Neural Computing and Applications (2023) 35:17501–17520
https://doi.org/10.1007/s00521-023-08620-y

ORIGINAL ARTICLE

Lexicon-based probabilistic indexing of handwritten text images

Enrique Vidal[1] · Alejandro H. Toselli[1] · Joan Puigcerver[1,2]

Received: 12 September 2022 / Accepted: 17 April 2023 / Published online: 10 May 2023

Abstract
Keyword Spotting (KWS) is here considered as a basic technology for Probabilistic Indexing (PrIx) of large collections of handwritten text images to allow fast textual access to the contents of these collections. Under this perspective, a probabilistic framework for lexicon-based KWS in text images is presented. The presentation aims at providing formal insights which help understanding classical statements of KWS (from which PrIx borrows fundamental concepts), as well as the relative challenges entailed by these statements. The development of the proposed framework makes it clear that word recognition or classification implicitly or explicitly underlies any formulation of KWS. Moreover, it suggests that the same statistical models and training methods successfully used for handwriting text recognition can advantageously be used also for PrIx, even though PrIx does not generally require or rely on any kind of previously produced image transcripts. Experiments carried out using these approaches support the consistency and the general interest of the proposed framework. Results on three datasets traditionally used for KWS benchmarking are significantly better than those previously published for these datasets. In addition, good results are also reported on two new, larger handwritten text image datasets (Bentham and Plantas), showing the great potential of the methods proposed in this paper for indexing and textual search in large collections of untranscribed handwritten documents. Specifically, we achieved the following Average Precision values: IAMDB: 0.89, George Washington: 0.91, Parzival: 0.95, Bentham: 0.91 and Plantas: 0.92.

Keywords Pattern recognition · Posteriorgram · Relevance probability · Hidden Markov model · Recurrent neural network · Handwritten text analysis and recognition · Keyword spotting · Large-scale indexing and search

1 Introduction

Massive quantities of historical manuscripts have been converted into high-resolution images in the last decades as a result of digitalization works carried out by archives and libraries world wide. Billions of handwritten text images have been produced through these efforts, and this is only a minuscule part of the amount of handwritten documents which are still waiting to be digitalized. The aim of manuscript digitization is not only to improve preservation, but also to make the handwritten documents easily accessible to interested scholars and general public. However, access to the real wealth of these images, namely, their *textual contents*, remains elusive and there is a fast growing interest in automated methods which allow the users to search for relevant textual information contained in handwritten text images.

In order to use classical plain-text indexing and search Information Retrieval (IR) methods [1–4], a first step would be to convert the handwritten text images into digital text. But the image collections for which text indexing is highly in demand are so large that the cost of manually transcribing these images is entirely prohibitive, even by means of crowd-sourcing approaches. An obvious alternative to manual transcription is to rely on automatic *Handwritten Text Recognition* (HTR) [5–7]. However, despite the great recent advances in the field [8–10], fully automatic transcripts of the kind of historical images of interest still lack the accuracy required to enable useful

✉ Enrique Vidal
evidal@prhlt.upv.es

Alejandro H. Toselli
ahector@prhlt.upv.es

Joan Puigcerver
jpuigcerver@google.com

[1] PRHLT Research Center, Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Valencia, Spain

[2] Present Address: Zürich, Switzerland

plain-text indexing and search. Another possibility is to use computer-assisted transcription methods [11], but so far these methods cannot provide the huge human-effort reductions needed to render semiautomatic transcription of large image collections feasible [12].

HTR accuracy becomes low on real historical handwritten text images for many reasons, including unpredictable, erratic layouts, lines with uneven interline spacing and highly variable skew, etc. In addition, unambiguous reading order of layout elements is difficult or impossible to determine. Current state-of-the-art HTR systems achieve good transcription results only if perfect layout, line detection and reading order are taken for granted—as it is the case in most published results. Clearly, for moderate sized image collections, many of these problems can often be fixed by simple and inexpensive manual postprocessing, but this is completely impracticable when collections of hundreds of thousands, or millions of images are considered.

Interestingly, most or all of these problems disappear or become much less severe if, rather than to achieve accurate word-by-word image transcripts, the goal is to determine how likely is that a given word is or is not written in some indexable image region, such as a text line, a text block or paragraph. This goal statement places our textual IR problem close to the field known as *Keyword Spotting* (KWS). A comprehensive survey on KWS for text images has recently been published in [13].[1]

In recent works, we have explicitly adopted this I R point of view to develop a search and retrieval framework for untranscribed handwritten text images called *Probabilistic Indexing* (PrIx) [19–24]. Generally speaking, KWS aims at determining *locations* on a text image or image collection which are likely to contain instances of the query words, without explicitly transcribing the image(s). This is also the aim of PrIx, but rather than focusing on specific keywords, the likely locations of all the words which are deemed possible keyword candidates are simultaneously determined and indexed, along with the corresponding probabilities.

Traditional taxonomy in KWS distinguishes *Query-by-Example* (QbE) and *Query-by-String* (QbS) formulations, depending on whether query words are specified by means of example-images or just as character strings, respectively [13]. Moreover, depending on whether or not word image locations are known in advance, we have the *"segmentation-based"* and *"segmentation-free"* KWS formulations [13] respectively. Many recent works assume an intermediate view of KWS were relatively large image regions (such as lines or paragraphs), which typically contain several words, are considered the search targets where word relevance likelihoods have to be determined. This view, often referred to also as (word-)segmentation-free, and called *"line-segmentation-based"* in [13], is particularly interesting: it is very suitable to support the kind of indexing and search features needed by PrIx to provide textual access to large collections of handwritten images and, moreover, automatic image segmentation into these larger regions is generally very much less problematic than individual word segmentation. Most of the developments and results of this paper loosely adhere to this view.

The proposed PrIx framework borrows fundamental concepts from segmentation-free, lexicon-based, query-by-string KWS formulations. As we will see, HTR and PrIx can advantageously share statistical models and training methods. However, it is important to realize that HTR and PrIx are fundamentally different problems, even if both may rely on identical probability distributions and models. The HTR decision rule attempts to obtain the best sequence of words or characters (transcript) for a given image. Therefore the result epitomizes just the *mode of the distribution*; once a transcript has been obtained, the distribution itself can be safely discarded. In contrast, PrIx decisions are delayed to the query phase and, for each decision, (an approximation to) the *full distribution* is used. This obviously explains why proper KWS and PrIx can always achieve better overall search results than those provided by naive KWS based on plain HTR transcripts.

Our PrIx framework is also connected to existing works published in the Content-based Image Retrieval community. In particular [25–27] employ some type of distribution to model the semantic content of an image, expressed as keywords, and use that distribution to perform either QbS or QbE image retrieval. Our work, however, uses much richer models to represent this probability distribution (i.e., recurrent neural networks, hidden Markov models, n-gram language models, etc.), and better exploits the correlations present in human language, given that the content of the images is text itself. We also describe how to build search indexes allowing for serving fast queries, which is seldom considered in previous works.

An indexing and search system can be evaluated by measuring its *precision* and *recall* performance for a given (large) set of keywords. Precision is high if most of the retrieved results are correct while recall is high if most of the existing correct results are retrieved. In the case of naive indexing, based on automatic HTR transcripts, precision and recall are fixed numbers, which are obviously closely correlated with the accuracy of the recognized transcripts. In contrast, for a PrIx system based on the

[1] Among the works cited in this survey, it is worth noting that many recent developments are inspired in one form or the other in earlier KWS works in the field of automatic speech recognition (ASR), such as [14–18]. This is also the case of the work presented in this paper.

likelihood that a keyword is written in an image region, arbitrary precision-recall tradeoffs can be achieved by setting a threshold to decide whether the likelihood is high enough or not. We refer to this flexible search and retrieval framework as the *"precision-recall tradeoff model"*. Under this model, it becomes even more clear that proper KWS and PrIx has the opportunity of achieving better results than naive KWS based on HTR transcripts, as previously discussed.

Some of the developments and results presented in this paper are based on techniques described in [19], or follow research directions outlined in that paper. Contributions of this paper to the state of the art of handwritten image indexing and search include: First, a sound probabilistic framework is presented which helps understanding the relations between PrIx and other classical, maybe not-probabilistic statements of KWS, and provides probabilistic interpretations to many of these approaches. Second, the development of this framework makes it clear that word recognition implicitly or explicitly underlies any proper formulation of KWS, and suggests that the same statistical models and training methods successfully used for HTR can advantageously used also for PrIx. Third, experiments carried out using this approach on datasets traditionally used for KWS benchmarking yield results significantly better than those previously published for these datasets. And fourth, PrIx results on two new, larger handwritten text image datasets are reported, showing the great potential of the methods proposed in this paper for accurate indexing and textual search in large collections of handwritten documents.

The remaining sections of this paper are as follows: The problem we are interested in is formally stated in Sect. 2, which also presents the proposed general framework to be developed in the following sections. Section 3 introduces the concept of pixel-level word posteriors. While this concept is instructive, the computational costs entailed are exceedingly high. Therefore, in Sect. 4, we develop the idea of computing relevance probabilities for adequately sized, indexable image regions and explain how these probabilities can be accurately computed. In Sect. 5 we briefly review popular KWS approaches under the proposed statistical framework and discuss our specific PrIx proposal to efficiently compute accurate relevance probabilities for line-shaped image regions. The experimental settings and results are presented in Sect. 6 and Sect. 7. Finally Sect. 8 concludes the paper summarizing the work carried out and outlying future research.

2 Probabilistic indexing framework

The literature on text image KWS, outlined in Sect. 1, considers the following general question, regarding a query word v and a certain image or image region $\mathcal{X}$: *"Is v written in $\mathcal{X}$?"*

This is a "simple" yes/no question which, from a probabilistic point of view, can be properly modeled by a binary random variable. Associated with this question there is another one which might appear more complex: "What are the locations (if any) of word v within $\mathcal{X}$?". However, this can often be answered as a byproduct of solving the main question. The probabilistic framework proposed in [22] and presented here deals with these questions.

First, we need the above binary random variable which, following common notation in the IR field, will be named R (after *"relevant"*). This entails a reformulation of the original question as: "is the image $\mathcal{X}$ *relevant* for the word v?", considering that $\mathcal{X}$ is relevant for v if at least one instance of v is rendered in $\mathcal{X}$.

Second, we propose another random variable X over the set of image regions. A value of X (i.e., an arbitrary image region), will be denoted as $\mathcal{X}$. At this point we do not need to consider what are the possible sizes or shapes of image regions (a page, a paragraph, a line, a word-sized bounding box, etc.) and, until we need to be more specific, we will simply use the term *"image"* for a value of X.

Finally, we introduce the random variable, Q, over the set of all possible user queries. An arbitrary value of Q is generally denoted as q. The proposed framework properly admits arbitrary types of queries: from single words, to Boolean word combinations [28], or even "example image patches", as in QbE KWS [29]. However, to keep the presentation simple, in this paper we consider only conventional string query search, where queries are individual keywords, v, from a given vocabulary, V. Therefore, from now on, a generic value of Q will be denoted as v.

We can now introduce the *relevance probability distribution*[2]:

$$P(R = \text{yes} \mid X = \mathcal{X}, Q = v) \quad \equiv \quad P(R \mid \mathcal{X}, v) \tag{1}$$

which denotes the probability that $\mathcal{X}$ is relevant for the keyword v. The relevance probability can be obviously interpreted as the statistical expectation that v is written in $\mathcal{X}$ and, therefore, the expected number of words from V written in $\mathcal{X}$ an be simply computed as $\sum_{v \in V} P(R \mid \mathcal{X}, v)$. On the other hand, $P(R \mid \mathcal{X}, v)$ is also just the posterior

[2] To simplify notation, from now on we will generally write $P(R \cdots)$ and $P(a \cdots)$, rather than $P(R{=}\text{yes} \cdots)$ and $P(A{=}a \cdots)$, respectively, except when the full notation helps enhancing clarity and/or avoiding ambiguity.

probability underlying the following 2-class *classification* problem:

Given v, classify each $\mathcal{X}$ into one of these two classes:
- yes : v is (one of the words) written (somewhere) in $\mathcal{X}$
- not : v does not appear in $\mathcal{X}$

(2)

Using a *loss* matrix λ to weight the cost of each yes/not decision, the resulting decision theoretic Bayes' or *minimum expected risk rule* amounts to classify $\mathcal{X}$ into the class "yes" *iff* [30]:

$$P(R \mid \mathcal{X}, v) \; > \; \tau\,, \quad \tau \; = \; \frac{\lambda_{YN} - \lambda_{NN}}{\lambda_{NY} - \lambda_{YY} + \lambda_{YN} - \lambda_{NN}} \tag{3}$$

where in this two-class case, λ reduces to a single threshold τ. Under the *precision-recall tradeoff model*, this is exactly the threshold to be adjusted in order to achieve the required tradeoffs.

In the next sections we explain how to compute relevance distributions for given images. We will start in Sect. 3 by introducing the concept of *"posteriorgram"*, which represents word posterior probabilities computed at the pixel level. While such a representation is conceptually enlightening, its computation is expensive and, moreover, it would require prohibitive amounts of memory and time for keyword indexing and search. Therefore we will argue that keyword search does not really need such a fine-grained resolution and, in Sect. 4, we discuss the convenience of computing the required probabilities for whole indexable image regions of adequate size. Finally, in Sects. 4.1 and 4.2 we explain how $P(R \mid \mathcal{X}, v)$ can be accurately derived from pixel-level word posteriors when $\mathcal{X}$ is an adequate image region.

3 Pixel level keyword search: image posteriorgram

The *posteriorgram* of a text image $\mathcal{X}$ and a keyword v is the probability that v uniquely and completely appears in a bounding box containing the pixel (i, j). In mathematical notation:

$$P(Q = v \mid X = \mathcal{X}, L = (i,j)) \;\equiv\; P(v \mid \mathcal{X}, i, j)\,, \quad 1 \le i \le I, \;\; 1 \le j \le J, \;\; v \in V \tag{4}$$

where L is a random variable over the set of locations (pixel coordinates) and I, J are the horizontal and vertical dimensions of $\mathcal{X}$, respectively. $P(v \mid \mathcal{X}, i, j)$ is a proper probability distribution over the vocabulary V; that is:

$$\sum_{v \in V} P(v \mid \mathcal{X}, i, j) \; = \; 1, \quad 1 \le i \le I, \;\; 1 \le j \le J \tag{5}$$

A simple way to compute $P(v \mid \mathcal{X}, i, j)$ is by considering that v may have been written in any possible bounding box b in $\mathcal{B}(i,j)$, the set of all bounding boxes which contain the pixel (i, j):

$$\begin{aligned} P(v \mid \mathcal{X}, i, j) \; &= \sum_{b \in \mathcal{B}(i,j)} P(v, b \mid \mathcal{X}, i, j) \\ &= \sum_{b \in \mathcal{B}(i,j)} P(b \mid \mathcal{X}, i, j)\, P(v \mid \mathcal{X}, b, i, j) \end{aligned} \tag{6}$$

$P(v \mid \mathcal{X}, b, i, j)$ in Eq. (6) is the probability that v is the (unique) word written in the box b (which includes the pixel (i, j)). Therefore it is conditionally independent of (i, j) given b, and Eq. (6) simplifies to:

$$P(v \mid \mathcal{X}, i, j) \; = \; \sum_{b \in \mathcal{B}(i,j)} P(b \mid \mathcal{X}, i, j)\, P(v \mid \mathcal{X}, b) \tag{7}$$

This marginalization process is illustrated in Fig. 1; and Fig. 2 shows real results of computing $P(v \mid \mathcal{X}, i, j)$ in this way for an example image $\mathcal{X}$ and a specific keyword v.

The distribution $P(b \mid \mathcal{X}, i, j)$ of Eq. (7) should be interpreted as the probability that some word (not necessarily v) is written in the image region delimited by the bounding box b. Therefore, this probability should be high for word-shaped and word-sized bounding boxes centered around the pixel (i, j), like some of those illustrated in Fig. 1. In contrast, it should be low for boxes which are too small, too large, or are too off-center with respect to (i, j). For simplicity, it can be assumed that this distribution is *uniform for all reasonably sized and shaped boxes around (i, j)* (and null for all other boxes), and then just replace this distribution with a constant in Eq. (7). Such a simplification encourages the peaks of the posteriorgram to be rather flat, as in Fig. 2.

On the other hand, the term $P(v \mid \mathcal{X}, b)$, is exactly the posterior probability needed by any system capable of recognizing a pre-segmented word image (i.e., a sub-image of $\mathcal{X}$ bounded by b). Actually, such an isolated word

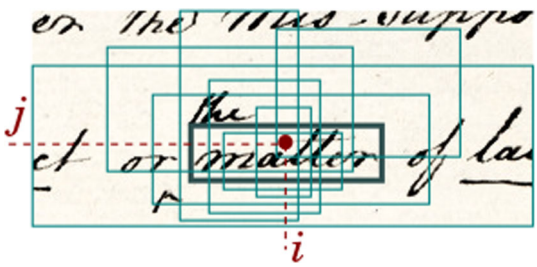

Fig. 1 Marginalization bounding boxes $b \in \mathcal{B}(i,j)$. For $v =$ "`matter`", the thick-line box will provide the highest value of $P(v \mid \mathcal{X}, b)$, while most of the other boxes will contribute only (very) low amounts to the sum

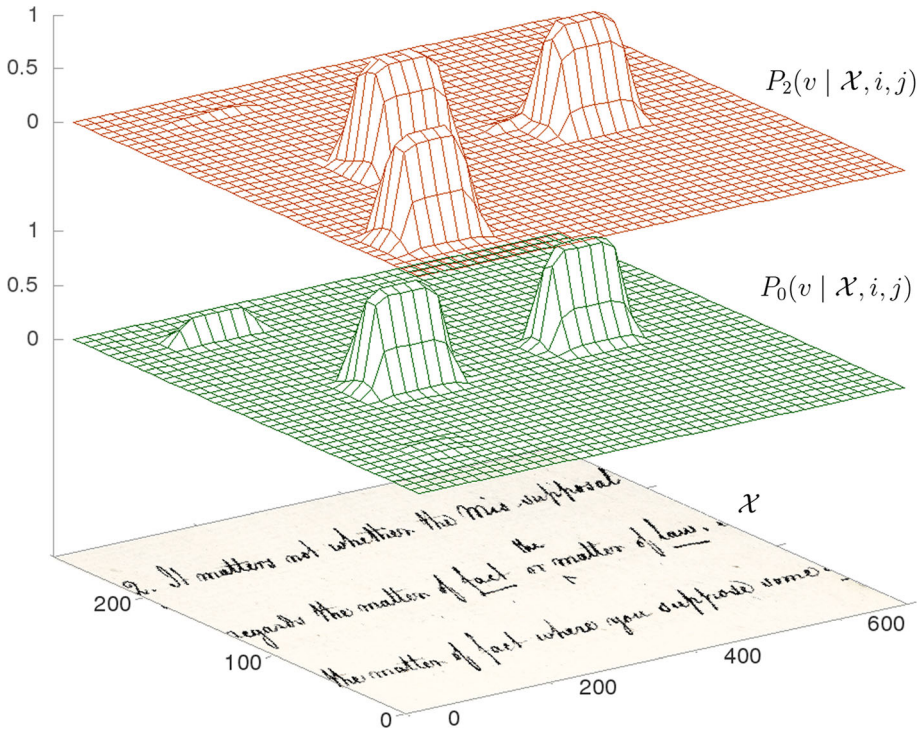

Fig. 2 Identical optical HMMs were carefully trained in order to help computing two 2-D posteriorgrams, P_0 and P_2, for a text image $\mathcal{X}$ and keyword $v =$ "matter". A context-agnostic HMM+0-gram isolated word classifier was used to obtain P_0. But much better posterior estimates are offered by P_2, obtained using a contextual, HMM+2-gram classifier

recognition task can be formally written as the following classification problem:

$$\hat{v} \;=\; \arg\max_{v \in V} \; P(v \mid \mathcal{X}, b) \tag{8}$$

In general, any system capable of recognizing pre-segmented word images implicitly or explicitly computes $P(v \mid \mathcal{X}, b)$ and can thereby be used to obtain the posteriorgram according to Eq. (7). For example, using a k-Nearest Neighbor classifier, it can be approximated just as [30]:

$$P(v \mid \mathcal{X}, b) \;=\; \frac{k_v}{k} \tag{9}$$

where k_v is the number of v-labelled prototypes out of the k which are nearest to to the image in the bounding box b of $\mathcal{X}$.

Obviously, the better the classifier, the better the corresponding posteriorgram estimates. This is illustrated in Fig. 2, which shows two examples of image posteriorgrams obtained according to Eq. (7) using two different word image recognizers. In both cases, well trained optical hidden Markov models (HMM) were used to compute $P(v \mid \mathcal{X}, b) \;\; \forall b \in \mathcal{B}(i,j)$. $P_0(v \mid \mathcal{X}, i, j)$ was obtained directly, using a plain, context-agnostic optical recognizer, and $P_2(v \mid \mathcal{X}, i, j)$ was produced using a more precise *contextual word recognizer*, additionally based on a well trained bi-gram. As it can be seen, P_0 values are only good for the two clear instances of "matter", but almost vanish for a third instance, probably because of the faint character "m". Worse still, P_0 values are relatively high for the similar, but wrong word "matters"; in fact very much higher than for the third, faint instance of the correct one. In contrast, the contextual recognizer leads to high P_2 values for all the three correct instances of "matter", even for the faint one, while the values for the wrong word are very low. Clearly bigrams such as "It matter" and "matter not" are unlikely, thereby preventing $P_2(v \mid \mathcal{X}, b)$ to be high for any box b around the word "matters". On the other hand, the bigrams "the matter" and "matter of" are very likely, thereby helping the optical recognizer to boost $P_2(v \mid \mathcal{X}, b)$ for boxes b around the faint instance of "matter".

Pixel-level posteriorgrams could be directly used for keyword search: Given a threshold $\tau \in [0, 1]$, a word $v \in V$ is spotted in all image positions where $P(v \mid \mathcal{X}, i, j) > \tau$. Varying τ, adequate *precision–recall* tradeoffs could be achieved.

4 Image regions for keyword indexing and search

Computing the full posteriorgram as in Eq. (7) for all the words of a large vocabulary (as needed for indexing purposes) and all the pixels of each page image entails a formidable amount of computation. The same can be said for the exorbitant amount of memory which would be needed to explicitly store all the resulting posterior probabilities. Therefore such a direct approach is obviously

inappropriate for indexing purposes and, moreover, it becomes unfeasible for the size of text image collections considered in this work. Clearly, rather than working at the *pixel level*, some adequately small *image regions*, x, which are indexable and suitable search targets for users, need be defined to compute the relevance probabilities introduced in Sect. 2.

While these concerns are seldom discussed in the KWS literature, *region proposal* [31] has been the focus of a number of studies in the object recognition community, as well as in the field of document analysis—see e.g., [32], which deals with graphic pattern spotting in historical documents and [33–35], which apply region proposal neural networks to various document layout analysis tasks.

In the traditional KWS literature, word-sized regions are often considered by default. This is reminiscent of segmentation-based KWS methods which required previously cropped accurate word bounding boxes. However, as discussed in Sect. 1, this is not realistic for large image collections. More importantly, by considering isolated words, it is difficult for the underlying word recognizer to take advantage of word linguistic contexts to achieve good spotting precision (as illustrated in Fig. 2).

At the other extreme we may consider whole page images, or relevant *text blocks* thereof, as the search target image regions. While this can be sufficiently adequate for many textual content retrieval applications, a page may typically contain many instances of the word searched for and, on the other hand, users generally like to get narrower responses to their queries.

A particularly interesting intermediate search target level consists of *line-shaped regions*. Lines are useful targets for indexing and search in practice and, in contrast with word-sized image regions, lines generally provide sufficient linguistic context to allow computing accurate word classification probabilities. Moreover, as will be discussed in Sect. 5.2, line region posteriorgrams can be very efficiently computed.

4.1 Image-region relevance probabilities

Let us now examine how to obtain the relevance probabilities $P(R|\mathcal{X},v)$ defined in Eq. (1) when $\mathcal{X}$ is a suitable (typically a line-shaped or any other small) *image region*. To emphasize this greater concreteness, we will write x rather than $\mathcal{X}$.

4.1.1 Naive Word Posterior Interpretation of $P(R \mid \mathbf{x}, v)$

If x were a word-sized, tight bounding box, then $P(v|\mathrm{x})$ could be used as a proxy for $P(R|\mathrm{x},v)$ as:

$$\begin{aligned} P(R=\text{yes} \mid \mathrm{x}, v) &\approx P(v|\,\mathrm{x}) \\ P(R=\text{not} \mid \mathrm{x}, v) &\approx \sum_{u \neq v} P(u \mid \mathrm{x}) \;=\; 1 - P(v \mid \mathrm{x}) \end{aligned} \tag{10}$$

As will be discussed in Sect. 5, this is in fact the approach implicitly or explicitly adopted by all word-segmentation-based KWS methods. So it is not surprising that researchers have tried to stretch this idea even if x is *not* a tight word bounding box (i.e., it may contain multiple words). In this case, however, the intuition behind the classification problem underlying $P(v \mid \mathrm{x})$ is unclear: How the (unique) "most likely word" $\hat{v} = \arg\max_v P(v|\mathrm{x})$ should be interpreted? Moreover, $P(v|\mathrm{x})$ sums up to one for all $v \in V$; but in keyword search, each word actually written in x should have high relevance probability and, as mentioned above (Sect. 2), the sum should rather approach the expected number of different words written in x.

In Sect. 7, we empirically study whether using $P(v|\mathrm{x})$ with line image regions can still provide useful KWS performance. While the posterior underlying any isolated word classifier can straightforwardly be used to obtain $P(v|\mathrm{x})$, in the comparative experiments of Sect. 7 we tried to use the same underlying probability distributions for all the methods. To this end, one can realize that $P(v|\mathrm{x})$ can be readily obtained as a simple pixel-average of the posteriorgram as follows:

$$P(v \mid \mathrm{x}) \;= \sum_{ij} P(v,i,j \mid \mathrm{x}) \;\approx\; \frac{1}{I \cdot J} \sum_{ij} P(v \mid \mathrm{x}, i, j) \tag{11}$$

where $I \cdot J$ is the number of pixels of x and, for simplicity, possible positions (i, j) of words are assumed to be equiprobable.

4.1.2 Proposed approximations to $P(R \mid \mathbf{x}, v)$

To start with, let the correct transcript of x be the sequence of words $w = w_1, w_2, \ldots, w_n$, $w_k \in V, 1 \leq k \leq n$, and let us abuse the notation and write $v \in w$ to denote that $\exists k,\ w_k = v$. The definition of the class "yes" in Eq. (2) can then be written as:

$$(R = \text{yes}) \;\equiv\; (v \in w) \;\equiv\; (w_1 = v \vee w_2 = v \ldots \vee w_n = v) \tag{12}$$

Of course, if w were known, the relevance probability $P(R \mid \mathrm{x}, v)$ would trivially be 1 if $v \in w$ and 0 otherwise. In KWS or PrIx, no transcripts are available, but an obvious, naive idea is to approximate w with a best HTR transcription hypothesis, $\hat{w}(\mathrm{x})$ (see Sect. 4.4). This results in:

$$P(R \mid \mathrm{x}, v) \;\approx\; \begin{cases} 1 & \text{if } v \in \hat{w}(\mathrm{x}) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

While the simplicity of this idea makes it really enticing (and it has in fact become quite popular), we anticipate that $\hat{w}(\mathrm{x})$ is seldom accurate enough in practice, and this method generally yields poor *precision-recall* performance.

Therefore, we propose less simple but hopefully more accurate developments. According to [36] and Eq. (12), $P(R \mid \mathrm{x}, v)$ can be exactly written as:

$$\begin{aligned} P(R \mid \mathrm{x}, v) \;=& \sum_{k=1}^{n} P(w_k = v \mid \mathrm{x}) \\ &- \sum_{l<k} P(w_k = v, w_l = v \mid \mathrm{x}) \\ &+ \sum_{m<l<k} P(w_k = v, w_l = v, w_m = v \mid \mathrm{x}) \\ &\cdots \\ &(-1)^{n-1}\; P(w_1 = v, \ldots w_n = v \mid \mathrm{x}) \end{aligned} \tag{14}$$

If the image regions are sufficiently small (e.g., line regions), it can reasonably be expected that only one instance of each keyword may appear in each region. In these cases, all the joint probabilities in Eq. (14) vanish and it simplifies to:

$$P(R \mid \mathrm{x}, v) \;\approx \sum_{1 \le k \le n} P(w_k = v \mid \mathrm{x}) \;\stackrel{\text{def}}{=} \sum_{1 \le k \le n} P_{kv\mathrm{x}} \tag{15}$$

where the terms $P_{kv\mathrm{x}} \stackrel{\text{def}}{=} P(w_k{=}v \,|\, \mathrm{x})$ have been introduced to simplify forthcoming notation. A drawback of this approximation is that $P(R|\,\mathrm{x}, v)$ may become improper since the sum can be greater than one if a keyword appears more than once in x.

To avoid this problem, rather than plainly ignoring the joint probabilities of Eq. (14), they can be approximated by naive Bayes estimates:

$$\begin{aligned} P(R \mid \mathrm{x}, v) \;\approx& \sum_{k=1}^{n} P_{kv\mathrm{x}} \\ &- \sum_{l<k} P_{kv\mathrm{x}}\, P_{lv\mathrm{x}} \\ &+ \sum_{m<l<k} P_{kv\mathrm{x}}\, P_{lv\mathrm{x}}\, P_{mv\mathrm{x}} \\ &\cdots \\ &(-1)^{n-1}\; P_{1v\mathrm{x}} \ldots P_{nv\mathrm{x}} \end{aligned} \tag{16}$$

Equation (16) can be efficiently computed by dynamic programming according to the following recurrence relation, which can be proved by simple induction:

$$P(R \mid \mathrm{x}, v) \;\approx\; q(n)\,, \quad \text{where}$$
$$q(k) = \begin{cases} P_{1v\mathrm{x}} & \text{if } k = 1 \\ P_{kv\mathrm{x}} + q(k-1)\,(1 - P_{kv\mathrm{x}}) & \text{if } k > 1 \end{cases} \tag{17}$$

Finally, inspired by the Fréchet's bounds [37], another approximation to Eq. (14) is proposed which does not suffer from the problem of Eq. (15) and, moreover, is much simpler than Eq. (16)/(17).

$$P(R \mid \mathrm{x}, v) \;\approx \max_{1 \le k \le n} P_{kv\mathrm{x}} \tag{18}$$

This approximation is intuitively appealing (see Figs. 2 and 3 for illustrations) and, as will be seen below, leads to the simplest and most effective method to obtain image region relevance probabilities.

4.2 Estimating image-region relevance probabilities from posteriorgrams

In PrIx or in KWS no transcript of x is available, but $P_{kv\mathrm{x}} \equiv P(w_k = v \mid \mathrm{x})$ can be estimated from the posteriorgram for $k \in \{1, 2, \ldots\}$. To this end, we can divide the whole region x, into n (maybe slightly overlapping or disjoint) subregions or blocks, $B_1, \ldots B_k, \ldots B_n$, where a sufficiently high and wide (usually rather flat) local maximum of $P(v \mid \mathrm{x}, i, j)$ is observed for some $v \in V$ (see Fig. 2, where n should be around 25, the number of likely words in x; or more concretely, the unidimensional illustration of Fig. 3, where n would be 9 or 10). Then:

$$P_{kv\mathrm{x}} \;\approx \max_{(i,j) \in B_k} P(v \mid \mathrm{x}, i, j) \tag{19}$$

This estimate can be used in any of the approximations given by Eqs. (15,16)/(17, 18) discussed in Sect. 4.1. Since only approximate local maxima of $P(v|\,\mathrm{x}, i, j)$ are required, simple maximum detection techniques can be used. However, the performance of the resulting relevance probabilities will depend on one or two adjustable parameters (n and/or an adequate threshold) which need to be optimized using validation data.

The case of Eq. (18) deserves special attention. In this case, a closed-form expression for the relevance probability distribution can be straightforwardly derived:

$$P(R \mid \mathrm{x}, v) \;\approx \max_{\substack{1 \le k \le n \\ (i,j) \in B_k}} P(v \mid \mathrm{x}, i, j) \;=\; \max_{i,j} P(v \mid \mathrm{x}, i, j) \tag{20}$$

Interestingly, the number of subregions, n, and the subregions themselves, needed in this derivation, finally become irrelevant and no extra adjustable parameters are needed. On the other hand, since the maximization of Eq. (20) can be carried out during the process of computing $P(v|\,\mathrm{x}, i, j)$ itself, it does not add any computational cost to this

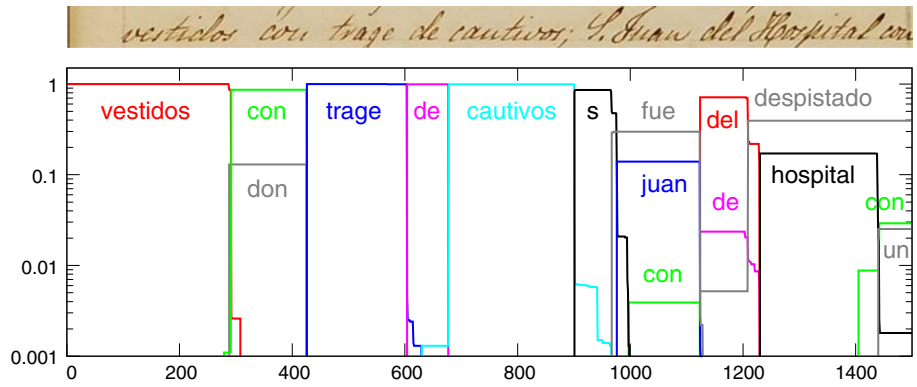

Fig. 3 A 1-D posteriorgram obtained using a HMM+2-gram contextual recognizer, following the approach outlined in Sect. 5.2

process. Moreover, the precise locations of the spotted words within spotted regions are trivially obtained as a byproduct of the maximization in Eq. (20).

One may argue that this approximation may be exceedingly rough. This issue has been empirically studied in [38] for the BENTHAM dataset described in Sect. 6.2. The results show that Eq. (20) generally yields fairly precise approximations and, moreover, the results are practically *exact* for more than 99.5% of the (line) regions and words spotted.

The derivation of Eq. (20) from Eqs. (14), (18) and (19) provides a formal explanation to a similar *max heuristic* which has been successfully used for confidence estimation in several works of automatic speech and handwritten text recognition [39, 40], and recently also for keyword indexing and search in [19].

The relative performance of all the proposed approximations will be empirically studied in Sect. 7.1. We anticipate that a best choice will be Eq. (20), based on Eq. (18).

4.3 Line-region keyword indexing and search

At the beginning of Sect. 4, line image regions were suggested as particularly adequate targets for keyword indexing and search. In connection with the discussion in Sect. 3, the following two main advantages of these regions can be identified:

1. Line regions provide *rich linguistic context* which allows computing precise word classification probabilities.
2. Line region posteriorgrams can be very *efficiently computed* by smart *choices* of the sets of relevant *marginalization boxes*, $\mathcal{B}(i,j)$, and wise *vertical subsampling*.

For a line-shaped region, the relevant sets of *marginalization boxes* needed to compute the posteriorgram according to Eq. (6) can be defined just by *horizontal segmentation*. As discussed later, these sets can be accurately and efficiently obtained as a byproduct of using a holistic, segmentation-free, context-aware handwritten recognizer on the whole line image region.

Vertical subsampling, on the other hand, can be made by guessing a line height and running, with some overlap, a vertical-sliding rectangular window of this height, as in [41] (Fig. 2 was made in this way, with large overlap to achieve high vertical resolution). However, in many cases, line detection and segmentation techniques [34, 35, 42–45] can be accurate enough to allow for computation reductions and maybe improved precision. The possible lack of robustness of this approach can be alleviated by means of *over-segmentation* [46].

In what follows we regard (as in [19, 47–52], among others—see also [13]) line-shaped image regions as our target resolution level for PrIx keyword indexing and search. As discussed in Sect. 4.2, once a line spot is determined, the exact location(s) of the keyword searched for within the line can also be easily obtained.

We assume that each handwritten page image has undergone basic preprocessing steps including correction of overall page skew and other simple geometrical distortions [11]. We do *not* need to assume that preprocessing includes any kind of character or word segmentation. Line segmentation is not essentially needed either [41] but, as discussed above, for effectiveness, efficiency and simplicity, text can be assumed to be organized into distinguishable, roughly horizontal lines.

4.3.1 1-D posteriorgram and line-region relevance probabilities

A line image region x can be processed *"frame-wise"* by extracting m narrow vertical boxes (or "frames") at uniformly spaced horizontal positions. This way, x can be seen as a *sequence* of m frames, effectively reducing x to a 1-dimensional object. The corresponding posteriorgram is also 1-D: $P(v \mid \mathrm{x}, i)$, $1 \le i \le m$, and Eq. (7) becomes:

$$P(v \mid \mathrm{x}, i) = \sum_{s \in \mathcal{S}(i)} P(s \mid \mathrm{x}, i)\, P(v \mid \mathrm{x}, s) \tag{21}$$

where $\mathcal{S}(i)$ is the set of *reasonably shifted and sized segments which contain the frame* i and, as in Eq. (7), $P(s \mid \mathrm{x}, i)$ can be assumed uniform and replaced by an adequate constant.

A fairly complete real example of this kind of context-aware line image posteriorgram is shown in Fig. 3. As discussed previously, an important advantage of line-level processing is that it allows to easily take into account the rich context provided by words surrounding each query word.

It is straightforward to rewrite the derivations and discussions of Sect. 4.2 to obtain line-region relevance probabilities from $P(v \mid \mathrm{x}, i)$. In particular, Eq. (20) becomes:

$$P(R \mid \mathrm{x}, v) \approx \max_i P(v \mid \mathrm{x}, i) \tag{22}$$

4.4 Keyword spotting and handwritten text recognition

Many authors in the field of KWS consider that KWS and HTR are different problems which require distinct methods. Aiming to shed light on this debate, in this subsection KWS is re-visited from the HTR point of view.

In Sect. 2, it was pointed out that KWS essentially boils down to answering the question: "Is the word v written in the text image region x?". As discussed in Sect. 4.1, a direct answer to this question would be to check whether v appears in a word sequence w which constitutes the transcript of x. But, since w is unknown, here we consider it as the value of a new random variable, W, defined over all the possible transcripts of x. This allows us to obtain the KWS relevance probability by marginalization on W:

$$\begin{aligned} P(R \mid \mathrm{x}, v) &= \sum_w P(R, W = w \mid \mathrm{x}, v) \equiv \sum_w P(R, w \mid \mathrm{x}, v) \\ &= \sum_w P(R \mid w, \mathrm{x}, v)\, P(w \mid \mathrm{x}, v) \end{aligned} \tag{23}$$

where w ranges over the set of all sequences of words in V. Now, since w is given in $P(R \mid w, \mathrm{x}, v)$, this probability can take only two values: 1 if $v \in w$, or 0 otherwise. On the other hand, since image transcripts (w) are independent form user queries (v), the term $P(w \mid \mathrm{x}, v)$ in Eq. (23) simplifies to $P(w \mid \mathrm{x})$ and we can write:

$$P(R \mid \mathrm{x}, v) = \sum_{w : v \in w} P(w \mid \mathrm{x}) \tag{24}$$

That is, KWS relevance probabilities can be properly computed on the base of the probability that a sequence of words w is the transcript of the image region x. Interestingly, this is exactly the same distribution used by modern HTR systems, which provide a most likely transcript of the given text image region x according to the minimum Bayes error criterion [30]; that is:

$$\hat{w} = \arg\max_w P(w \mid \mathrm{x}) \tag{25}$$

Note that the sum of Eq. (24) requires considering multiple HTR decoding hypotheses, not just the best one defined by Eq. (25). Each of these hypotheses entails a corresponding word segmentation hypothesis, which implicitly provides the marginalization boxes (segments in this 1-D case) commented in Sect. 4.3. See details in Sect. 5.2 and [19, 53, 54].

As compared with the approximations proposed in Sects. 4.1–4.2, Eq. (24) can be used to obtain *"exact"* relevance probabilities. However, the sum in Eq. (24) constitutes a complex computational problem. It can be solved by means of a dynamic programming technique similar to the "forward" approach proposed in [55]. But, even using this technique, the computational cost is still exceedingly high [54]. In this paper, we will omit these computational details, even though this "exact" approach will be used as a reference in the comparative results of Sect. 7.1.

5 Interpretation of other KWS methods and details of the proposed approaches for PRIx

In the framework proposed in Sect. 3, a KWS method is assumed to implicitly or explicitly visit all the image locations (i, j) and all the possible bounding boxes (BB) b containing (i, j) (or adequately selected subsets of locations and BBs thereof). For each (i, j) and b, an (isolated) word recognizer or word-matching technique of some kind is used to estimate the posterior probability, $P(v \mid \mathcal{X}, b)$, that a given keyword v is the (only) word written in b. Then Eq. (7) is somehow used to compute the pixel-level posteriorgram, from which region relevance probabilities are computed as explained in Sects. 4.1 and 4.2.

A single b encompasses $O(I \cdot J)$ pixels. Thus, the computing cost of estimating $P(v \mid \mathcal{X}, b)$ for all $v \in V$ is at least $\Omega(N\,I\,J)$, where N is the number of keywords. In general, for each image location there are $O((I\,J)^2)$ possible b's, and the number of locations is $O(I\,J)$. Therefore, the overall computational complexity of *directly* computing a posteriorgram in this way, for all the $I \cdot J$ pixels in a full image, is really huge: at least $\Omega(N\,(I\,J)^4)$.

It is then no surprise that the history of development of KWS for text images can be interpreted in terms of how to

deal with the different components of this exorbitant computational cost.

5.1 Interpretation of KWS methods

According to the framework introduced in this paper, three main aspects can be identified which characterize most (QbS) KWS methods for (handwritten) text images proposed so far.

- How to effectively sample the exceedingly large number of image pixel locations.
- How to select a sufficiently effective set $\mathcal{B}(i,j)$ of marginalization BBs and how to deal with the summation required in Eq. (7)
- How to estimate the word classification posterior $P(v|\mathcal{X},b)$ for each $b \in \mathcal{B}(i,j)$.

We start discussing the first two aspects, which are closely inter-related and together aim to deal with the computational costs which essentially depend on the image size, $I \cdot J$.

All of the word-segmentation-based KWS techniques [13] circumvent the high computing cost of Eq. (7) by reducing the summation to just one fixed word-sized BB. Moreover, KWS "scores" (proxy for word posteriors, see below) are computed only at the relatively very small number, l, of previously given locations of these word BBs. Obviously, by naively assuming perfect word image detection *and* segmentation, the computational cost is dramatically reduced down to $O(N\,l)$, which clearly explains the mighty popularity of this simplistic idea.

Some works, such as [56], rely on automatic over-segmentation of the text images to mitigate the impact of word segmentation errors. Such techniques rely on a richer, more realistic subsampling and, to some extent, go toward approximating the marginalization in Eqs. (6) and (7).

In fully segmentation-free KWS methods [13], subsampling generally performed through a *sliding-window* sweep over the image—see, e.g., [57]. However, full pixel-by-pixel sweep is again much too expensive and, in many works, an adequately small number p of *key-points* which define possible elements of the objects of interest (words), are previously located [13, 57]. This way, assuming marginalization is simplistically reduced to just one candidate BB or "patch" (which is usually the case) computational cost can be reduced down to $O(N\,p)$ where, in general, $p >> l$.

On the other hand, in KWS approaches which work with (word-unsegmented) *line* image regions, the summation in Eq. (7) becomes unidimensional (i.e., Eq. (21)). In many of these approaches this sum is more or less explicitly approximated only by the dominating addend (which is typically a good approximation – generally much better than relying on a single, given BB). Then, *dynamic programming* techniques are used to avoid repeated computations during a sliding window process over the horizontal positions of x. This is specifically the case of word-segmentation-free *dynamic time warping* KWS methods such as [47, 48], as well as all the modern techniques based on HMMs [49, 51, 52] and recurrent neural networks [50].

Nevertheless, obtaining a full 1-D posteriorgram for each of the M line-regions in an image would still entail high computational cost. The size of the set of marginalization segments, $S(i)$, is $O(I^2)$ and segment lengths are $O(I)$. Therefore, even if repeated computations are avoided, the overall asymptotic time complexity is $O(NMI^2)$. Fortunately, in this simpler 1-D case, reasonably good and computationally cheaper approximations can be obtained in a variety of ways. In Sect. 5.2 we describe the approach we propose to deal with this computational complexity.

Let us now discuss the last aspect which characterize a KWS method; namely how to estimate the word classification posteriors $P(v|\mathcal{X},b)$. Three main approaches can be identified: *distance-based*, *HMM*s and (recurrent) neural networks (*RNN*).

Many early approaches to KWS, notably segmentation-based ones, are *based on distances* between vector representations of queries and images. Most distance-based methods are QbE [13], but some recent QbS proposals such as [58] are also distance-based. It is well known that distances can be used to approximate probability distributions in several ways [30]. If y and z are, respectively, representations of a query word v and an image BB $(\mathcal{X},b)$, then a simple estimator of the classification posterior required in Eq. (7) is: $P(v \mid \mathcal{X},b) \approx \phi(y,z) \,/\, \sum_u \phi(u,z)$, where $\phi(u,z) = \exp(-d(u,z))$, $d(\cdot,\cdot)$ is the distance, and u ranges over (an adequate set of) query word representations. Distance-based KWS methods [13] generally drop the denominator (sum in u) and use just unnormalized distance-based "scores". While the resulting lack of basic probabilistic properties may not change the *individual* average precision of each query (see Sect. 6.1), unnormalized scores may severely hinder the global average precision for a large *set of queries*.

Consider now HMMs. A word v is commonly modeled as a concatenation of character HMMs, which estimate the likelihood $P(\mathcal{X},b|v)$ that v is rendered in the BB $(\mathcal{X},b)$. This is proportional to $P(v|\mathcal{X},b)$ assuming $P(v)$ and $P(\mathcal{X},b)$ are uniform. But improper normalization leads to similar problems as in distance-based methods and some heuristic form of word-length normalization is often required, as it is typically the case with the popular *"filler"* models [51] (see more details in [55]).

Let us finally focus on RNN [50]. For a given image BB $(\mathcal{X},b)$, these networks directly provide a sequence of

posterior probabilities $P(c \mid \mathcal{X}, b, i)$ where c is a character and i is a horizontal position within b. For a keyword v, composed of characters $c_1, \ldots, c_K$, dynamic programming can be used to obtain a best matching path $\Phi(\cdot)$, which assign each position i to one of the K characters of v. Then usual independence assumptions lead to the naive Bayes approximation: $P(v \mid \mathcal{X}, b) \approx \prod_i P(c_{\Phi(i)} \mid \mathcal{X}, b, i)$, where i ranges over the horizontal positions of b.

5.2 Proposed approaches for PrIx

To finish the formal part of this paper, we present details of the specific approaches we propose to efficiently compute accurate, context-aware line-region posteriorgrams, and the corresponding relevance probabilities. They formally follow the statistical framework developed in the preceding sections and, as previously mentioned, rely on techniques introduced in [19]. The main idea is to use a *word lattice* or *graph* (WG) [19, 59], obtained as a byproduct of solving Eq. (25) of Sect. 4.4 [11, 19]. A WG of a (line) image region, x, is a very compact representation of a huge amount of alternative image transcription results, including the probability of each of the (millions of) hypothesized words and the corresponding word segmentation boundaries.

A posteriorgram $P(v \mid \mathrm{x}, i)$ can be obtained from a WG of x following essentially the same arguments as in Eq. (15) or, more specifically, its 1-D version, Eq. (21). The basic idea is to consider that, for each position i, the "relevant, reasonably shifted and sized" segments in $\mathcal{S}(i)$ are those given by the multiple word segmentation hypotheses associated with all the WG edges, e, labeled with the word v and such that i is included within the segmentation boundaries specified by the departing and ending nodes of e. See [19] for more details. These word boundaries are generally very accurate, not only for the words in the best hypothesis of the WG (called the "1-best" transcript), but also for most of the edges associated with high-probability paths of the WG. Therefore, these boundaries and probabilities provide highly informative data to allow very accurate computation of Eq. (21).

The four aspects which characterize a KWS method are now briefly outlined for the proposed approaches: To cope with the exceedingly large number of pixel locations in a page image, it is first sampled vertically by adopting line image regions as discussed in Sect. 4.3, and horizontally according to the segmentation boundaries included in each line region WG. Similarly, we let the word segments represented in each WG define the sets of marginalization BBs. The word classification posteriors $P(v \mid \mathcal{X}, b)$ are obtained from the word likelihoods associated to the WG edges (which were computed essentially as discussed above for HMMs or RNNs). As explained in detail in [19], a specific normalization process is applied to the WG so as to convert edge likelihoods into the so-called *edge posteriors*, which are then directly interpreted as values of relevance probability (RP). Finally, with all these data we efficiently compute the sum of Eq. (21). Figure 4 shows the key components and methods explained above and illustrates the proposed PrIx generation workflow.

For a line image region of length I, the computational cost of obtaining a posteriorgram in this way is in $\Theta(\kappa I)$, where κ depends on the size of the WG [19]. Given the posteriorgram, keyword indexing relevance probabilities are cheaply computed by any of the 1-dimensional versions of Eqs. (17)–(20) (with no extra costs in the case of Eq. (22)). So, once the WGs of the M extracted line image regions are available, the overall computational effort per page image is $O(N M \kappa I)$. According to Toselli et al. [19, 60], this cost is generally dominated the cost of producing the WGs themselves, which is also basically linear with N, M and I, but grows much faster with κ. See [19, 60] for more details, including real computing times of WG generation.

6 Experimental framework

The experimental setup adopted to assess the KWS performance of the proposed approaches is presented here, including: evaluation measures, benchmark datasets, query sets, and empirical settings adopted for RNN and HMM optical modeling and for the different methods used to compute image-region relevance probabilities.

6.1 Evaluation measures

Let $\mathcal{L}$ be the set of (line) image regions and $\mathcal{Q}$ the set of queries. Let $\mathcal{E} = \mathcal{L} \times \mathcal{Q}$ be the set of *"events"*. According to Sect. 2, an event, $(\mathrm{x}, q) \in \mathcal{E}$, is relevant if and only if q is truly written in x. Let r be the total number of *relevant* events in $\mathcal{E}$ and, for a given threshold, τ, let $d(\tau)$ and $h(\tau)$ be the number of relevant events *detected* (retrieved) by the system and the number of correctly detected events (also called *"hits"*), respectively. The *Recall*, $\rho(\tau)$, and the raw (non-interpolated) *precision*, $\pi'(\tau)$, are defined as:

$$\rho(\tau) = \frac{h(\tau)}{r}, \quad \pi'(\tau) = \frac{h(\tau)}{d(\tau)} \tag{26}$$

The interrelated trade-off between recall and precision can be conveniently displayed as the so-called *recall-precision* (R-P) curve, $\pi'(\rho)$ [61]. Raw precision can be ill-defined and, moreover, raw R-P curves can present an undesired saw-tooth shape [1]. Both of these issues, which may lead to counter-intuitive performance results, can be avoided by

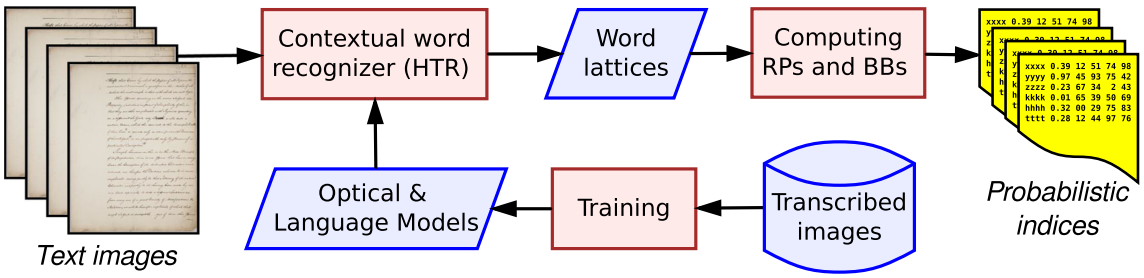

Fig. 4 PrIx workflow based on word lattices obtained using HTR optical and language models

the so-called *interpolated precision*, π, which at a certain recall level ρ is defined as:

$$\pi(\rho) = \max_{\rho':\rho' \geq \rho} \pi'(\rho') \tag{27}$$

Intuitive arguments in favor of $\pi(\rho)$, which is generally adopted in the literature, are discussed in [1]. Good search and retrieval systems should achieve both high precision and high recall for a wide range of values of τ. A commonly accepted scalar measure which meets this intuition is the area under the (interpolated) R-P curve, $\pi(\rho)$, referred to as (interpolated) *average precision* (AP) [62, 63].

Interpolated precision becomes even more necessary for fair evaluation of the naive 1-best KWS approach (Sect. 4.1, Eq. (13)), where relevance probabilities are 1 or 0, independently of τ. In this case, only one R-P point, (ρ_0, π_0), is defined in the raw R-P curve, leading to a *null* raw AP which prevents comparison with other KWS approaches. In contrast, the interpolated precision curve becomes: $\pi(\rho) = \pi_0$ if $0 \leq \rho \leq \rho_0$, $\pi(\rho) = 0$ otherwise; and the resulting interpolated AP is: $\pi_0 \cdot \rho_0$.

6.2 Datasets

The main experiments were conducted on two relatively large historical handwriting datasets, called BENTHAM [64].[3] and PLANTAS [12]. Examples of images from these collections can be seen in the web sites of the demonstrators discussed in Sect. 8[4]. In addition, comparative experiments were carried out on smaller, more usual benchmarking datasets; namely, IAM, PARZIVAL (PAR) and GEORGE WASHINGTON (GW). Main dataset features are summarized in Table 1 and full details can be consulted in [38].

The standard fourfold cross-validation scheme adopted by most other authors for GW (which is a very small set) was also adopted here and the figures in Table 1 are averages over the 4 folds. In all the cases, ground-truth line segmentation was available, both for training and testing images, and it was used in the experiments. We leave for future works to experiment with the vertical subsampling and over-segmentation ideas discussed in Sect. 4.3. All the datasets are publicly available and the details needed to produce results comparable with those reported in this paper appear in the corresponding dataset repositories[5].

6.3 Query sets

Several criteria can be assumed to select the keywords to be used in KWS assessment experiments. Clearly, a KWS system may perform better or worse depending on the query words it is tested with and how these words are distributed on the test set. Of course, the larger the set of keywords, the more reliable the empirical results. Since our approach is aimed at indexing applications, testing with large keyword sets is mandatory.

According to these observations, in this work all the words that appear in the training partition of each dataset are considered keywords. For the benchmark datasets (IAM, GW and PAR) this allows us to produce results which are exactly comparable with those of the best approaches published so far. Also for this reason, we use the query set provided by Dr. Frinken and used in previous works [19], which contains the most frequent IAM training words excluding punctuation marks and *stop words* (around 3.4K words). Finally, about one hundred words corresponding to numbers were not included in the query set for PLANTAS. Table 1 shows the sizes of the query sets used in each of the five datasets considered.

It is important to remark that, in contrast with other keyword selection criteria which adopt only test-set words for queries, here many of the keywords do not actually

[3] It is exactly the same dataset used in the ICFHR-2014 HTRtS competition [64], but differs from other datasets, also based on Bentham's manuscripts, used in the ICFHR'14 and ICDAR'15 KWS competitions.

[4] BENTHAM: http://transcriptorium.dsic.upv.es/demots/kws/index.php/ui/chapters/bentham and PLANTAS: http://transcriptorium.dsic.upv.es/demots/kws/index.php/ui/chapters/plantas.

[5] BENTHAM: https://zenodo.org/record/44519#.YxCuDEhBzZ8 PLANTAS: https://zenodo.org/record/6608342#.YxCu-0hBzZ9 IAM, -PAR, GW: http://www.fki.inf.unibe.ch/databases.

Table 1 Main datasets features

	BENTHAM	PLANTAS	IAM	PAR	GW
Hands	Several	One	Many	Three	Two
Words/line	9.3	10.1	8.8	6.3	7.5
Training + valid. running words	99.0	80.0	62.4†	19.7	3.7
Test running words	7.9	117.0	8.3	8.4	1.2
Training lexicon	8.7	11.0	7.8†	3.2	0.9
Keywords	8.7	10.9	3.4	3.2	0.9
Relevant	4.9	1.5	1.1	1.2	0.2

All numbers are thousands of words, except for "Words/line"

†Additional text-only data used for language modeling: 3.1 million running words, with a lexicon of approx. 19 K words

appear in any of the test images. We say that these keywords are *non-relevant*, while the remaining ones are *relevant*. The amounts of relevant keywords are also shown in Table 1. Trying to spot non-relevant words is challenging since, depending on the system accuracy, similar relevant words may be erroneously spotted, thereby leading to important precision degradations.

6.4 Experimental setup

The PrIx (or KWS) approaches to be assessed require statistical optical character models and language models which must be trained from the available training images and transcripts. For language modeling, the simple and time-honored state-of-the-art *n*-gram approach [65] was adopted in all the cases. But, for optical modeling, two alternative approaches were considered: HMMs, which is a well understood, proper statistical approach [65], and RNNs, which are recently showing superb performance in HTR [8, 10, 66]. Both methods have been used in the main experiments (BENTHAM and PLANTAS), while only RNNs have been used in the comparative experiments with IAM, PARZIVAL and GW.

In general, a similar system architecture was used in all the experiments. However, depending on the dataset and the optical modeling (HMM or RNN), some details of image pre-processing and feature extraction were different. The values of all training (or testing) meta-parameter which needed to be tuned were optimized using the validation set of each dataset. The general architectures and the specific details are discussed below.

6.4.1 Hidden Markov optical modeling

For HMM optical modeling, line images were preprocessed for slant, slope and size normalization [11] and then represented as sequences of feature vectors. Feature extraction (for BENTHAM and PLANTAS) was based on geometric moments normalization [67]. HMM training was carried out with the *embedded Baum Welch* algorithm [65], using all the training line images and their corresponding transcripts.

A left-to-right HMM was used for each character. The number of states and Gaussian densities per state where roughly set up taking into account the average characters' width and other dataset features, and finally optimized using validation data. More details about meta-parameter settings are given in [38, 68].

6.4.2 Recurrent neural network optical modeling

The RNN architecture adopted, shown in Fig. 5, was essentially the same as that introduced in [10]. It consists of a stack of 2D convolutional blocks (including *LeakyReLU* activations and *max pooling* layers), followed by a stack of *bidirectional long short-term memory* (BLSTM) recurrent layers [8, 69], which process the result of the previous blocks column-wise. Finally, each column is linearly transformed to have as many features as characters are in the particular dataset, plus an additional symbol used by CTC. Dropout was used to reduce overfitting in between the BLSTM layers and before the final linear layer output.

For IAM, exactly the same models as those trained in [10] were used.[6] For the other datasets, however, we used a smaller model with four convolutional blocks (with 16, 16, 32, 32 features per block, respectively), and three recurrent blocks (with 256 units each). This smaller architecture was adopted to reduce the experimental costs. Although the results may then be suboptimal for these datasets, they are superior to previous comparable state-of-the-art results, as shown later.

All the models were trained using CTC [8]. On the other hand, to combine the RNN outputs with *n*-gram language models, RNN output posterior probabilities were

[6] Training scripts are available here: https://github.com/jpuigcerver/Laia.

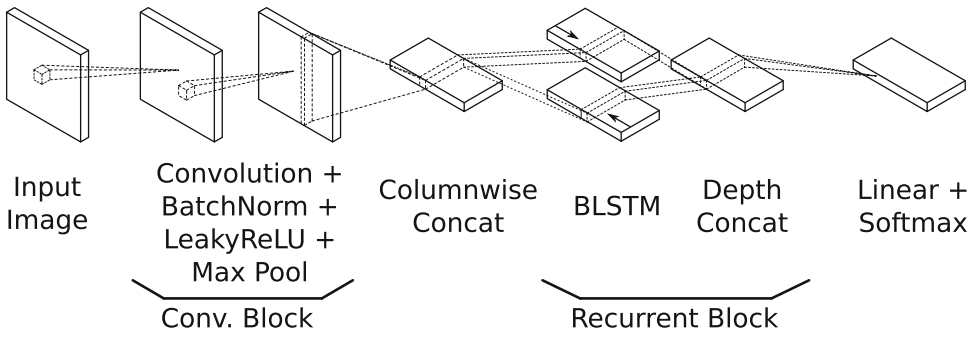

Fig. 5 Outline of the convolutional-recurrent neural network architecture used for optical modeling

transformed into pseudo-likelihoods [70]. See [10] for full details.

6.4.3 Lexicon and Language Modeling

A lexicon was extracted from the training partition of each data set. For the benchmarking corpora, the standard tokenization for each dataset was adopted. The lexicae of Bentham and Plantas were extracted through a slightly improved, specific tokenization scheme, described in detail in [68].

For each corpus, except Plantas and IAM, a 2-gram word language model was straightforwardly trained from the corresponding training text, using Kneser-Ney back-off smoothing [71]. For Plantas, a 2-gram language model was similarly trained from the original *tagged* training text produced in [12]. Then tags and modernized word versions were removed from the resulting model, thereby leaving a diplomatic-only language model for its use in PrIx. This language model is available for download along the full Plantas dataset. In the case of IAM, the large standard external text-only LOB dataset (also available from the IAM repository) was used to train the corresponding 2-gram.

Viterbi decoding meta-parameters associated with each 2-gram (grammar scale factor and word insertion penalty) were tuned to optimize the WER on the corresponding validation sets.

6.4.4 Line image word-graphs

The PrIx approaches adopted in the experiments rely on line image WGs, either to compute posteriorgrams, as discussed in Sect. 5.2, or just to straightforwardly obtain 1-best image transcription hypotheses. For HMM optical models, the procedure is a well-known variation of standard Viterbi decoding, as discussed in detail in [19]. In the case of RNN, output character posterior probabilities were converted into pseudo-likelihoods [10], which were then considered as emission densities of (single-state) character HMMs (see details in [70]). Using these simplified HMMs, the standard HMM approach was followed both to combine character likelihoods with language model probabilities and to obtain the WGs as a byproduct of Viterbi decoding (see [72, 73]).

As discussed in [19, 60], the computational cost of obtaining a WG grows very fast with the WG size. Both RNN and HMM WG sizes were controlled by means of standard pruning during the decoding process. In addition, explicit WG pruning [74] was applied to the generally larger HMM WGs, so as to make them comparable with those obtained using RNNs (see details in [38]).

WGs were *normalized* as discussed in [19]. A calibration parameter (called "logarithm base factor" in [19]) was used in this step to empirically tune the posterior probability calibration [75]. It was tuned for each optical modeling approach on the validation partition of each corpus.

6.4.5 Posteriorgrams and line-region relevance probabilities

From the normalized WGs, 1-D posteriorgrams, $P(v\,|\,\mathrm{x}, i)$, were obtained as explained in Sect. 5.2. Finally, line-level keyword relevance probabilities, $P(R\,|\,\mathrm{x}, v)$, were calculated as explained in Sects. 4.1 and 4.2. In particular, Eq. (20) [or, more specifically, its 1-D version, Eq. (22)] was used in all the experiments, while other approaches were tested only for the results reported in Sect. 7.1.

In two of these approaches [the 1-D versions of Eqs. (15) and (16)], a threshold was employed to find significant local maxima of the 1-D posteriorgram. A local maximum was detected if $P(v\,|\,\mathrm{x}, i-1) - P(v\,|\,\mathrm{x}, i)$ became larger than the threshold. The values of this threshold were optimized using validation partitions.

7 Results

First, only the Bentham dataset and RNN optical models were used to empirically explore the relative performance of the approximations here proposed to compute keyword relevance probabilities. Then, for the best approximation,

PrIx (or KWS) evaluation results were obtained both for BENTHAM and PLANTAS using both RNN and HMM optical modeling. Finally, our best system was applied to the three benchmarking datasets and the results are compared with other state-of-the-art results for these corpora.

7.1 Testing different approximations to compute relevance probabilities

For the first series of experiments, aimed to assess and compare the different approximations to the relevance probability $P(R \mid \mathrm{x}, v)$ proposed in Sects. 4.1 and 4.2, only the BENTHAM dataset and RNN optical modeling were used. Table 2 reports the (interpolated) average precision (AP) achieved. The approximations range from the roughest ones given by $P(v \mid \mathrm{x})$ [Eq. (10), using Eq. (11)] and 1-best KWS, to the potentially most accurate, but also much more computationally expensive approximation, given by by Eq. (24). In order to illustrate the challenges entailed by trying to spot *non-relevant* keywords, AP results using only *relevant* queries are also shown in Table 2 (column AP_r).

The results achieved by Eqs. (16)/(17), (18)/(20) and (24) are practically identical. As discussed in Sect. 4.4, the computational cost of Eq. (24) is too high as compared with the posteriorgram-based approaches. Therefore this result is reported here only as a reference point. As expected, the other approximations [Eqs. (11), (13), (15)] are significantly worse, the naive 1-best providing the worst performance.

Among the approximations considered, Eq. (18)/(20) is as good as the best ones, and also the fastest and simplest one, and it does not have any meta-parameter which needs to be tuned. In what follows, results will be reported only for this approach.

Table 2 BENTHAM interpolated average precision (AP) for various approximations to $P(R \mid \mathrm{x}, v)$, using RNN optical models

Approximations to $P(R \mid \mathrm{x}, v)$	AP	AP_r
Equation (11)—line-region word posterior	0.782	0.884
Equation (13)—1-best transcripts	0.763	0.821
Equation (15)—Sum (improper distribution)	0.879	0.918
Equation (16)—Naive Bayes, by DP, using Eq. (17)	0.913	0.950
Equation (18)—Max(direct with Eq. (20))	0.914	0.952
Equation (24)—"Exact" (high computing cost)	0.913	0.950

AP_r corresponds to the reduced query sets of *relevant-only* queries (Table 1)

7.2 Comparing HMM and RNN optical modeling

Using only Eq. (18)/(20) [or its 1-D version, Eq. (22)], the second series of experiments were devoted to study how the search and retrieval performance is affected by adopting different optical models (HMM and RNN) to compute the posteriorgrams. In addition to the BENTHAM dataset, the other large dataset presented in Sect. 6.2 (PLANTAS) is considered. The resulting R-P curves are shown in Fig. 6, along with the corresponding AP results. For completeness, naive 1-best KWS results [Eq. (13)] using RNNs are also given.

According to these results, only small performance differences are observed between choosing HMM or RNN for optical modeling. This applies to both BENTHAM and PLANTAS, both of which are fairly large datasets which exhibit important and different handwriting challenges.

While RNNs are known to generally outperform HMMs for character optical modeling, the present results suggest that this superiority mainly affects the *modes* of the modeled distributions – thereby typically leading to 1-best transcript with better character error rates. However, when, as in PrIx, the whole word-level distribution is brought into play, the superiority appears less obvious.

7.3 Results obtained with benchmarking datasets

Additional experiments were carried out with three well established benchmark datasets IAM, PARZIVAL (PAR) and GEORGE WASHINGTON (GW). Figure 6 shows R-P curves and AP results obtained for these datasets, using RNN optical modeling and the relevance probability approximation of Eq. (18)/(20).

To place our results in comparison with previously published work, Table 3 presents word-segmentation-free, query-by-string, line-level KWS results obtained by other authors on the same three datasets. The following approaches have been considered: convolutional deep belief network (CDBN) [76, 77], dynamic time warping (DTW) [48, 76], Bayesian logistic regression classifier (BLRC) [78], HMMs [19, 51, 77, 79] histogram of gradients (HOG) [48], our previous work on lexicon-based KWS [19], HMM-filler with background modeling (Filler-BGR) [52], and bidirectional long-short term memory (BLSTM) [50], The pyramidal histogram of characters (PHOC) approach presented in [58] is *not* included because it is fully *segmentation-based* [13].

It should be noted that the experimental setups adopted in some of these works may vary significantly with respect to the setup adopted in this work. In particular, the results marked with † were obtained using query sets selected

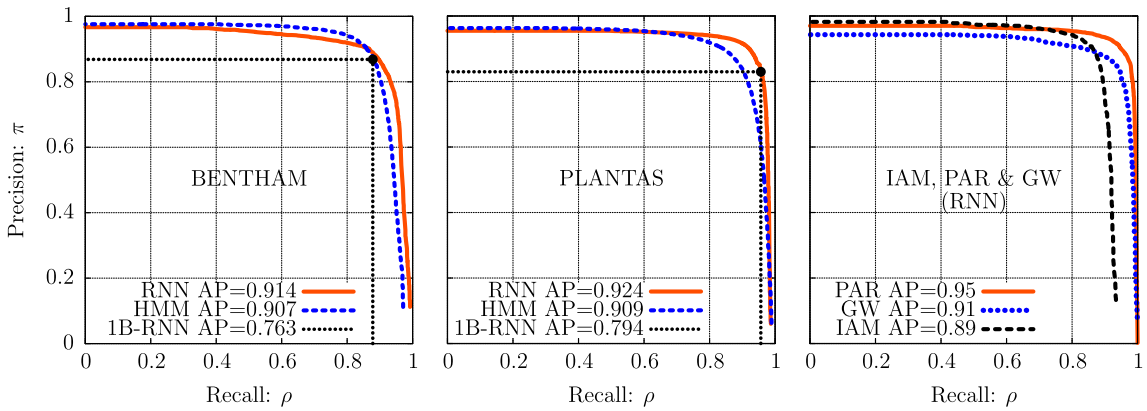

Fig. 6 Interpolated R-P curves and AP results for BENTHAM and PLANTAS, using both HMM and RNN optical modeling, and IAM, PAR and GW using only RNN optical modeling. The (degenerate) curves for naive 1-best KWS are also shown for BENTHAM and PLANTAS

Table 3 AP results achieved by several line-region KWS approaches on IAM, PAR and GW, with varied empirical setups loosely comparable with ours

References	Approach	IAM	PAR	GW
[76]	CDBN-DTW	–	0.59†	0.56†
[78]	BLRC	0.49‡	–	–
[79]	2-gram Filler-HMM	0.55†	–	0.74†
[51]	Classic Filler-HMM	–	0.86†	0.62†
[52]	Filler-BGR	0.58‡	–	–
[77]	CDBN-HMM	0.65†	**0.92** †	0.71†
[48]	HOG-DTW	–	–	**0.79** †
[19]	HMM + 2-gram LM	**0.72**	0.89	0.77
[50]	BLSTM	**0.78**	**0.94**	**0.84**
This work (Fig. 6)		**0.89**	**0.95**	**0.91**

Three best results for each dataset are typeset in boldface

†Query set selected from test transcripts (all keywords are relevant)

‡Query set is much smaller than that used in this work

from the test partitions and those marked with ‡ using much smaller query sets. In addition, in some cases it is not completely clear whether the results are provided in terms of AP or mAP.[7] Therefore, the results of Table 3 can only be considered loosely comparable. Notwithstanding the differences, we think the superiority of the methods proposed in this work can be sensibly acknowledged.

7.4 Storage efficiency

As compared with most KWS techniques, our approach relies on "off-line" pre-computation of PrIx's that allow fast "on-line" search for information in large collections of text images. Therefore it is relevant to analyze how much information needs PrIx to pre-compute and store.

A rule of thumb, followed by our methods, is that PrIx's should be comparably (much) smaller than the images for which they provide the indexing. For example for PLANTAS and BENTHAM, the average size of a 300 dpi uncompressed page image is about 30 MB. For the same datasets, PrIx produces an average of about 2000 spots per page (see details in Table 4 and in [38]). And the size of a PrIx spot is no more than 20 bytes, assuming we allot 8 characters per word on average, plus 4 and 8 bytes to store the relevance probability and the bounding box, respectively. Therefore the ratio from (uncompressed) PrIx-size to Image-Size is less than 0.0015.

Clearly, the size of an image depends on its resolution and many other factors. Therefore, we prefer a less ambiguous analysis where PrIx size is instead compared with the size of a "perfect" transcript of the image. We refer to this metric as *PrIx density* and define it as the ratio between the total number of PrIx spots and the number of running words written in the indexed images. To put it another way, it is the average number of word hypotheses that PrIx needs to retain to achieve its AP performance. This metric, along with the AP values already given in Fig. 6, is reported in Table 4 for BENTHAM and PLANTAS. Obviously, the density of 1-best HTR transcripts is close to 1, but the AP is poor. In contrast, the AP of PrIx is much better, at the expense of a larger density.

8 Conclusions and outlook

A probabilistic indexing framework for query-by-string, word-segmentation-free, lexicon-based KWS, aimed at providing access to the textual contents of large collections of handwritten text images, has been presented. The

[7] Mean of single-keyword APs, see [38].

Table 4 Search performance (AP) and indexing density for PLANTAS and BENTHAM, using the two optical modeling approaches, HMM and RNN, for PrIx and 1-best HTR. Density is the ratio between the number of PrIx spots and the number of running words in the ground-truth transcripts

	PrIx HMMs		PrIx RNN		1-best HMMs		1-best RNN	
	AP	Density	AP	Density	AP	Density	AP	Density
BENTHAM	0.907	7.8	0.914	7.9	0.740	1.0	0.763	0.9
PLANTAS	0.909	13.0	0.924	13.5	0.722	1.0	0.794	1.0

formulation of this framework, referred to as PrIx, makes it self-evident that KWS is always more or less explicitly based on word recognition posterior probabilities, and provides probabilistic interpretations of many classical KWS views and methods. Various developments of this framework into specific PrIx approaches have been proposed and empirically evaluated. The most efficient and effective of these approaches are based on (line) image region posteriorgrams, obtained from word-graph representations of the joint probability distributions of image regions and the text contained therein. As discussed in previous works, these distributions can be advantageously estimated using the same statistical models and training methods, and similar decoding procedures as those used in state-of-the-art handwritten text recognition systems. Following this idea, our approaches outperform more or less significantly all the methods proposed and tested so far on three traditional benchmarking datasets. Moreover, results obtained for two larger, more practically realistic historical corpora are also very good and clearly show the capabilities of these approaches for indexing large collections of handwritten text images.

The PrIx approaches presented in this paper are all *lexicon-based* (LB). As applied to large-scale indexing, LB methods in general are known to be faster and more accurate than *lexicon-free* (LF) ones, based on raw character processing. However, since LB KWS relies on a predefined lexicon, fixed in the training phase, it does not support queries involving out-of-vocabulary (OOV) keywords. This issue has *not* been explicitly studied in the present work. In fact, aiming to obtain results comparable with other state-of-the-art KWS approaches, the experiments were carried out with query sets selected from the training texts, thereby guaranteeing that all query words are in-vocabulary. It is worth noting, however, that while the OOV problem may be serious if the indexed vocabulary is small, it becomes much less important with very large vocabularies – which is generally the case in real indexing applications. Several live PrIx demonstrators which support this fact are publicly available for on-line testing[8] (these systems also support flexible queries such as searching for word sequences and the Boolean AND/OR/NOT query methods described in [28]).

OOV queries can be supported by smoothomg the (implicitly null) relevance probabilities of OOV query words, using the indexed probabilities of "similar" in-vocabulary words [80]. While reasonably good results are achieved with these methods, they always entail query response time penalties for OOV queries – and these penalties can become prohibitive for large collections of say hundreds of thousands or millions of images. Therefore our current work for LF PrIx abandons the use a lexicon altogether to favor working at character level. However, it also attempts to keep the good performance of LB PrIx by actually producing relevance probabilities for "pseudo-words" (in fact arbitrary character sequences) which are "discovered" in the very test images being indexed. Key ideas and results in this direction have already appeared in [22, 54, 81] and actual search interfaces have been developed and are publicly available for some large-scale collections of historical manuscripts, such as: CHANCERY [20] (more than 83,000 page images),[9] The (full) BENTHAM PAPERS[10] (90,000 images), SPANISH THEATER GOLDEN AGE[11] (TSO, 40,000 images), CARABELA[12] (125,000 images) and FINISH COURT RECORDS[13] (FCR, 100,000 images). The techniques used in these works will be presented in an upcoming publication devoted to lexicon-free probabilistic indexing.

Acknowledgements Work partially supported by ValgrAI—Valencian Graduate School and Research Network of Artificial Intelligence (Generalitat Valenciana) co-funded by the European Union; and by a María Zambrano grant of the Spanish Ministerio de Universidades and the European Union NextGenerationEU/PRTR.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data Availability Statement The datasets generated during and/or analyzed during the current study are available in the following publically available repositories: BENTHAM: https://zenodo.org/record/44519#.YxCuDEhBzZ8, PLANTAS: https://zenodo.org/record/

[8] See: http://prhlt-carabela.prhlt.upv.es/PrIxDemos.

[9] http://prhlt-kws.prhlt.upv.es/himanis.

[10] http://prhlt-kws.prhlt.upv.es/bentham.

[11] http://prhlt-carabela.prhlt.upv.es/tso.

[12] http://prhlt-carabela.prhlt.upv.es/carabela.

[13] http://prhlt-kws.prhlt.upv.es/fcr.

6608342#.YxCu-0hBzZ9 and IAM, PAR, GW: http://www.fki.inf.unibe.ch/databases.

Declarations

Conflict of interest All authors declare that they have no conflicts of interest.

References

1. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press
2. Bache R, Ballie M, Crestani F (2013) The likelihood property in general retrieval operations. Inf Sci 234:97–111
3. Joung Y-J, Yang L-W (2014) On character-based index schemes for complex wildcard search in peer-to-peer networks. Inf Sci 272:209–222
4. Wu M-S (2015) Modeling query-document dependencies with topic language models for information retrieval. Inf Sci 312:1–12
5. Bazzi I, Schwartz R, Makhoul J (1999) An omnifont open-vocabulary OCR system for English and Arabic. IEEE Trans PAMI 21(6):495–504
6. Vinciarelli A, Bengio S, Bunke H (2004) Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. IEEE Trans PAMI 26(6):709–720
7. Toselli AH, Juan A, Keysers D, González J, Salvador I, Ney H, Vidal E, Casacuberta F (2004) Integrated handwriting recognition and interpretation using finite-state models. Int J Pattern Recogn Artif Intell 18(4):519–539
8. Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. IEEE Trans PAMI 31(5):855–868
9. Sánchez JA, Romero V, Toselli AH, Vidal E (2016) ICFHR2016 competition on handwritten text recognition on the READ dataset. In: Proc. of 15th ICFHR, pp 630–635
10. Puigcerver J (2017) Are multidimensional recurrent layers really necessary for handwritten text recognition? In: Proc. of 14th ICDAR
11. Romero V, Toselli AH, Vidal E (2012) Multimodal interactive handwritten text transcription. In: Perception and artif. intell. (MPAI). World Scientific
12. Toselli AH, Leiva LA, Bordes-Cabrera I, Hernández-Tornero C, Vicent B, Vidal E (2017) Transcribing a 17th century botanical manuscript: longitudinal interactive transcription evaluation and ground truth production. Digit Scholarsh Humanit 33(1):173–202
13. Giotis AP, Sfikas G, Gatos B, Nikou C (2017) A survey of document image word spotting techniques. Pattern Recogn 68:310–332
14. Christiansen R, Rushforth CK (1977) Detecting and locating key words in continuous speech using linear predictive coding. IEEE Trans Acoust, Speech Signal Process 25(5):361–367
15. Chelba C, Silva J, Acero A (2007) Soft indexing of speech content for search in spoken documents. Comput Speech Lang 21(3):458–478
16. Chia TK, Sim KC, Li H, Ng HT (2010) Statistical lattice-based spoken document retrieval. ACM Trans Inf Syst 28(1):2–1230
17. Tabibian S, Akbari A, Nasersharif B (2016) A fast hierarchical search algorithm for discriminative keyword spotting. Inf Sci 336:45–59
18. Tabibian S, Akbari A, Nasersharif B (2018) Discriminative keyword spotting using triphones information and n-best search. Inf Sci 423:157–171
19. Toselli AH, Vidal E, Romero V, Frinken V (2016) HMM word graph based keyword spotting in handwritten document images. Inf Sci 370–371:497–518
20. Bluche T, Hamel S, Kermorvant C, Puigcerver J, Stutzmann D, Toselli AH, Vidal E (2017) Preparatory KWS experiments for large-scale indexing of a vast medieval manuscript collection in the HIMANIS project. In: Proc. of 14th ICDAR
21. Lang E, Puigcerver J, Toselli AH, Vidal E (2018) Probabilistic indexing and search for information extraction on handwritten German parish records. In: 16th ICFHRx, pp 44–49
22. Puigcerver J (2018) A probabilistic formulation of keyword spotting. PhD thesis, Universitat Politècnica de València
23. Toselli AH, Romero V, Sánchez JA, Vidal E (2019) Making two vast historical manuscript collections searchable and extracting meaningful textual features through large-scale probabilistic indexing. In: Int. conf. on document analysis and recogn. (ICDAR). IEEE, pp 108–113
24. Vidal E, Romero V, Toselli AH, Sánchez J-A, Bosch V, Quirós L, Benedí JM, Prieto JR, Pastor M, Casacuberta F, et al (2020) The Carabela project and manuscript collection: large-scale probabilistic indexing and content-based classification. In: 2020 17th international conference on frontiers in handwriting recognition (ICFHR). IEEE, pp 85–90
25. Ono A, Amano M, Hakaridani M, Satou T, Sakauchi M (1996) A flexible content-based image retrieval system with combined scene description keyword. In: Proceedings of the 3rd IEEE international conference on multimedia computing and systems. IEEE, pp 201–208
26. Mori Y, Takahashi H, Oka R (1999) Image-to-word transformation based on dividing and vector quantizing images with words. In: 1st international workshop on multimedia intelligent storage and retrieval management, pp 1–9
27. Bradshaw B (2000) Semantic based image retrieval: a probabilistic approach. In: Proceedings of the 8th ACM international conference on multimedia, pp 167–176
28. Toselli AH, Vidal E, Puigcerver J, Noya-García E (2018) Probabilistic multi-word spotting in handwritten text images. Pattern Anal Appl 22:23–32
29. Vidal E, Toselli AH, Puigcerver J (2015)High performance query-by-example keyword spotting using query-by-string techniques. In: Proc. of 13th ICDAR, pp 741–745
30. Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley
31. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans PAMI 39(6):1137–1149
32. En S, Petitjean C, Nicolas S, Heutte L, Jurie, F (2016) Region proposal for pattern spotting in historical document images. In: Proc. of 15th ICFHR. IEEE, pp 367–372
33. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969

34. Prusty A, Aitha S, Trivedi A, Sarvadevabhatla RK (2019) Indiscapes: instance segmentation networks for layout parsing of historical Indic manuscripts. In: 2019 international conference on document analysis and recognition (ICDAR). IEEE, pp 999–1006
35. Studer L, Alberti M, Pondenkandath V, Goktepe P, Kolonko T, Fischer A, Liwicki M, Ingold R (2019) A comprehensive study of imagenet pre-training for historical document image analysis. In: 2019 international conference on document analysis and recognition (ICDAR). IEEE, pp 720–725
36. Ventsel H (1973) Théorie des probabilités. Éditions MIR, Moscow
37. Hailperin T (1965) Best possible inequalities for the probability of a logical function of events. Am Math Mon 72(4):343–359
38. Vidal E, Toselli AH, Puigcerver J (2017–2021) A probabilistic framework for lexicon-based keyword spotting in handwritten text images. Tech. Rep. arXiv:2104.04556
39. Sanchis A, Juan A, Vidal E (2012) A word-based Naive Bayes classifier for confidence estimation in speech recognition. IEEE Trans Audio, Speech, Lang Process 20(2):565–574
40. Serrano N, Giménez A, Civera J, Sanchis A, Juan A (2014) Interactive handwriting recognition with limited user effort. Int J Doc Anal Recogn (IJDAR) 17(1):47–59
41. Barrere K, Toselli AH, Vidal E (2019) Line segmentation free probabilistic keyword spotting and indexing. In: Iberian conf. on pattern recognition and image analysis. Springer, pp 201–217
42. Oliveira SA, Seguin B, Kaplan F (2018) dhsegment: a generic deep-learning approach for document segmentation. In: 2018 16th international conference on frontiers in handwriting recognition (ICFHR). IEEE, pp 7–12
43. Grüning T, Leifert G, Strauß T, Michael J, Labahn R (2019) A two-stage method for text line detection in historical documents. Int J Doc Anal Recogn (IJDAR) 22(3):285–302
44. Quirós L (2018) Multi-task handwritten document layout analysis. arXiv preprint arXiv:1806.08852
45. Renton G, Soullard Y, Chatelain C, Adam S, Kermorvant C, Paquet T (2018) Fully convolutional network with dilated convolutions for handwritten text line segmentation. Int J Doc Anal Recogn (IJDAR) 21(3):177–186
46. Bluche T, Moysset B, Kermorvant C (2014) Automatic line segmentation and ground-truth alignment of handwritten documents. In: Proc. of 14th ICFHR, pp 667–672
47. Kolcz A, Alspector J, Augusteijn M, Carlson R, Viorel Popescu G (2000) A line-oriented approach to word spotting in handwritten documents. Pattern Anal Appl 3:153–168
48. Terasawa K, Tanak Y (2009) Slit style HOG feature for document image word spotting. In: Proc. of 10th ICDAR, pp 116–120
49. Thomas S, Chatelain C, Heutte L, Paquet T (2010) Alpha-numerical sequences extraction in handwritten documents. In: Proc. of ICFHR, pp 232–237
50. Frinken V, Fischer A, Manmatha R, Bunke H (2012) A novel word spotting method based on recurrent neural networks. IEEE Trans PAMI 34(2):211–224
51. Fischer A, Keller A, Frinken V, Bunke H (2012) Lexicon-free handwritten word spotting using character HMMs. Pattern Recogn Lett 33(7):934–942
52. Wshah S, Kumar G, Govindaraju V (2014) Statistical script independent word spotting in offline handwritten documents. Pattern Recogn 47(3):1039–1050
53. Toselli AH, Vidal E (2013) Fast HMM-filler approach for key word spotting in handwritten documents. In: Proc. of the 12th ICDAR. IEEE Computer Society, Washington
54. Toselli AH, Puigcerver J, Vidal E (2016) Two methods to improve confidence scores for lexicon-free word spotting in handwritten text. In: Proc. 15th ICFHR, pp 349–354
55. Puigcerver J, Toselli AH, Vidal E (2015) Probabilistic interpretation and improvements to the HMM-filler for handwritten keyword spotting. In: Proc. of the 13th ICDAR, pp 731–735
56. Rodríguez-Serrano JA, Perronnin F (2009) Handwritten word-spotting using hidden Markov models and universal vocabularies. Pattern Recogn 42:2106–2116
57. Hast A, Fornés A (2016) A segmentation-free handwritten word spotting approach by relaxed feature matching. In: Proc. of 12th DAS Wksh., pp 150–155
58. Almazán J, Gordo A, Fornés A, Valveny E (2014) Word spotting and recognition with embedded attributes. IEEE Trans PAMI 36(12):2552–2566
59. Ortmanns S, Ney H, Aubert X (1997) A word graph algorithm for large vocabulary continuous speech recognition. Comput Speech Lang 11(1):43–72
60. Toselli AH, Romero V, Vidal E (2017 first online 2016) Word graphs size impact on the performance of handwriting document applications. Neural Comput Appl 28(9):2477–2487
61. Egghe L (2008) The measures precision, recall, fallout and miss as a function of the number of retrieved documents and their mutual interrelations. Inf Process Manag 44(2):856–876
62. Zhu M (2004) Recall, precision and average precision. Working Paper 2004-09 Department of Statistics & Actuarial Science, University of Waterloo
63. Robertson S (2008) A new interpretation of average precision. In: Proc. of the Int. Conf. on R &D in Information Retrieval (SIGIR'08), pp 689–690
64. Andreu Sánchez J, Romero V, Toselli AH, Vidal E (2014) ICFHR2014 competition on handwritten text recognition on transcriptorium datasets (HTRtS). In: Proc. of 14th ICFHR, pp 785–790
65. Jelinek F (1998) Statistical methods for speech recognition. MIT Press
66. Sánchez JA, Romero V, Toselli AH, Villegas M, Vidal E (2019) A set of benchmarks for handwritten text recognition on historical documents. Pattern Recogn 94:122–134
67. Kozielski M, Forster J, Ney H (2012) Moment-based image normalization for handwritten text recognition. In: Proc. of ICFHR. ICFHR. IEEE Computer Society, pp 256–261
68. Toselli AH, Vidal E (2015) Handwritten text recognition results on the bentham collection with improved classical N-Gram-HMM methods. In: Proc. 3rd int. wksh. on historical document imaging and process (HIP), pp 15–22
69. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681
70. Bluche T (2015) Deep neural networks for large vocabulary handwritten text recognition. PhD thesis, Univ. Sud-Paris XI
71. Kneser R, Ney H (1995) Improved backing-off for N-gram language modeling. In: Proc. of ICASSP, vol 1, pp 181–184
72. Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, Hannemann M, Motlicek P, Qian Y, Schwarz P, Silovsky J, Stemmer G, Vesely, K (2011) The Kaldi speech recognition toolkit. In: IEEE workshop on automatic speech recognition and understanding
73. Povey D, Hannemann M, Boulianne G, Burget L, Ghoshal A, Janda M, Karafiát M, Kombrink S, Motlíček P, Qian Y, et al (2012) Generating exact lattices in the WFST framework. In: IEEE ICASSP, pp 4213–4216
74. Zens R, Ney H (2005) Word graphs for statistical machine translation. In: Proc. of the ACL wksh. on building and using parallel texts, pp 191–198
75. Niculescu-Mizil A, Caruana R (2005) Predicting good probabilities with supervised learning. In: Proc. of the 22nd int. conf. on machine learning (ICML), pp 625–632

76. Wicht B, Fischer A, Hennebert J (2016) Keyword spotting with convolutional deep belief networks and dynamic time warping. In: Proc. 25th int. conf. on artificial neural networks, pp 113–120
77. Wicht B, Fischer A, Hennebert J (2016) Deep learning features for handwritten keyword spotting. In: Proc. of the 23rd ICPR, pp 3434–3439
78. Kumar G, Govindaraju V (2014) A Bayesian approach to script independent multilingual keyword spotting. In: Proc. of 14th ICFHR, pp 357–362
79. Fischer A, Frinken V, Bunke H, Suen CY (2013) Improving HMM-based keyword spotting with character language models. In: Proc. of 12th ICDAR, pp 506–510
80. Puigcerver J, Toselli AH, Vidal E (2017) Querying out-of-vocabulary words in lexicon-based keyword spotting. Neural Comput Appl 28(9):2373–2382
81. Toselli AH, Puigcerver J, Vidal E (2015) Context-aware lattice based filler approach for key word spotting in handwritten documents. In: Proc. of the 13th ICDAR, pp 736–740