one of several) of the i-th candidate in a sequence.
3. The best candidates are chosen according to the best base forms of the subsequent positions; in case of several candidates sharing the same base form, the first candidate on the initial list produced by the word classifier is chosen.

The accuracy of the base form LM was 92.85% and the error reduction was 62.9%.

In order to improve the result, we tried to combine both successful models, namely the word tri-gram LM and the base form tri-gram LM. However, taking into account the linguistic point of view, we built a two step mechanism of rejection and choice:

1. First, the Global Base Form Consistency algorithm is applied, but this time all candidates sharing the winning base form are preserved, the other ones are rejected.
2. All preserved candidates are restored on the shortened lists i.e. they exchange base forms.
3. The Global Word Consistency algorithm is applied to the shortened lists of candidates.

The combined model achieved 92.98% of accuracy and 64.2% of error reduction. The increase in the result shows that the combination of the two steps increased the ability of the LM to make generalisation.

4. MORPHO-SYNTACTIC MODEL

Both LMs of the Sec. 3 capture only direct local associations of word forms and do not take into account the morpho-syntactic properties of word forms. These properties define constraints on possible combinations of word forms, e.g. morpho-syntactic agreement of an adjective and a noun, and can be exploited in order to construct a structure sensitive LM.

The basis for the model was the *TaKIPI* tagger and the assumption that the tagger, while making decisions, reports the higher probability of a decision, as the more certain the decision is on the basis of learning examples. As the probabilities of tagger's decisions for

different words are not directly comparable [10] a measure of the confidence of decision for a word w, written CD(w) is calculated for each decision of the tagger:

$$CD(w) = max_{t \in tags(w)}(p(t)) - min_{t \in tags(w)}(p(t)) \quad (2)$$

where t is a tag, w — a word, $p(t)$ — the final probability of a decision.

TaKIPI generates higher CD measures for words syntactically consistent with some close context. The proper syntactic constructions are the majority in *KorMedIIS* (at least on the level of phrases). The tagger does not depend in its decisions only on one word. It reads morpho-syntactic properties of at least of -3 and +2 surrounding word forms, by using *simple operators*. Moreover, *complex operators*, which are built into many parts of *TaKIPI*, check also larger context. The complex operators express, e.g. the long distance morpho-syntactic agreement between parts of some syntactic construction. In the case of a free word order language like Polish, these agreements are the primary means of expressing the syntactic structure.

Thus, in order to check tagger's decision, we need to apply *TaKIPI* several times to different possible contexts of each candidate while processing candidate lists. The process goes across positions of the input. For each candidate of the position being currently processed, we pass to the tagger each possible permutation of candidates from the context of the $\langle -n,+m \rangle$ positions plus the best candidates from outside of the context up to the sentence boundaries. From the outside of the window the best candidates for that moment are taken:

- before the window — as evaluated by our algorithm,
- after the window — according only to the probabilities from the word classifier.

For each permutation, the CDs of processed candidates are collected. We decided to omit zero values of CD, i.e. for candidates which are morpho-syntactically non-ambiguous, and to calculate the average from non-zero CDs. This average is called a *context consistency measure* (CCM). The calculation of CCM for all candidates of a word w is defined below:

$$CCM(c) = max_{p \in perms(w)}[(\Sigma_{c' \in p} CD(c')) / number\_of\_non\_zero\_CDs(p)] \quad (3)$$

CCM used alone achieves a result comparable with the word classifier, i.e. about 82% of accuracy. However, after combination with the scores of the word classifier and introduction of some heuristics for unknown words [4], the result was increased to 89.02%.

An example of correction is given below: 1) the written sentence, 2) the recognition by the word classifier alone, 3) the combination of the word classifier with CCM and heuristics.

1) *) Przy przyjęciu w badaniu fizykalnym bez istotnych odchyleń od stanu prawidłowego.*
   (There was no serious deviations from the regular state in physical examination during admission.)
2) *Przy przyjęciu* ***n*** *badaniu* ***fizukalnym*** *bez* ***tstotnych*** *odchyleń* ***oo*** *stanu prawidłowego.*
3) *Przy przyjęciu w badaniu fizykalnym bez istotnych odchyleń* ***oo*** *stanu prawidłowego.*

5. CORRECTION BY PARSING

LM exploits the immediate context of word form co-occurrences and CCM is based on the local syntactic dependencies, i.e. in the distance of few words on average. However, the overall structure of a sentence (or a complex phrase) is often determined by longer range syntactic dependencies, e.g. the main verb predicate and its arguments. In order to capture dependencies of this type, we decided to introduce the next step of correction based on a *probabilistic parser*, i.e. a program performing syntactic analysis and assigning to each part of a syntactic structure some probability. A probabilistic parser can be lexicalised if the probabilities assigned to parts of the structure depend on the word forms attached to the terminal nodes of the *parsing tree*.

Unfortunately, there is no probabilistic parser of Polish, and moreover any robust Polish parser is not publicly available. A probabilistic parser can be constructed automatically on the basis of a corpus annotated by syntactic structures, however any bigger Polish corpus of this type has not been created yet. Thus, according to a kind help of Zdenek Žabokrstký and Tomasz Holan we decided to use their Czech parsers [5] adapted to Polish. In order to make adaptation possible, we constructed program for the automatic conversion of Polish tags in the IPI PAN Corpus (IPIC) [13] format into Czech tags in the Prague Dependency Tree Bank (PDT) format. Unfortunately, the criteria of dividing word forms into grammatical classes in IPIC and PDT are significantly different. The IPIC format follows strictly the syntactic and morphological criteria while the PDT format refers often also to semantic criteria, e.g. personal pronouns in the third person are one class in IPC but the corresponding Czech word forms are dived into two classes in PDT, namely: personal pronouns and possessive pronouns. In spite of using sophisticated heuristic rules in the conversion, a significant level of errors is still produced.

Next, the annotation of the manually disambiguated part of IPC was automatically converted to the PDT format. The data were used for adapting the rules of Žabokrstký's parser and re-training Holan's parser. After the manual inspection, we have chosen Žabokrstký's parser for the further experiments.

The crucial dictionary of verb subcategorisation (describing the arguments requested by verbs) is absent in the Polish version of Žabokrstký's parser and its accuracy is significantly decreased. However, we hoped that, for the needs of the OCR correction, more important is consistency in parser decisions across different sentences than its exact accuracy. Facing the lack of a syntactically annotated corpus, as the basis for the construction of a probabilistic parsing model, we decided to simulate an application of a probabilistic parser in OCR in the following way:

1. The Training Corpus was parsed by the Polish version of Žabokrstký's parser.
2. A probabilistic, syntactic model was built on the basis of the parsed corpus.
3. The parser was applied to sentences formed from candidates (sequences generated in the same way as in the moprho-syntactic model in Sec. 4).
4. Probabilities of the parser trees were estimated according to the chosen probabilistic parsing model on the basis of frequencies collected from the parsed corpus.
5. Candidates were selected in way maximising the probabilities of the parser trees.

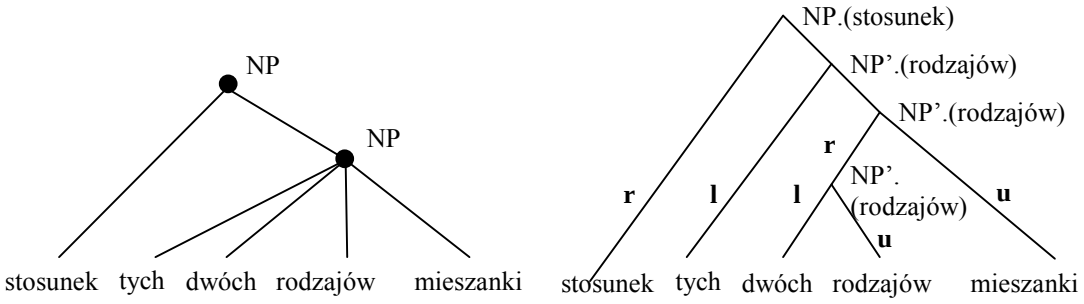

Fig.1. Example of a parsing tree: before (left) and after transformation.

The underlying assumption of the above procedure is that typical syntactic constructions, i.e. which occur more frequently in the parsed corpus than the other ones, receive higher probabilities. Thus, the sequences of candidates that can form such typical constructions will receive higher probabilities and will be chosen during recognition.

As the lexicalised, probabilistic parsing models express better accuracy [8], we decided to follow the model presented by Chelba & Jelinek [2], called *Structured Language Model* (SLM). SLM is based on a lexicalised syntactic representation of natural language expressions. The representation is a binary parsing tree developed in the left-to-right manner (e.g. the tree in the right part of Fig. 1), i.e. the structure created for some already recognised initial subsequence of tokens is complete to the left of that point and is never amended later during the further processing. New branches or token nodes (leaves) are added only to the right of that current point. Originally, SLM was trained on the *Penn Tree Bank* (PTB) corpus of English [9] which is manually annotated with syntactic trees. As SLM works on binary trees with *headwords* assigned to every node, all trees from the PTB corpus had to be transformed to this structure. A syntactic *headword* is a main element of a phrase and can represent the phrase and its syntactic properties, e.g. in the case of a *Noun Phrase* (NP) its headword is the main noun with the morphological case identical to the case of the whole noun phrase. In the PTB corpus, headwords are not assigned in the annotation, and had to be added automatically by performing *headword percolation* i.e. selected lexemes from leaf nodes are copied to the internal nodes according to manually constructed, context sensitive rules. The result is visible in the right tree in Fig. 1. In SLM, the headword percolation is combined with *binarization* i.e. transformation of n-branching structures (the result of a context free grammar) into 2-branching structures by introducing additional internal nodes labeled with artificial syntactic categories, e.g. NP' created as an internal node of a NP tree. Binarization is performed according to the manually constructed rules which are defined for syntactic categories of internal nodes. In Fig. 1 the structure of the tree on the right side is the result of the binarization of the left one — the additional syntactic category NP' was introduced.

In the case of the Žabokrstký's parser we need one additional step of transformation, as it produces a *dependency structure* of a sentence. The structure consists of links expressing relations between words, but the internal nodes are labelled only with word forms without any syntactic categories assigned to them. We constructed additional rules operating in a bottom up direction gradually identifying the head elements (word forms assigned to the internal nodes are not necessarily identical with headwords) and on this basis assigning syntactic categories to nodes. In Fig. 1 the left tree is already transformed from

the dependency structure to the phrase structure, and the right tree is the result of the headword percolation and binarization.

SLM defines the parsing tree as the result of three kinds of operations performed sequentially: word form prediction, tagging and a *parser actions* (one or two). There are 3 possible parser actions: `adjoin-right`, `adjoin-left`, and `unary`. The first two add a new branch or create a place for a new branch of the parsing tree, and the third action creates a new unary branch — a leaf representing a token from text. During the construction of the model, the sequence of actions is determined by the structure of a transformed tree. During prediction, SLM defines parsing as a stochastic, indeterministic process, in which the probabilities are assigned to all operations, and the triples of operations: word form prediction, tagging and a parser actions are sequentially repeated until the whole sequence of input tokens is not processed. The output of SLM parser is a set of partial parsing trees (a forest generated for the given input) — in the case of a proper natural language expression as the input the set consists of exactly one tree.

The probability of a word sequence W and its complete parse T is defined as following:

$$P(W,T)= \prod_{k=1}^{n+1} [P(w_k|h_0,h_{-1})\ P(t_k|w_k, h_0.tag, h_{-1}.tag) \prod_{i=1}^{Nk} P(p_i^k|h_0,h_{-1})] \qquad (1)$$

where w_k is a token on the k-th position, t_k — morpho-syntactic tag assigned to the k-th token, h_0,h_{-1} are, respectively: the recently created and the one before it internal nodes (pairs: headword, tag), $h_i.tag$ — the tag of the last i-th node, and p_i^k — the i-th parser action performed during processing of the k-th token.

In SLM, probabilities of the operations are estimated on the basis of frequencies collected from the trees of the corpus. SLM selects the best path across all possible sequences of operations for the given input applying the proposed *synchronous multistack search algorithm* (a modification of the *beam search* algorithm). Each stack contains parsing trees including the same number of parsing actions.

We prepared our corpus of trees by parsing the whole Training Corpus with the Žabokrstký's parser. As we wanted to achieve consistency between analysis in the corpus and analysis during recognition, the same deterministic parser was used during recognition:

1) for each candidate c_i^k on the position k and each generated sequence $W(c_i^k)$ of candidates (including the c_i^k candidate)
   tokens of S are tagged by *TaKIPI*
   the Žabokrstký's parser is applied and a tree T is produced
   the probability $P(W,T)$ is calculated according to the model SLM
2) for the position k a candidate c_m^k is selected such that $W(c_m^k)$ maximises $P(W,\bullet)$

In the step 1) the candidates are selected for the sequence W in exactly the same way, as it was done in the morpho-syntactic model in Sec. 4, i.e. all possible sequences from a small window and the best candidates from the outside of the window. In the step 2) we want to choose a candidate c_*^k for the k position such that a sequence of candidates including c_*^k receives a lexicalised parsing tree with the highest probability. The probabilities of trees generated during recognition are calculated according to equation (1) on the basis of frequencies of parser operations collected from the parsed Training Corpus. The probabilities are smoothed by the Laplace algorithm. We did not need to apply any

search algorithm, as each tested sequence of candidates were parsed by the deterministic Žabokrstký's parser and there was only one parsing tree for each sequence. The tree probability was calculated on the basis of its already built structure.

The results of the correction by parsing when applied alone, as the only corrector, are presented in Tab. 1. The experiments were performed on the same test set as the previous ones Sec. 3 i 4. The results are significantly lower than achieved by application of the two other models. However, we could not apply the same improving heuristics as it was done in the morpho-syntactic model which probably results in the decrease of the accuracy by 1-2%. The probabilities of trees directly depend on the number of nodes created in the tree — the number of parser operations. Winning candidates in one sentence can have very different values of probability (i.e. maximal probability of a parsing tree). We could not set any global threshold, and what was even worse, we could not find any method to combine the probabilities of winning candidates with the scores obtained from the word classifier as it was done in the morpho-syntactic model. We tested also different variants of candidate evaluation on the basis of parsing trees, namely:

- exchange of the maximum probability of trees for the given candidate (across different possible sequences) to the average of candidate tree probabilities,
- and different procedures of normalising tree probability by the number of nodes or parser actions.

However, the results were even worse. As one of the potential problems, we identified the large percentage of unknown words caused that SLM assigned higher probabilities to syntactic structures including unknown words. However, when we applied a morphological guesser — the SLM+G model in Tab. 1, no improvement was observed. In the SLM+G model unknown words are often identified with some known ones, but not the actual ones from the input text. After that, the parser recognises among false candidates some proper syntactic structures, that misleads the final recognition.

Table 1. Results of the correction by parsing.

SLM	SLM+I	SLM+G	PCFG	PCFG(Poleng)
78.72%	77.42%	76.08%	44.51%	67.59%

In the SLM+I model, we tried to decrease the probability of constructions including unknown words by assigning each unknown word to its own, singleton syntactic category, e.g. ign_konieta (the symbol of the unknown category: `ign` plus the misspelled form of a *woman*). The parser behaved in a different way, but the result was roughly the same.

In order to better asses the results achieved by the SLM model, we implemented also a simpler syntactic model based on non-lexicalised *Probabilistic Context Free Grammar* (PCFG), e.g. [8]. In PCFG, n-branching parsing trees are used, nodes are labelled only with syntactic categories, the probabilities are assigned to grammar productions (represented by nodes in a tree), and the probability of a tree equals the multiplication of node probabilities. Models based on PCFG express usually lower accuracy than lexicalised models, as they do not take into account word forms and their associations, that is the case here, too, see Tab. 1.

6. CONCLUSIONS

Application of more sophisticated models not always brings good results, and it is clearly the case of the correction of OCR of medical handwriting. The simple LM of combined base form and word form n-grams outperforms the other two models. The result of correcting by parsing is even worse than of doing nothing, i.e. leaving the decision of word classifier unchanged (almost 82%). There are two reasons for this situation. Firstly, the class of text is very specific: vocabulary is limited, documents are written according to some small set of schemes, the number of different authors is limited, and the authors are often copying after themselves. Secondly, there are many mistakes in electronic texts used for training the models, the mistakes decrease performance of the more sophisticated models.

As we use a kind of Viterbi search algorithm in the n-gram LM, we can easily obtain a the best path across candidates but we cannot easily limit the number of candidates for the subsequent positions. In order to do this we should make the full search, that would be computationally very expensive. Delivering to the morpho-syntactic model, and especially to parsing model all candidates decreases the results of the both models.

The n-gram LM takes into account only local co-occurrences of word forms. Models of semantic similarity of word forms, e.g. [8] [16], extracted from their co-occurrences in lexico-syntactic contexts can be used as the basis for a measure of semantic consistency of a sequence of word forms in larger passage of text. As these measures do not depend on a precise description of syntactic structure, as the morphosyntactic model and parser do, and are closer to n-gram models, one can expect, that the result of their application to correcting OCR should be better than the former ones.

7. ACKNOWLEDGEMENT.

This work was financed by the Ministry of Education and Science project No 3 T11E 005 28 and partially from the funds assigned to the Institute of Applied Informatics, Wrocław University of Technology.

BIBLIOGRAPHY

[1] BUNKE, H. Recognition of Cursive Roman Handwriting - Past, Present and Future Proc. of the 7th International Conference on Document Analysis and Recognition (ICDAR'03), IEEE, pp. 448-460, 2003.
[2] CHELBA, C., JELINEK, F. In Boitet, C. & Whitelock, P. (ed.) Exploiting Syntactic Structure for Language Modeling. Proc. of the 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics, pp. 225–231, Morgan Kaufmann Publishers, 1998,
[3] COLLINS, M. J. A new statistical parser based on bigram lexical dependencies. Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, pp. 184–191, 1996
[4] GODLEWSKI G. & PIASECKI M., SAS J. In Brailsford, D. F. (ed.) Application of Syntactic Properties to Three-level Recognition of Polish Hand-written Medical Texts Proceedings of 2006 ACM Symposium on Document Engineering, pp. 115–121 ACM, 2006.
[5] HOLAN, T., ŽABOKRSTKÝ, Z. Combining Czech Dependency Parsers. In [15] pp. 95–102.
[6] MALBURG, M. Comparative Evaluation of Techniques for Word Recognition Improvement by Incorporation of Syntactic Information. Proceedings of ICDAR `97, Ulm, Germany., IEEE, 1997.

[7] KOERICH, A. L.; SABOURIN, R. & SUEN, C. Y. Large vocabulary off-line handwriting recognition: A survey Pattern Anal Applic, Vol. 6, pp. 97-121, 2003.
[8] MANNING, C. D., SCHÜTZE, H. Foundations of Statistical Natural Language Processing The MIT Press, 2001
[9] MARCUS, M. P.; SANTORINI, B., MARCINKIEWICZ, M. A. Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics, Vol. 19, pp. 313–330, 1994.
[10] PIASECKI, M., GODLEWSKI, G. Effective Architecture of the Polish Tagger. In [15], pp. 213–220.
[11] PIASECKI, M. & GODLEWSKI, G. In Maglaveras, N. et. al. (ed.) Language Modelling for the Needs of OCR of Medical Texts Biological and Medical Data Analysis. 7th International Symposium, ISBMDA 2006, Thessaloniki, Greece, December 7-8 2006, LNCS, Springer, 2006
[12] PIASECKI, M., GODLEWSKI, G., PEJCZ, J.: Corpus of medical texts and tools. In: Proc. of Medical Informatics and Technologies 2006, pp. 273–280 Silesian University of Technology, 2006.
[13] PRZEPIÓRKOWSKI A. The IPI PAN Corpus Preliminary Version Institute of Computer Science PAS, 2004.
[14] SAS J., LUZYNA M. Combining character classifier using member classifiers assessment, Proc. of 5th Int. Conf. on Intelligent Systems Design and Applications, ISDA 2005, pp. 400–405, IEEE Press, 2005.
[15] SOJKA, P.; KOPECEK, I., PALA, K. (ed.) Proceedings of the Text, Speech and Dialog 2006 Conference LNCS, Springer, 2006
[16] WIDDOWS, D. Geometry and Meaning. CSLI Publications, 2004.
[17] WOLIŃSKI, M. Morfeusz — a practical tool for the morphological analysis of Polish. In Kopotek, M. A.; Wierzchoń, S. T. & Trojanowski, K. (ed.) Proceedings of the International IIS: IIPWM'06 Conference held in Zakopane, Poland, June, 2006, pp. 511–520, Springer, 2006.
[18] ZIMMERMANN, M., BUNKE H. Parsing N-best Lists of Handwritten Sentences 7th Int. Conference on Document Analysis and Recognition, IEEE Computer Society Press, 2003.