Navarro Cerdan, JR.; Arlandis Navarro, JF.; Llobet Azpitarte, R.; Perez-Cortes, J. (2015). Batch-adaptive rejection threshold estimation with application to OCR post-processing. Expert Systems with Applications. 42(21):8111-8122. doi:10.1016/j.eswa.2015.06.022.

Batch-Adaptive Rejection Threshold Estimation with application to OCR Post-processing

J. Ramon Navarro-Cerdan*, Joaquim Arlandis, Rafael Llobet, Juan-Carlos Perez-Cortes

Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Valencia, Spain

Abstract

An OCR process is often followed by the application of a language model to find the best transformation of an OCR hypothesis into a string compatible with the constraints of the document, field or item under consideration. The cost of this transformation can be taken as a confidence value and compared to a threshold to decide if a string is accepted as correct or rejected in order to satisfy the need for bounding the error rate of the system. Widespread tools like ROC, precision-recall, or error-reject curves, are commonly used along with fixed thresholding in order to achieve that goal. However, those methodologies fail when a test sample has a confidence distribution that differs from the one of the sample used to train the system, which is a very frequent case in post-processed OCR strings (e.g., string batches showing particularly careful handwriting styles in contrast to free styles).

In this paper, we propose an adaptive method for the automatic estimation of the rejection threshold that overcomes this drawback, allowing the operator to define an expected error rate within the set of accepted (non-rejected) strings of *a complete batch of documents* (as opposed to trying to establish or control the probability of error of a *single string*), regardless of its confidence distribution. The operator (expert) is assumed to know the error rate that can be acceptable to the user of the resulting data. The proposed

*Corresponding author at: Instituto Tecnológico de Informática (Universitat Politécnica de Valencia), Tel.: +34 963877242; Fax.: +34 963877239

Email addresses: jonacer@iti.upv.es (J. Ramon Navarro-Cerdan), arlandis@iti.upv.es (Joaquim Arlandis), rllobet@iti.upv.es (Rafael Llobet), jcperez@iti.upv.es (Juan-Carlos Perez-Cortes)

system transforms that knowledge into a suitable rejection threshold.

The approach is based on the estimation of an expected error *vs.* transformation cost distribution. First, a model predicting the probability of a cost to arise from an erroneously transcribed string is computed from a sample of supervised OCR hypotheses. Then, given a test sample, a cumulative error *vs.* cost curve is computed and used to automatically set the appropriate threshold that meets the user-defined error rate on the overall sample. The results of experiments on batches coming from different writing styles show very accurate error rate estimations where fixed thresholding clearly fails. An original procedure to generate distorted strings from a given language is also proposed and tested, which allows the use of the presented method in tasks where no real supervised OCR hypotheses are available to train the system.

Keywords:
Rejection threshold, OCR post-processing, Language models, Weighted finite-state transducers, Error *vs.* cost curve, Cumulative error *vs.* cost curve, OCR error-generation model

---

1. Introduction

In many OCR and text recognition systems, a post-process addressed to the verification or correction of the errors yielded by the classifier is performed, due to its beneficial impact on the system accuracy. In many cases, a language-model-based correcting technique can be applied to find the best transformation of an OCR hypothesis into a string compatible with a given set of linguistic constraints.

Very different techniques have been employed to implement the post-processing of the OCR hypotheses (usually, strings of characters). Some examples can be found in Hull & Srihari (1982); Tong & Evans (1996); Perez-Cortes et al. (2000); Kolak & Resnik (2005); Llobet et al. (2010). Most of them provide or can be easily modified to provide an estimation of the effort needed to make the output from the OCR classifier comply with the language constraints. This estimation is often called **transformation cost**. Sometimes this cost is substituted by an inversely related measure called **correction confidence** or **reliability index**, reflecting the likelihood that the OCR hypothesis agrees with the model.

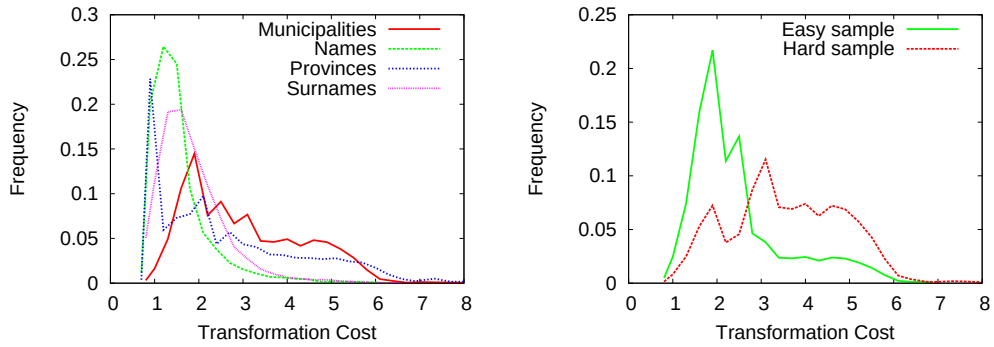

Figure 1: Transformation-cost distributions of strings belonging to: (left) samples from different language models (described in Table 1), and (right) two different samples of the same language, from carefully written (*Easy Test*) and carelessly written (*Hard Test*) Spanish Municipalities (sample composition described in Section 6.1).

Using a threshold on the transformation cost to reject the less reliable hypotheses, a variable level of (expected) accuracy can be imposed on the output of the fully automatic recognition process. The feature of allowing the user to specify an acceptable level of the expected error instead of having to deal with a threshold in an unfamiliar task-dependent scale is a clear advantage for the operator. This capability has important implications in many practical cases: the complete process is often requested to meet a maximum acceptable amount of erroneous transcriptions within the set of non-rejected strings in a batch of documents (known as false acceptance rate or simply **error rate**) to ensure a quality of service; and the rejected strings are usually sent to a high cost manual data-entry process that should be kept as limited as possible All these considerations suggest that a trade-off, where the threshold selection plays an important role, exists and has a significant impact on the practical and economic performance of the system.

To decide on the acceptance or rejection of a single string, a simple threshold is relatively straightforward to estimate, but to maintain a control of the error rate of a batch of documents, some strings that would be rejected might be accepted and vice-versa depending on the remaining strings of the batch. If the measurements suggest that the quality of the set is good (or bad), the threshold can be higher (or lower), accordingly.

The optimization of this process is not straightforward. In an OCR system, the number of rejections is not predictable because it depends on the

particular task at hand, being highly sensitive to factors such as the handwriting style, scanning process, image quality, field registration, etc, as well as the characteristics of the language model used (e.g., its perplexity). For example, for a given threshold, the amount of rejected strings in two sets of documents can be very different if the first one is composed of carefully written strings and the second set is poorly written. Thus, if we take a sample of OCR strings (observations) and compute its transformation costs using a given post-processing method, like the one described in Section 3, the distribution obtained can vary for different language models, as can be seen in Figure 1 (left), as well as for different samples of a single language, as shown in Figure 1 (right). This means that choosing a consistent rejection threshold on the transformation cost is difficult, since the number of accepted/rejected strings for a given threshold value will vary depending on the characteristics of both the language model and the sample processed. In fact, a slight variation of the threshold value can lead to unpredictable changes on the ratios of accepted/rejected strings, as well as on the error rate, as shown in Figure 2. Therefore, in most cases, different thresholds should be applied to different samples to obtain the same error rate within the set of accepted strings.

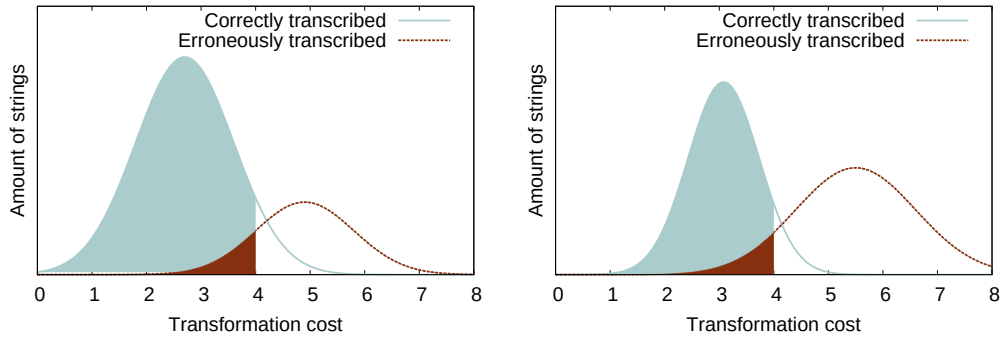

Figure 2: If the cost distribution varies among different samples, e.g., a carefully written sample (left) and a carelessly written sample (right), a given threshold can lead to different error rates for the accepted instances (ratio between red and blue-filled areas for a fixed threshold equal to 4). Therefore, different thresholds should be applied to obtain the same error rate.

Traditional analytic tools like Error-Reject Trade-off Chow (1970), Receiver Operating Characteristic (ROC) Fawcett (2006), and Precision-Recall Rijsbergen (1979) (and their different variations), have provided useful infor-

mation for analyzing and comparing classifier performances, based on measurements obtained from a training sample. Particularly, predictions on the relationship between the rejection threshold and several indexes like error rate, precision, accuracy, number of rejections, false positive and false negative rates, can be established from such tools. Nevertheless, the accuracy of those predictions is strongly conditioned by the distribution of the confidence values found in the training sample used. In other words, predicting the error rate of a new sample entails the assumption that a similar confidence distribution is expected in it. However, in the task described here, this assumption can be unacceptable, as explained above: the amount of symbol errors of an OCR classifier can widely vary for different samples depending on many factors, and consequently, the distribution of the post-processing transformation costs, can vary too. Therefore, in this case, applying a fixed threshold to different samples will not guarantee meeting a pre-specified error rate.

In this work, we consider the hypothesis that a probability distribution of erroneous transcriptions can be estimated from the transformation costs produced by the application of a language model, and that it can be used to predict the error rate of a set of strings of the language, regardless of its cost distribution. Thus, in Section 4, given a set of transformation costs corresponding to a supervised sample of OCR hypotheses, the error probabilities associated to each cost (*Error vs. Cost distribution*) are obtained. In Section 5, we propose an approach for adaptive rejection thresholding, where the Error *vs.* Cost distribution of a language model is used to find the rejection threshold to be applied on a whole batch of strings in order to meet a given target error rate. We tested this approach in two scenarios:

- The transformation costs are obtained from a real sample of OCR hypotheses, and used to compute the Error *vs.* Cost distribution. Performance evaluation in this scenario is presented in Section 6.

- The Error *vs.* Cost distribution of a new language is automatically estimated, with a method based on the generation of synthetic OCR errors from the positive sample used to build the language model. No supervision is needed in this case, avoiding the time-consuming process of optical recognition and manual labeling of a significant amount of strings of the language. This is particularly important in practice when new language models are needed frequently for short batches of documents (even if they are subsets or special variants of previous language

models), or in case of tasks where the labeling process is not possible or convenient. It is described and evaluated in Section 7.

2. Background and related work

2.1. Quality control

Industrial Quality Control is commonly approached from two points of view. The first one takes into account the quality acceptable by the recipient of the product or service and the quality level bearable by the producer. These qualities correspond to what is known in the literature as the risks of the producer and the consumer. The inspection plans for sample-based statistical quality control are designed from the Characteristic Operation Curve and these risks. In Paladini (2000), an expert system is presented to help in decision-making for tasks such as the determination of the need for such inspection and the type of inspection to be performed.

A second view is based on the design of data-mining models to account for the confidence on the produced elements taking as input explanatory variables measured during the production process itself. These models aim to predict the confidence on the final products. In Köksal et al. (2011), an analysis of data management practices and data mining applications related to manufacturing quality is presented.

Our proposal combines both paradigms by establishing a relationship between the expected error according to the explanatory variable "transformation cost", from which a dynamic threshold is found to generate an acceptable final error (risk of the producer and the consumer). The expected error can be seen as the knowledge that an expert operator (producer) has determined along time with the contact or negotiation with the consumer and can therefore express explicitly. The proposed system applies a model of this knowledge to convert it into a numerical threshold with no obvious or explicit meaning.

2.2. Large-scale OCR systems and post-processing

The different OCR and text recognition systems available can be categorized depending on the specific type of task they address and on their functionalities. A typical architecture for industrial-scale batch OCR systems includes an image acquisition and pre-processing phase; feature extraction and character or word classification; and, finally, an additional post-processing phase where the strings proposed by the classifier (potentially, having errors)

are constrained to be compatible with the rules inherent to the document, field or task under consideration.

Post-processing OCR hypotheses, typically, has a significant positive impact on the global system performance. For instance, in a handwritten form recognition system, if a Spanish name is expected to be found in a field, and the output of the classifier for that field is the string "HARIA" (where the character "H" should be "M"), a post-process technique could take "HARIA" as input and produce "MARIA" as output. A correction confidence is often provided along with the output string.

Very different techniques have been employed to post-process OCR hypotheses, which are usually strings, although they can be sequences of vectors of *a posteriori* probabilities or other more complex structures. Word and sentence level models typically apply dictionary search methods, Hidden Markov Models, Edit Distance-based techniques, and other character or word category transition models. In Hall & Dowling (1980), an excellent survey of approximate string search methods is presented. Traditionally, simple methods lookup on a lexicon to validate input strings. More complex methods are based on n-grams or finite-state machines Berghel (1987); Breuel (1994); Farooq et al. (2009), where an input string is parsed and the set of transitions with the lowest cost (highest probability) determine the output string. Some classical approaches parse the string provided by the symbol-input system, using a language model, and an Error Model and apply the classical Viterbi Algorithm Neuhoff (1975); Amengual & Vidal (1998) to find the maximum likelihood path on a finite-state machine representing a regular grammar.

In the context of language modeling, many works have been carried out for Continuous Speech Recognition tasks Jelinek (1993). Although the requirements are very different, most basic techniques used in that field can be applied to OCR tasks with little modification. Thus, several works use language modeling techniques for error-correcting applied to OCR and text recognition tasks, either on constrained or unconstrained environments Hull & Srihari (1982); Tong & Evans (1996); Perez-Cortes et al. (2000); Kolak & Resnik (2005); Llobet et al. (2010). Confidence measures reflecting the likelihood that a given OCR hypothesis belongs to the model are provided by many of them. In fact, some works on typical Natural Language Processing applications, including OCR and text recognition, propose improvements on confidence measures targeted to yield reliable procedures Bertolami et al. (2006); Pitrelli et al. (2006); Schlapbach et al. (2008); He et al. (2009). However, they are not addressed to solve the specific goal of rejecting output

strings based on a pre-defined error rate for a whole test-set (or batch of documents), as is intended in the present work.

2.3. Rejection thresholding

Rejection threshold optimization has been broadly studied in Machine Learning and Statistics. Generic and task-dependent approaches have been developed to minimize (or at least allow some control of) the number of Type I and Type II errors produced after the application of a rejection threshold to the observations of a sample. A Type I error is defined as the rejection of a potentially true null hypothesis, while a Type II error is the failure to reject a false null hypothesis.

In the context of OCR post-processing based on language models, we considered as null hypothesis: "The output string provided by the post-processing is the string that the user meant to write". Where, for supervised samples, the string that the user meant to write is represented by a ground truth label (belonging to a given language). Under these assumptions, a Type I error corresponds to a False Negative and a Type II error corresponds to a False Positive (FP). The rate of "erroneously transcribed" strings among the accepted (non-rejected) instances, i.e., $FP/(FP+TP)$, where TP is the number of accepted strings that satisfy the null hypothesis, is commonly known as error rate in a system having a reject option, where $(1-\text{error rate})$ is called precision. Thus, in our case, controlling the error rate entails limiting the amount of FP by optimizing the rejection threshold.

The problem of threshold optimization arose originally in Signal Detection Theory and now is found in multiple applications of Machine Learning, and Pattern Recognition in particular, as well as in a variety of scientific applications from social sciences (e.g., Financial and Economics or Psychology, among others), Bioinformatics and Diagnostic Systems. Also, from Expert and Intelligent Systems, applications involved with quality control deal with controlling FP and FN, which could entail particular types of threshold optimization, Wu et al. (2011). In many of such applications, error risks are a serious concern and, consequently, the rejection threshold plays a significant role and has been extensively studied.

In the scope of Pattern Recognition, the early work of Chow (1970) established the basis of the error-rejection trade-off that arises in classification problems. Chow described an optimum rejection rule based on the conditional *a posteriori* probabilities provided by a classifier, and presented a general relation between error and reject probabilities. Many subsequent works

have been based on the Chow's rule to propose contributions on the use of rejection in recognition systems, like class-selective rejection with performance constraints Grall-Maës & Beauseroy (2009) or class-related thresholds Fumera et al. (2000). Other works dealing with error risk and rejection optimization are based on improving the use of ROC curves from which many analysis and interpretations can be found in the literature Fawcett (2006).

Nevertheless, as detailed in Section 1, the estimations obtained using those approaches are strongly conditioned by the distribution of the confidence values in the sample used to train the system, which prevent them to be used for estimating the error rate on OCR batches having different writing styles. Avoiding this dependence is one of the targets of our work. In this regard, we can mention the work of Landgrebe et al. (2006) where a factor to tune the number of expected false positives is introduced in ROC curves (P-ROC) to deal with imprecise environments. This can be seen as an effort to compensate differences between training and test distributions.

So far, we are aware of very few works in the literature of Machine Learning and Intelligent Systems that propose solutions applicable to this particular problem. In their work, Li & Sethi (2006b) proposed an approach to design classifiers under controlled error rate requirements for the case of a two-class problem. While their method solves related goals and could be extrapolated to our task to a certain degree, its implementation requires using two thresholds and different functions to compute the set of rejected observations. Hanczar & Dougherty (2008) applied the Li proposal to classification in gene expression data, and Li & Sethi (2006a) extended their work to Active Learning. None of these studies is focused on OCR post-processing or suggest the option to generate synthetic samples to be used for error estimation.

Another similar approach to the one presented in this paper is proposed by Serrano et al. (2014). Here, the problem of error prediction is addressed in the context of interactive-predictive handwriting recognition. The purpose was to assist the user in locating possible transcription errors: the user decides on a maximum tolerance threshold for the recognition error and the system adjusts, interactively, the required supervision effort on the basis of an estimate for this error. For a given token being supervised, the error estimation is based on the previous user-supervised tokens using a similar strategy to the one proposed in the present work, but no directly applicable to a complete sample, particularly when no supervised information is available.

In summary, none of the previous works addressing the topic of auto-

matic control of the recognition error rate solves the problem of estimating a threshold to be applied on a whole sample of OCR post-processed strings, for which a maximum error rate is requested, particularly when different samples may have different confidence distributions. As we stated before, it is often the case in some real applications related to OCR post-processing, where input batches with different degrees of recognition difficulty have to be processed, and different target error rates can be demanded.

In Arlandis et al. (2010), we presented previous results involving the estimation of the expected error rate distribution of an unknown language model from a training set composed of known language models using regression techniques. In the present work, a new, more flexible, approach giving better results is proposed. Extended datasets, and more complete experiments are presented using an updated algorithm for the language and error models.

3. Post-processing algorithm and language models used

A technique based on Weighted Finite-State Transducers (WFSTs) combining language, hypothesis and error models has been used to post-process the OCR hypotheses Llobet et al. (2010). It is based on a finite-state transducer built from a formal grammar that encodes the strings in the lexicon or language sample. In this case, a k-Testable Language in the Strict Sense is inferred from a set of available language strings. The transducer is composed with other two WFSTs, representing the error model and the candidate OCR hypothesis, including a posteriori probabilities from the OCR classifier. The weight in each individual transition of this final WFST is the negative logarithm of the transition probability that results from the product of probabilities obtained from the WFST composition operation (computed as a sum of logarithms). Then, the shortest path in the final transducer shows which is the most likely string in the model, and the average weight of the transitions in the selected path is used as the transformation cost.

This technique has been applied to OCR hypotheses obtained from four fields belonging to forms with handwritten contents corresponding to four different field-level deterministic language models (languages where the complete lexicon of valid strings is known Perez-Cortes et al. (2000)) fairly usual in commercial forms. They are: *Names* (which include simple and compound names) and *Surnames* (mostly, simple ones) from the last census of Spain (complete), with probabilities derived from their frequencies; all the Spanish

Table 1: Sizes (number of complete strings) of the languages and available samples, and sample error rates at 0% rejection. The average number of characters per string of each language is also shown. The column "Freq." indicates if the language uses frequencies (probability of occurrence of each string) or not (all strings considered equiprobable).

Language	Lang. size	Avg. length	Freq.	Sample size	Error rate
Names	66,363	10.78	Yes	5,630	2.73%
Surnames	97,157	7.49	Yes	12,100	4.42%
Municipalities	8,201	11.67	No	8,280	19.81%
Provinces	52	7.38	No	8,400	11.64%

Municipalities; and all the Spanish *Provinces*. These languages have been chosen since they are representative of real tasks and span a wide range of sizes and complexities, and the samples have been obtained from document batches in a real workflow using the same OCR classifier.

The models have been built using a grammatical inference algorithm to build a WFST that accepts the smallest k-Testable Language in the Strict Sense (k-TS language) consistent with the strings of the language Garcia & Vidal (1990). The set of strings accepted by this automaton is equivalent to the language model obtained using n-grams, for $n = k$. The stochastic extension of the basic k-TS language is performed through a maximum likelihood estimation evaluated according to the frequency of utilization of each path by the language strings. To obtain deterministic language models, a value of k equal to the longest string in each language has been used.

The most relevant features of the languages and sample sets used are shown in Table 1. As expected, the WFST-based parsing algorithm attained lower error rates when the language models were inferred with frequencies. Figure 1 (left) shows the transformation cost distribution of the strings belonging to the samples described in Table 1.

4. Modeling the *Error vs. Cost* distribution

Given a language model and a set of transformation costs obtained using a post-processing algorithm with a sample of supervised OCR hypotheses (for which ground-truth transcriptions are available), a smoothed error rate curve as a function of the cost c can be computed using the equation,

$$H(c, w) = \frac{|S^{-}_{c,w}|}{|S_{c,w}|} \tag{1}$$

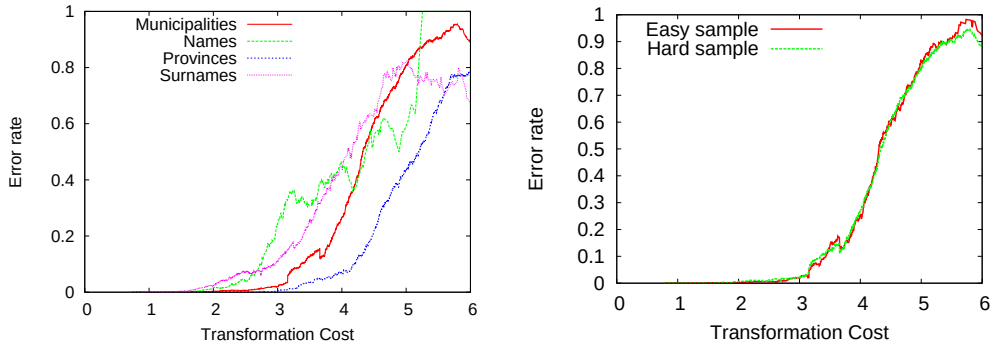

Figure 3: On the left, EC curves of the samples of the language models described in Table 1 (their corresponding cost distributions are shown in Figure 1, left). It is clearly shown that different languages gave rise to different EC curves. On the right, very similar EC curves were obtained from two samples of the same language model having different cost distributions (corresponding to *Municipalities* as shown in Figure 1, right).

where w is a smoothing (rectangular) window size parameter, $|S^{-}_{c,w}|$ is the number of strings erroneously transcribed into incorrect strings (belonging to the language, but different to those intended by the writer) having a cost between $c - w$ and $c + w$, and $|S_{c,w}|$ is the total number of strings having a cost also in that interval. More complex window functions could also be used.

The proposed ratio can be interpreted as the probability of a cost c to arise from an erroneously transcribed string. For a given language model, H, or the Error *vs.* Cost (EC) function, can be used as a source of information to decide the appropriate cost threshold to use when we want to set the expected error rate of a new sample, as explained in Section 5.

Figure 3 (left) plots the EC curve of each sample of the language models described in Table 1 using $w = 0.25$, based on its corresponding ground-truth transcriptions. The figure shows that the probability of an OCR hypothesis being wrong for a given cost can become very different for different language models. This clearly suggest that different thresholds will be needed to control the amount of false positives in each case.

Conversely, Figure 3 (right) shows that samples from a language model having different cost distributions give rise to very similar EC curves, which suggests that their corresponding error rates could be controlled using a single EC function for that language.

5. Batch-adaptive rejection threshold estimation

The error *vs.* cost function represented by H allows to obtain a direct estimation of the probability that an OCR hypothesis is wrong, based on its transformation cost. So, for a single string, we can simply look up H, and decide to reject it if the error estimation is higher than an error threshold.

But, if the goal is to control the error rate of a set of observations, i.e., a sample or batch of strings rather than a single OCR string, then different rejection thresholds are typically needed for different samples to achieve a given average error rate: string sets having a small number of OCR errors will give rise to low correction costs which will imply higher thresholds, while batches with a large number of OCR errors will require lower thresholds for the same target error rate.

In order to meet a specified (target) error rate ϵ on a sample of a language, a rejection threshold $\mathcal{T}$ can be calculated from a cumulative averaged version of H of its corresponding language model. Thus, let C be a sequence of transformation costs associated to a sample of OCR hypotheses sorted increasingly,

$$C = \{c_1 \dots c_i, c_{i+1} \dots c_n\}, \quad c_1 \leq c_i \leq c_{i+1} \leq c_n,$$

an estimation of the error rate incurred by accepting the subset of observations with costs smaller than or equal to c_i in C can be computed as follows:

$$E(c_i) = \sum_{c=c_1}^{c_i} \frac{H(c, w)}{i}, \tag{2}$$

and the rejection threshold $\mathcal{T}$ associated to a given target error rate ϵ can be obtained as:

$$\mathcal{T}(C, \epsilon) = \max_{E(c_j) \leq \epsilon} (c_j)$$

The $\mathcal{T}$ value sought is the highest cost where the curve E reaches ϵ. Since the curve can have local minima, the highest c_j is chosen to maximize the number of accepted strings for a given ϵ.

Examples of error rate estimations using the function E, or Cumulative Error *vs.* Cost (CEC) function, are shown in Figure 4, where estimated CEC curves from samples of two languages, along with their corresponding cumulative real error curves, are plotted. Real errors at a cost c_i were computed

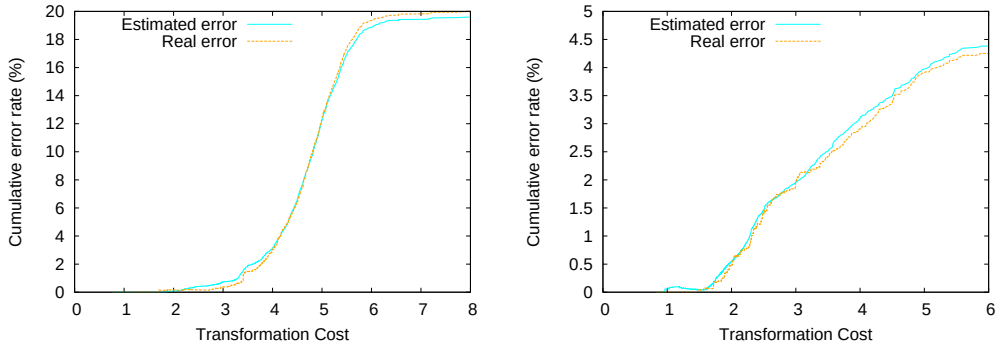

Figure 4: Example of CEC curves obtained using the proposed error estimation function E, and real error, for the sample of *Municipalities* (left) and *Surnames* (right) described in Table 1. Half of the sample was used to compute H and the other half to compute E.

as the number of strings "erroneously transcribed" having a cost up to c_i divided by the total amount of strings having a cost in the same interval (notice that it is equivalent to calculate E using $H(c, 0)$ on the sample). The small differences between both curves (error deviation) indicate that a good estimation can be achieved along the whole error range.

In some practical cases, being able to establish a limit on the rejection rate for a given set can be desirable, and estimating its corresponding error rate could be also useful. In this case, if the $(n - j)$ less reliable observations are to be rejected, the corresponding estimated error rate will be the one accumulated by accepting up to string c_j, which can be directly obtained by computing $E(c_j)$. As an extension of this feature, the traditional error-reject curve can be easily computed from the CEC curve.

Batch-adaptive rejection threshold estimation is intended to have usefulness in some practical applications, e.g., in an industrial data-input workflow, where a batch of forms or other documents must be processed for a customer, and a maximum acceptable error rate for the task is often required beforehand. The methodology proposed could be applied to any set of observations provided with a consistent confidence index.

In addition, the CEC function, computed as in Expression 2, can also be applied incrementally, since output strings $\{c_1 \dots c_i \dots\}$ can be accepted or rejected as they are produced by the system depending on whether the cumulative averaged error rate $E(c_i)$ estimated by accepting a new string c_i exceeds the target error ϵ.

6. Performance evaluation

The experiments have been designed to evaluate the accuracy of the adaptive rejection threshold estimation procedure presented. The ability of the system to approximate a pre-specified error rate (target error) on real samples having different degrees of representativity with respect to the "training set" has been characterized.

The supervised samples obtained from a real OCR workflow, as detailed in Section 3, were used in the experiments. In order to assess the performance of the proposed method on samples from the same language having different cost distributions, three different test sets were used: *Easy Test* having mostly low costs (high confident strings representing careful writing), *Hard Test* having mostly high costs, and *Total Test* which corresponds to the whole test sample.

The algorithm described in Llobet et al. (2010) has been applied to the OCR hypotheses, and their transcription costs have been obtained. Then, according to the experimental design described above, the EC and CEC functions from each language model and test samples, respectively, were computed.

For a given test sample, the results are provided in terms of error deviation, i.e., the difference between the target error rate (estimated) and the real error rate (measured). The error deviation is presented for different values of the target error. Results of the batch-adaptive rejection threshold estimation method are compared to those obtained by applying a classical fixed threshold. Details of the experiments and discussions are presented below.

6.1. Test sample composition and experimental design

In order to present statistically consistent results using the available samples, bootstrapping has been chosen as the experimental design technique. For each experiment, one hundred replications have been carried out. For each bootstrapping replication, the sample of a language was randomly split in two halves: one half used as the test set, and the other half used to compute the EC function.

To build the *Easy Test* and the *Hard Test*, the set of observations selected for testing in each replication was split: up to the 50^{th} percentile (lowest costs) and over this percentile (highest costs). The *Easy Test* was composed by randomly taking a 75% within the lowest costs plus a 25% within the highest ones, and vice-versa for *Hard Test*, while *Total Test* included all the

test observations. An example of *Easy Test* and *Hard Test* cost distributions for the language model of *Municipalities* is shown in Figure 1 (right).

6.2. Results on error rate estimation

To compute $H(c, w)$, the bins were calculated using a moving average on a centered rectangular window with a side $w = 0.25$. Thus, based on the experimental design described above, one hundred different EC functions H have been obtained for each language, one per replication, and the corresponding CEC functions E from the three *Easy Test*, *Hard Test*, and *Total Test* samples from each replication were obtained, too.

First of all, we will focus on analyzing the results obtained on *Total Test*. Figure 5 plots μ for the 100 values of error deviation for incremental targets obtained by means of adaptive and fixed thresholding techniques (dots around zero in the Y axis) from each language model. Their corresponding high and low 95%-confidence intervals –CI–, computed by means of the t-student distribution, are also shown (error bars) along with the area bounded by two standard deviations $\mu \pm 2\sigma$ (lines). Notice that, while the CI corresponds to the expected mean deviation, $\mu \pm 2\sigma$ corresponds to the deviation expected when processing a single test. The figure shows that those statistics are very similar for all the four experiments using either fixed or adaptive thresholding. Several important conclusions arise from these results:

- Given that the average deviation is very close to zero in all cases, a very good estimation of the error rate can be obtained using either fixed or adaptive thresholding. Notice that, given the experimental design detailed above, the samples used for calculating the EC curves ("training samples") are representative of the *Total Test* samples.

- The $\mu \pm 2\sigma$ intervals indicate that the accuracy of the estimations when processing a single batch is not too high. However, the CI's are very narrow, meaning that, in practice, the average error deviation obtained when processing a number of batches will be very small, even for very low target error rates.

- All the four language models tested provide very similar results despite they have different characteristics, like the size of the language, the sample error (at 0% rejection), or the use of frequencies on the positive sample.

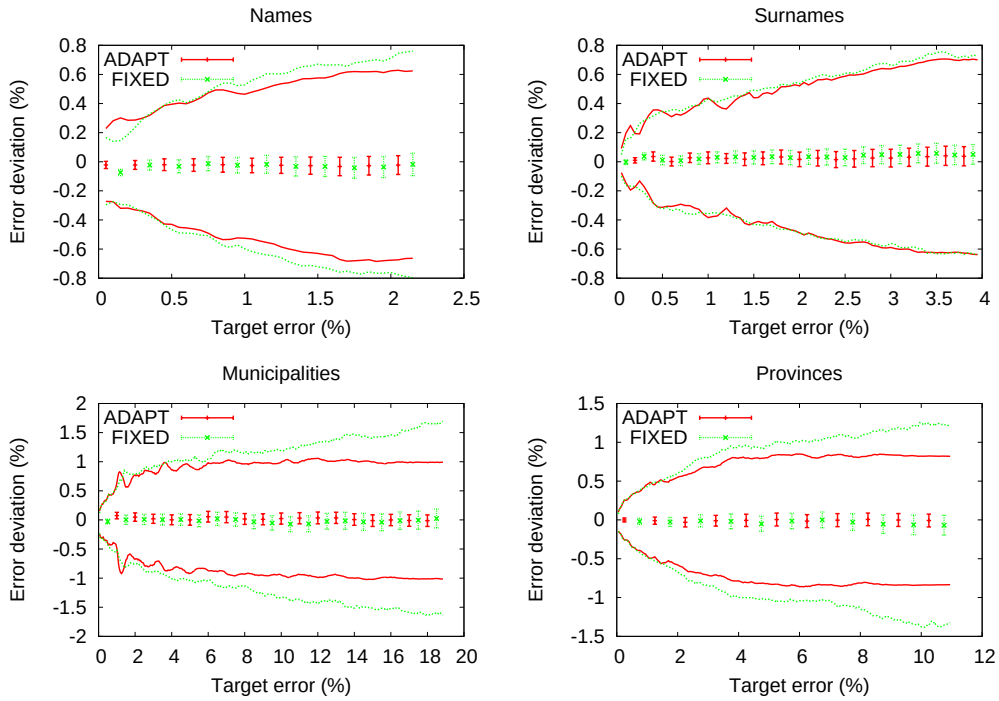

Figure 5: Average deviation on the estimation of the error rate (points around zero) resulting from the application of adaptive and fixed thresholding on the *Total Test*. The values were obtained by bootstrapping 100 replications for each language model. 95%-confidence intervals are marked with error bars and $\mu \pm 2\sigma$ are plotted with lines.

If we now study the error deviations obtained from the *Easy Test* and *Hard Test* samples using adaptive and fixed thresholding, the results, presented in Figure 6, show that the average error deviations are in a narrow range around zero for all the language models when using the adaptive thresholding method on both *Easy* and *Hard* tests. This suggest that the method would be useful in a practical application, allowing the user to specify a target error rate on different document batches very effectively for any given target error. Conversely, the classical fixed thresholding failed in all the cases: regarding the *Easy Test*, the error rates were radically overestimated which led to an unnecessary higher amount of false negatives, far from the efficient operational point, and; in the case of the *Hard Test*, the error rates were underestimated (too much false positives were accepted) and the system quality downgraded. In both cases, the error rate requirements were missed.

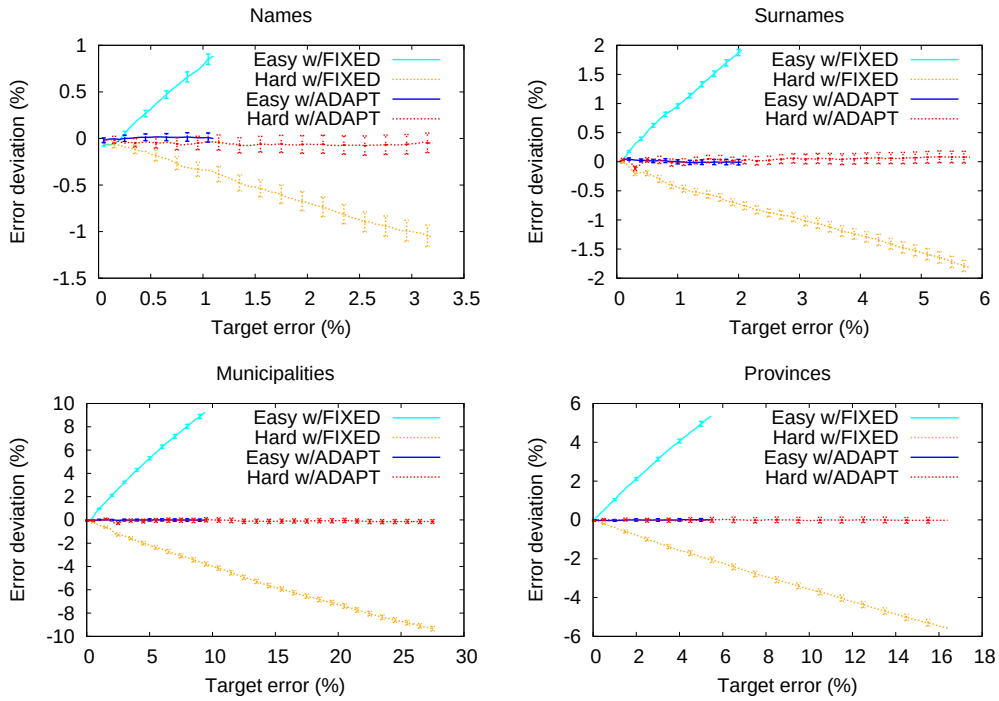

Figure 6: Average deviation (with 95%-confidence intervals) on the estimation of the error rate resulting from the application of adaptive and fixed thresholding on two samples having different cost distributions, *Easy Test* and *Hard Test*. All the error deviations are in a narrow range around zero when using adaptive thresholding while the classical fixed thresholding fails in all the cases.

These results show, on the one hand, the lack of robustness that can be attributed to fixed thresholding in error estimation of batches having different cost distributions from the sample used to estimate the threshold, and on the other hand, the validity of the presented method in those cases, giving the system the ability to adapt to document sets having different nature, in level of difficulty, during an operative workflow.

As an example, Table 2 shows a comparison on the number of rejected strings for a target error rate $\epsilon = 1\%$ on each test set using adaptive and fixed thresholding estimation, as well as the rejection corresponding to the real threshold. As a consequence of the results shown in Figure 6, the number of rejections is unnecessary higher when using fixed thresholding on the *Easy Test*. Further, the strong variability on the amount of rejected observations among samples of the same language is apparent. Controlling that

variability can be considered as an important practical issue because of its economic impact, which is successfully addressed using adaptive thresholding. Obviously, a fixed thresholding will never fulfil the system requirements on both *Easy Test* and *Hard Test*.

Table 2: Percentages of rejected strings as a result of applying adaptive, fixed and real thresholds for a 1% target error rate (averaged results). The strong variability on the amount of rejections among the three samples of the same language is successfully addressed using adaptive thresholding, while fixed thresholding causes an unnecessary higher amount of rejections on the Easy Test and an insufficient amount on the Hard Test. This behaviour is consistent for the four languages tested.

	Easy Test			*Total Test*			*Hard Test*		
	Real	Adpt.	Fixed	Real	Adpt.	Fixed	Real	Adpt.	Fixed
Names	0.80	0.80	1.93	3.99	3.96	3.91	7.88	7.48	5.89
Surnames	3.81	3.73	10.49	20.62	20.92	20.98	46.37	46.37	31.46
Municip.	14.85	14.81	17.50	35.09	35.40	35.11	59.05	58.35	52.52
Provinces	7.19	7.14	8.86	17.86	17.86	17.76	31.54	31.94	26.66

As explained in Section 5, the Error-Reject curve of a test sample can be easily computed using the proposed adaptive rejection threshold method. This can be useful in some practical cases, where requirements on limiting rejection and error rate can be combined.

7. Approach for new language models

In practice, when a new language model is defined in the system, a sample of OCR hypotheses could not be available to build the EC curve of that language. In this section, we propose to use literal strings from the positive sample of the new language to build a synthetic sample where *OCR-like errors* will be artificially introduced using a particular error-generation model. This way, the EC curve of the new language can be estimated from that synthetic sample.

An OCR error-generation model is defined as the set of probabilities of insertions and deletions of symbols, and substitutions between pairs of symbols. Using this model, a number of edit operations on the positive sample of the language can be applied to obtain a representative set of new strings having errors resembling the kind of mistakes produced in an OCR process. Obviously, the edit operation probabilities must match the errors expected,

covering all the range of errors observed in the OCR classifier being used. To take it into account, the method proposed for the generation of a synthetic sample will combine two error sources: the confusion matrix associated to the OCR classifier, and a second one conceived to account for other sort of errors, which is addressed below.

The OCR confusion matrix provides valuable information, including the expected OCR error rate (at the symbol level) and the probabilities of confusion between symbols, as well as the likelihood of insertions and deletions. In a practical setting, the OCR confusion matrix is usually available, obtained from previous tasks performed with the same OCR engine.

However, there are errors that can appear in a practical task and cannot be strictly attributed to the character recognition stage, but to variability sources of different nature, essentially unpredictable and difficult to include in the data sets used to estimate the confusion matrices, causing parts of the OCR strings to be severely affected. Some of them are due to defective image acquisition or pre-processing, e.g., distortions or translations in registration, bad character segmentation, incomplete cell removal, etc. Others are introduced by the writer, like crossing outs, mistakes, overwritten or abnormal characters, very careless or unusual writing style, as well as alternative or bad spellings (e.g., in some geographical areas, abbreviations or expansions of some words).

Strings with these type of errors usually cause high transformation costs not found in the standard OCR confusion matrices. To generate strings that reflect this variability, we propose to use an additional confusion matrix. Since the errors that we intend to model do not follow a known pattern, we used a uniform matrix built by assigning the same probability P to all the diagonal elements (substitutions of a symbol by itself), and the rest of the mass of probability, $1 - P$, uniformly distributed among the rest of the elements of the row (or column), corresponding to confusions between pairs of different symbols, including insertions and deletions, which account for some of the errors mentioned.

Figure 7 plots two examples of cost distributions from string samples obtained from each matrix, a mixture of both (synthetic), and the real sample. The curves show that, the strings generated by the uniform matrix cover a range of high transformation costs where the OCR matrix only generates a small proportion of strings, potentially contributing to a better estimation of the right tail of the EC curve.

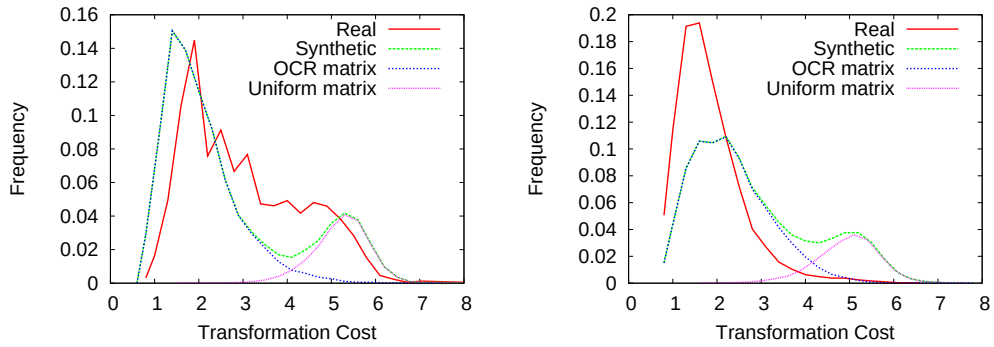

Figure 7: Transformation-cost distributions of the real and synthetic simulation samples for the *Municipalities* (left) and *Surnames* (right) language models. The "Synthetic" curve includes the strings generated from the OCR and uniform confusion matrices.

7.1. Results on error rate estimation for new language models

To validate the proposed approach, a synthetic sample size of 50,000 strings per language model was considered, and an OCR confusion matrix obtained from the OCR classifier was taken. The search for the best generation parameters of the synthetic sample was performed through optimization a) of the rate R between the amount of strings produced using the OCR confusion matrix and those produced using the uniform matrix, and b) of the mass of probability P assigned to the diagonal of the uniform matrix.

The estimation of parameters, R and P was performed by means of an orthogonal experimental design with 16 treatments for each language, considering R and P as the factors (4 levels each factor). The dependent variable was computed as the absolute error between the synthetic and real CEC curves, as shown in Expression 3.

$$J(R,P) = \sum_{i=1}^{n} \left| \widehat{E}(c_i) - E(c_i) \right| \tag{3}$$

where $\widehat{E}$ and E were calculated using the EC curves (with $w = 0.25$) from the synthetic and the *Total Test* samples, respectively, and c_n is the maximal cost between both samples (bootstrapping was applied).

By estimating a model that relates the objective function shown in Expression 3, with the independent parameters R and P, it is possible to compute the values R and P by minimizing the absolute error rate as shown in Expression 4.

$$R, P : \min J(R, P) = \min \sum_{i=1}^{n} \left| \widehat{E}(c_i) - E(c_i) \right| \tag{4}$$

The experiments were designed to allow a possible quadratic answer of factors like that of the regression model in Expression 5, that will be estimated by means of a minimum squared error algorithm.

$$J = \beta_0 + \beta_1 P + \beta_2 R + \beta_3 PR + \beta_4 P^2 + \beta_5 R^2 + \varepsilon \tag{5}$$

where ε is related with the homoscedastic residuals, that are distributed as a $N(0, \sigma)$. This σ allows us to estimate the final residual error, related with the variability explained by the rest of factors that have not been considered in the model.

As a result of the optimization process a single pair of values $R = 0.8$ (80% of strings produced by the OCR confusion matrix and 20% produced by the uniform matrix), along with a mass probability $P = 0.3$, were estimated for the whole set of languages. Finally, to assess the accuracy of the error rate estimation for new languages, those values of R and P were used to generate synthetic samples consisting of 50,000 strings per language, and their EC curves (with $w = 0.25$) were obtained. The corresponding CEC curves of the three *Easy*, *Hard*, and *Total Test* sets were computed (bootstrapping was applied), and the average error deviations obtained were those plotted in Figure 8. The standard deviation was found to be very similar to the one obtained in the experiments of Section 6.2.

Considering that no supervision information of any sample was used, the estimations obtained on *Easy Test* and *Total Test* can be considered useful in a wide range of cases. In practice, in tasks having low error rate requirements (like some form processing systems), typically acceptable field-level error rates could be between 0.5% and 3%. In that useful range, the error deviations were small enough to be usable, or at least, a good starting point in a process of stepwise refinement. Regarding the *Hard Test*, the error deviations obtained were higher and, depending on the task, they can be considered not suitable for some target ranges in some languages. In the case of less strict error rate requirements, like an OCR for a data-mining task, the system performed well with any test sample because the relative error deviation was low or very low for medium and high error rates.

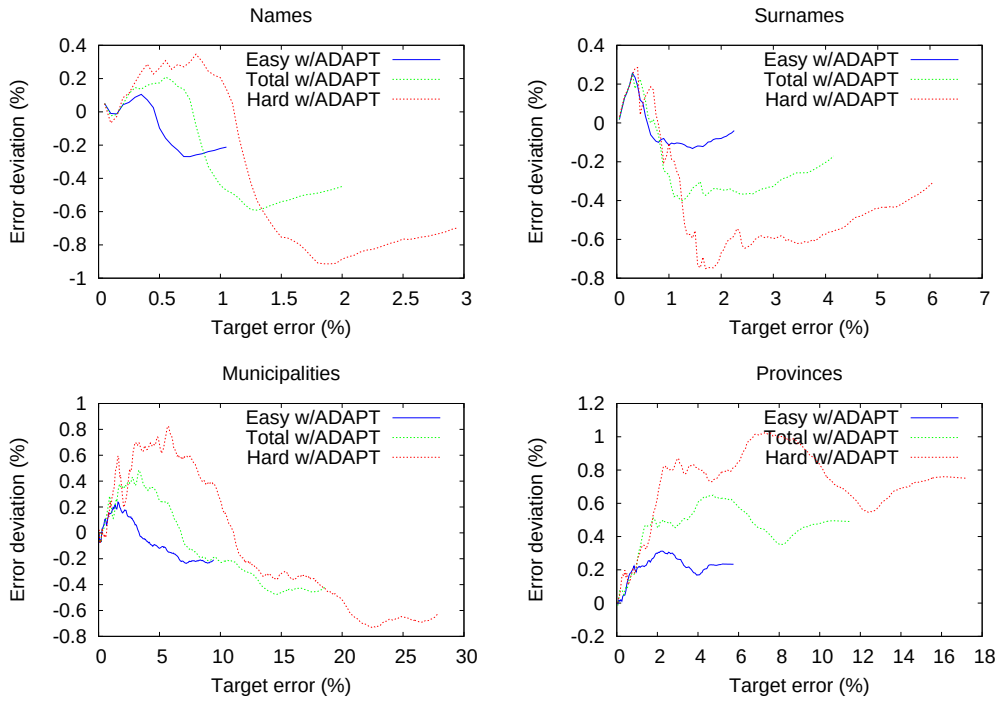

Figure 8: Difference between target and real error rates (average error deviation) computed by means of adaptive thresholding on the three samples having different cost distributions, *Easy Test*, *Hard Test*, and *Total Test* sets using synthetic samples to obtain $\widehat{H}$.

7.2. Additional considerations

Some considerations about the samples and their derived EC curves are presented and analyzed here. In Figure 9, EC curves H and $\widehat{H}$ computed from the real and synthetic samples, respectively, of all the languages studied can be compared. It is important to take into account that the estimation of $\widehat{H}$ does not require a previous character recognition process or a final manual supervision of a real sample of OCR hypotheses (necessary steps to obtain H), which is is an important advantage for some tasks, e.g. data mining. Also notice that an adaptive estimation like the one proposed becomes mandatory in this scenario, since building an arbitrary synthetic sample involves the assumption of an intrinsic error rate which can be very different from that of a (real) representative sample, rendering a fixed thresholding technique completely unusable for error rate estimation.

The estimation seems to be more accurate for the languages with real

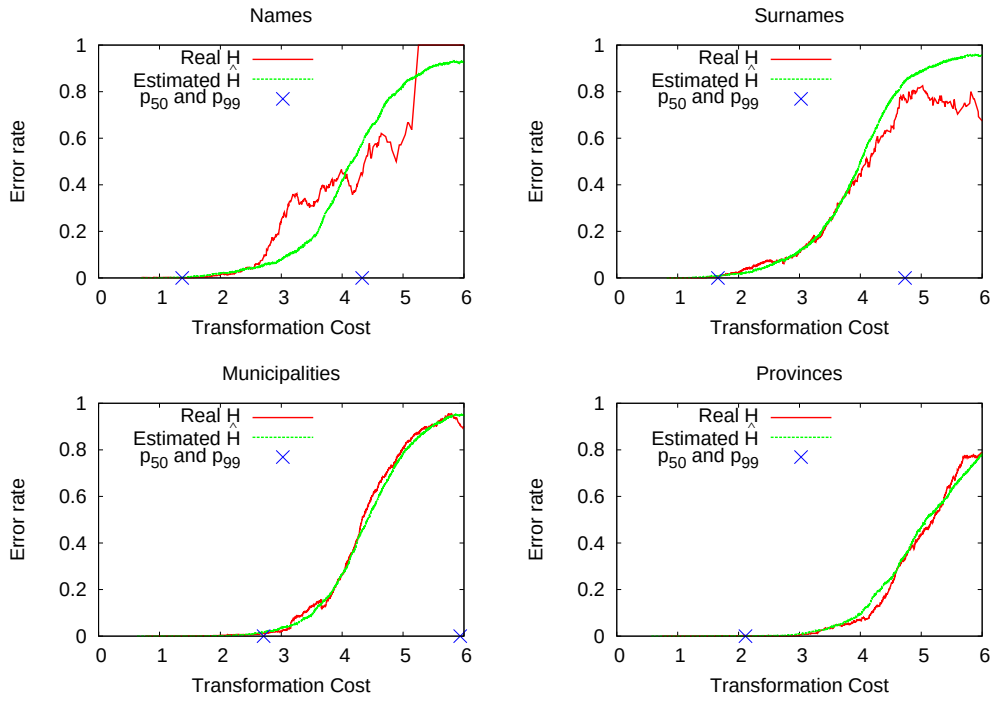

Figure 9: H and estimated $\widehat{H}$ EC curves obtained from the real and synthetic samples, respectively. The 50^{th} and 99^{th} percentiles of the cost distributions of the real sample are shown (in the *Provinces* model, the 99^{th} percentile is at Transformation Cost = 9.2).

and synthetic samples larger with respect to its language size: the ratio of sample size to language size is around 0.09 for *Names*, 0.12 for *Surnames*, 1 for *Municipalities*, and 161 for *Provinces* in the real samples (see table 1), while the amount of synthetic strings generated was fixed to 50,000 for all the language models. We think that the differences between some of the curves may not reflect a bad model estimation but a lack of representativity of the samples used, as the experimental results suggest.

The 50^{th} and 99^{th} percentiles of the cost distributions of the real samples are also indicated in the graphs. The 50^{th} percentile indicates that, in all languages, the probability of most of the OCR strings to be erroneously transcribed is very low and, therefore, any estimation considered on them will be made using a short run of the left tail of the curves. At the same time, the 99^{th} percentile indicates that estimations based on the right tail

of the curves will only apply to a very few strings in the case of *Names* and *Surnames*, which were inferred with frequencies and produced lower transformation costs and lower overall error rate than *Municipalities* and *Provinces*.

8. Conclusions

A method and experimental results for the automatic estimation of a rejection threshold on the confidence index (or transformation cost) of a sample of OCR hypotheses (batch) post-processed using a language model have been presented. The goal is that an operator should be able to set a target error rate for a whole sample instead of having to specify a threshold in an arbitrary scale.

The expert knowledge in this task comes naturally in the form of a target error rate which is negotiated between the producer and the consumer of the process (e.g. the company performing the OCR and the one that needs the resulting data for its business). This knowledge is converted into a rejection threshold that can be used in a flexible and realistic way.

A relationship between rejection threshold and error rate can be established using traditional analytic tools like Error-Reject Trade-off, Receiver Operating Characteristic (ROC), or Precision-Recall, and their different variations, wich are widely used in a diversity of Intelligent Systems. Using those tools, the error rate control of a new sample entails the assumption that a similar confidence distribution is expected with respect to the training sample used. However, in the task described here, this assumption can be unacceptable: the amount of symbol errors of an OCR classifier can widely vary for different samples depending on many factors, and consequently, the distribution of the post-processing transformation costs, can vary too. Therefore, in this case, applying a unique threshold (fixed thresholding), as the one provided by such tools, to different samples does not guarantee meeting a pre-specified error rate.

In this work, we consider the hypothesis that a probability distribution of erroneous transcriptions can be estimated from the transformation costs produced by the application of a language model (*Error vs. Cost curve* or EC curve), and that it can be used to predict the error rate of any set of strings of the language, regardless of its cost distribution. Thus, an approach for adaptive thresholding is proposed, where the EC curve of a language model is used to compute the *Cumulative Error vs. Cost curve* (or CEC curve),

which is directly used to automatically set the appropriate rejection threshold that meets the user-defined error rate on the overall sample.

We tested this approach in two practical scenarios, depending on whether a real supervised sample of OCR hypotheses of a given language is available to compute the EC curve, or not. In both cases, test sets having three different qualities of handwriting styles (easy, medium and hard) from four language models having different characteristics were tested. In the first scenario, the results show a very high accuracy in error rate estimation, with average deviations below 0.1% with respect to the real error for all the languages tested, even for high target error rates, while fixed thresholding clearly failed on the easy and hard styles from all the languages. These results are presented for all the range of possible error rates, and validate our approach against fixed threshold schemes.

In the second scenario, a synthetic sample with artificially introduced *OCR-like errors* is created by applying a new procedure, and used to estimate the EC curve. The advantage of this second scenario is that a labeled dataset is not needed, which is particularly beneficial for tasks where it is not available. In this case, building an arbitrary synthetic sample involves the assumption of an intrinsic error rate which can be very different from that of a (real) representative sample, rendering a fixed thresholding technique completely unusable for error rate estimation. The results of the application of the proposed adaptive rejection threshold in this scenario showed a potentially useful behavior in tasks having low error rate requirements (e.g., form processing), and good performances in terms of relative error deviation in case of less strict error rate requirements tasks (e.g., OCR for data mining).

Batch-adaptive rejection threshold estimation is intended for practical applications such as an industrial data-input workflow, where a batch of forms or other type of documents must be processed for a customer, and a maximum acceptable error rate for the task is often required beforehand (quality control), while keeping the amount of rejections as low as possible (production cost optimization). The methodology proposed could be applied to any set of observations provided with a consistent confidence index.

References

Amengual, J. C., & Vidal, E. (1998). Efficient error-correcting Viterbi parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, *20*, 1109–1116.

Arlandis, J., Pérez-Cortes, J. C., Navarro-Cerdan, J. R., & Llobet, R. (2010). Rejection threshold estimation for an unknown language model in an OCR task. In *Structural and Syntactic Pattern Recognition (SSPR) and Statistical Techniques in Pattern Recognition (SPR)* (pp. 738–747). Springer.

Berghel, H. L. (1987). A logical framework for the correction of spelling errors in electronic documents. *Information Processing and Management*, *23*, 477–494.

Bertolami, R., Zimmermann, M., & Bunke, H. (2006). Rejection strategies for offline handwritten text line recognition. *Pattern Recognition Letters*, *27*, 2005–2012.

Breuel, T. (1994). Language modeling for a real-world handwriting recognition task. In *AISB Workshop on Computational Linguistics for Speech and Handwriting Recognition*.

Chow, C. (1970). On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, *16*, 41–46.

Farooq, F., D., J., & V., G. (2009). Phrase-based correction model for improving handwriting recognition accuracies. *Pattern Recognition*, *42*, 3271–3277.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*, 861 – 874.

Fumera, G., Roli, F., & Giacinto, G. (2000). Reject option with multiple thresholds. *Pattern Recognition*, *33*, 2099–2101.

Garcia, P., & Vidal, E. (1990). Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Trans. on PAMI*, *12*, 920–925.

Grall-Maës, E., & Beauseroy, P. (2009). Optimal decision rule with class-selective rejection and performance constraints. *IEEE Trans. Pattern Anal. Mach. Intell.*, *31*, 2073–2082.

Hall, P. A. V., & Dowling, G. R. (1980). Approximate string matching. *ACM Comput. Surv.*, *12*, 381–402.

Hanczar, B., & Dougherty, E. R. (2008). Classification with reject option in gene expression data. *Bioinformatics (Oxford, England)*, *24*, 1889–1895.

He, C. L., Lam, L., & Suen, C. Y. (2009). A novel rejection measurement in handwritten numeral recognition based on linear discriminant analysis. In *10th International Conference on Document Analysis and Recognition* (pp. 451–455). IEEE Computer Society.

Hull, J., & Srihari, S. (1982). Experiments in text recognition with binary n-gram and Viterbi algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *4*, 520–530.

Jelinek, F. (1993). Up from trigrams, the strugle for improved language models. In *European Conference on Speech Communication and Technology, Berlin* (pp. 1037–1040).

Köksal, G., Batmaz, I., & Testik, M. C. (2011). Review: A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications*, *38*, 13448–13467.

Kolak, O., & Resnik, P. (2005). OCR post-processing for low density languages. In *Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT)* (pp. 867–874). Association for Computational Linguistics.

Landgrebe, T., Paclík, P., & Duin, R. P. W. (2006). Precision-Recall Operating Characteristic (P-ROC) curves in imprecise environments. In *International Conference on Pattern Recognition ICPR (4)* (pp. 123–127).

Li, M., & Sethi, I. K. (2006a). Confidence-based active learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, *28*, 1251–1261.

Li, M., & Sethi, I. K. (2006b). Confidence-based classifier design. *Pattern Recognition*, *39*, 1230–1240.

Llobet, R., Cerdan-Navarro, J.-R., Pérez-Cortes, J. C., & Arlandis, J. (2010). OCR post-processing using weighted finite-state transducers. In *International Conference on Pattern Recognition (ICPR)* (pp. 2021–2024).

Neuhoff, D. (1975). The Viterbi algorithm as an aid in text recognition. *IEEE Trans. on Inf. Theory*, *21*, 222–226.

Paladini, E. P. (2000). An expert system approach to quality control. *Expert Systems with Applications*, *18*, 133 – 151.

Perez-Cortes, J., Amengual, J., Arlandis, J., & Llobet, R. (2000). Stochastic error-correcting parsing for OCR post-processing. In *International Conference on Pattern Recognition ICPR-2000* (pp. 405–408). Barcelona, (Spain) volume 4.

Pitrelli, J. F., Subrahmonia, J., & Perrone, M. P. (2006). Confidence modeling for handwriting recognition: algorithms and applications. *International Journal of Document Analysis*, *8*, 35–46.

Rijsbergen, C. J. (1979). *Information retrieval*. Butterworths.

Schlapbach, A., Wettstein, F., & Bunke, H. (2008). Estimating the readability of handwritten text - a support vector regression based approach. In *19th. International Conference on Pattern Recognition* (pp. 1–4).

Serrano, N., Civera, J., Sanchis, A., & Juan, A. (2014). Effective balancing error and user effort in interactive handwriting recognition. *Pattern Recognition Letters*, *37*, 135 – 142.

Tong, X., & Evans, D. A. (1996). A statistical approach to automatic OCR error correction in context. In *Fourth Workshop on Very Large Corpora (WVLC-4)* (pp. 88–100).

Wu, H. Y., Chuang, C. L., Kung, Y. S., & Lin, R. H. (2011). Determination of optimal inspection sequence within misclassification error bound. *Expert Systems with Applications*, *38*, 5366–5372.