Line Segmentation Free Probabilistic Keyword Spotting and Indexing

Killian Barrere[1], Alejandro H. Toselli[2(✉)], and Enrique Vidal[2]

[1] Univ Rennes, Rennes, France
[2] Universitat Politècnica de València, Valencia, Spain
alto@upv.es

Abstract. Probabilistic Keyword Spotting and Indexing (PKWSI) allows effective search through untranscribed large collections of images. However, when text-line detection fails to detect foreground text, the PKWSI techniques also fail dramatically. In this paper, we develop a new line segmentation-free approach using a uniform line-sized image slicing instead of previous text-line detection. As a result, new issues arise due to overlapping slices, leading to several spot hypotheses for the same word. We develop solutions to take advantage of multiple spots and to consolidate them into single hypotheses. We test our approach on a difficult historical handwritten dataset and it yields promising results.

Keywords: Keyword spotting · Probabilistic indexing · Handwritten text recognition · Segmentation free

1 Introduction

In recent works, Probabilistic Keyword Spotting and Indexing (PKWSI) has proven to be a promising approach to make the textual contents of untranscribed handwritten text images accessible [1,4,8,12]. This applies to search and retrieval, as well as to other highly demanded information extraction tasks [4,8]. PKWSI has achieved in recent years a great level of maturity, allowing effective and accurate access to textual contents of large collections of handwritten text images.

One remaining bottleneck of this technology is the need for previous detection and extraction of the relevant text lines. While this can be very reliably done for clean, well-written handwritten documents, it becomes a problematic bottleneck for many historic manuscript collections. Clearly, when the text-line detection fails to detect foreground text, the PKWSI techniques also fail dramatically, often leading to useless probabilistic indices of many images of the collection.

In this paper, we develop and test a new PKWSI approach which does not rely on previous line detection. Instead, each image is uniformly scanned vertically into line-sized rectangular image slices. In this new approach, a new issue arises due to the fact that several line-shaped slices often become all relevant, leading to several spot hypotheses for the same word. We study this problem formally

A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11868, pp. 201–213, 2019.
https://doi.org/10.1007/978-3-030-31321-0_18

and develop solutions to consolidate multiple spots of the same word into a single spot with its corresponding relevance probability. The new approach is tested on a difficult dataset of historical handwritten images and the experiments yield promising results.

In this paper, Sect. 2 starts by describing current keyword spotting approaches. Then we introduce and explain formally our approach in Sect. 3. Following that, we present the different experiments and results in Sect. 4.

2 Probabilistic Keyword Spotting and Indexing

Keyword spotting can be seen as a binary classification problem to decide whether a particular image region x is relevant for a given query word v, i.e. try to answer the following question: "Is v actually written in x?". As in [8], we aim to compute the image region word relevance probability $P(R = 1 \mid X = x, V = v)$. From now on, for the sake of clarity, we will omit the random variable names, and for $R = 1$, we will simply write R.

Image region word relevance probabilities are computed without taking into account where the considered word may appear in the region x. Nevertheless, it should be pointed out that the precise positions of the words within x are easily obtained as a byproduct. The relevance probability of an image region x for a keyword v, $P(R \mid x, v)$, is approximately computed as:

$$P(R \mid x, v) \approx \max_{b \sqsubseteq x} P(R \mid x, b, v) \tag{1}$$

where b is a word-sized image region or Bounding Box (BB), and with $b \sqsubseteq x$ we mean the set of all BBs contained in x [8,12–14].

Since b is assumed to (tightly) contain only one word (hopefully v), it is straightforward to see that $P(R \mid x, b, v) = P(v \mid x, b)$. This is just the posterior probability needed to "recognize" the BB image (x, b), or more formally speaking, to classify the BB (x, b) into one of the possible words of some vocabulary. The approximate computation of this probability is carried out using processing, training, optical and language models steps, similar to those employed in handwritten text recognition, even though no actual text transcripts are produced in PKWSI. Instead, for each image x, the distribution $P(R \mid x, b, v)$ itself is obtained and adequately indexed to allow efficient textual search and information retrieval [8,12–14].

It is important to remark that the PKWSI framework is not limited to just a single word query; it can also accommodate sequences of words or characters [8]. This raises a distinction into the two approaches referred to as Lexicon-Based (LB) and Lexicon-Free (LF). In general, LB methods are known to be faster and more accurate than LF ones. However, since LB PKWSI relies on a predefined lexicon, fixed in the training phase, it does not support queries involving out-of-vocabulary keywords.

On the other hand, the LF PKWSI approach works usually at character level, but it attempts to keep the good performance of LB indexing by actually producing relevance probabilities for what character sequences called *pseudo-words*, which are automatically "discovered" in the very test images being indexed [8,11]. This approach has proved to be very robust, and it has in actually been used to very successfully index two iconic large collections: The French Chancery Collection [1],[1] and Bentham papers,[2] containing 90 000 manuscript images written in old English.

Target Image Regions

Up to this point we have not clearly specified what the image regions x are. Depending on the size and shape of x, Eq. (1) may become more or less difficult to compute. In the traditional keyword spotting literature, word-sized regions have often been considered. This is reminiscent of segmentation-based methods which required previously cropped accurate word BBs. 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.

On the other hand, we may consider whole page images, or relevant *text blocks* thereof, as the search target image regions. While it 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 to compute accurate word classification probabilities. Moreover, as discussed in [8,12,14], line region relevance probabilities can be very efficiently computed.

3 Full Segmentation-Free PKWSI

So far, to use line-shaped image regions in PKWSI, it is assumed that text lines have previously been detected. As discussed in Sect. 1, it constitutes a serious bottleneck. Here we propose a new approach which does not rely on previous line detection. Instead, each image is scanned vertically and is uniformly sliced into line-shaped, rectangular image regions, where the methods discussed in previous sections are applied. Thanks to the robustness of the relevance probability estimates, those image regions where no text is actually present generally get low probability for any word, while in regions which actually contain text, word relevance probabilities become high as in the previous line-based approach. According to the usual terminology in the field of keyword spotting [3], we will refer to our new approach as Line Segmentation-Free PKWSI (LSF-PKWSI).

3.1 Vertical Sampling of Line-Shaped Image Regions

The principle of vertical sampling is to extract consecutive line-shaped rectangular images using a page-wide sliding window of fixed height. This window

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

[2] http://prhlt-carabela.prhlt.upv.es/bentham.

determines the region to extract and is shifted by a fixed number of pixels at each step. While in line-based PKWSI there is one line-image per line of text, in LSF-PKWSI there are typically several overlapping rectangular windows, possibly containing the same (parts of a) text line. We also expect a lot of regions without text; mainly in the borders of pages and between columns, if they have more than one.

Figure 1 shows several consecutive windows resulting from the proposed vertical sampling process. It showcases that the method can extract relevant image regions even when text lines are significantly slanted. In the example, we expect to retrieve all the existing words, but on separate sliding windows.

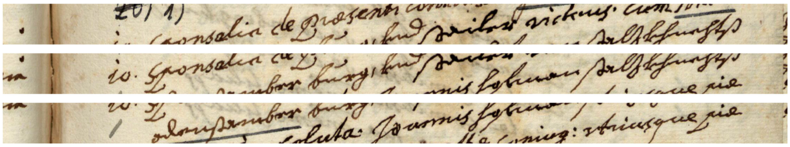


Fig. 1. Consecutive windows extracted with a Vertical Sampling Rate of 1.5.

By extracting more windows than those strictly required, LSF-PKWSI aims to avoid the need for text-line detection, thereby circumventing problems related with no keywords being spotted in image regions where lines are poorly detected.

3.2 Estimating Vertical Sampling Parameters

To adapt the vertical sampling to the writing density of each text image, an estimate of the height of the text lines composing each page is needed.

We are aware that there are already very good text-line detection methods [2]. Obviously, such systems could be used to estimate line heights, but we believe that a much simpler method should be sufficient for our purposes. In comparison, state-of-the-art line detection methods require the usage of Artificial Neural Networks and may require training. In contrast, our method is mainly based on applying Fourier transform on a signal representing the amount of ink in each row of the image. Figure 2 illustrates the whole process.

To obtain our estimates, we first pre-process the page images. We apply a Gaussian Blur to remove the high frequencies and the noise. Then, we binarize the image at a local level, using the Sauvola algorithm [10]. By applying a horizontal Run Length Smoothing Algorithm (RLSA) [15], we aim to highlight the text-line-like regions. After that, we compute the average number of black pixels in each row to obtain a signal which approximately represents line vertical positions. Finally, we apply a Fourier transform to estimate several line heights, $\hat{h}$, as the highest values of the amplitude spectrum.

Once line heights are estimated, two important parameters remain to be determined. The actual height of the sampling window, which is proportional to $\hat{h}$. We are referring to this proportional factor as the *Height Factor*. Then,

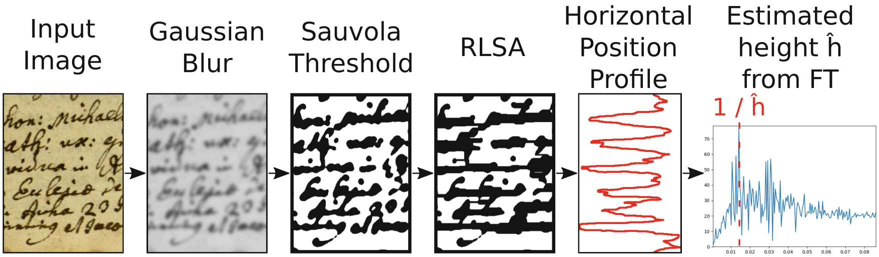


Fig. 2. Steps of the process to estimate the height of line-shaped windows.

the *Vertical Sampling Rate* which represents the number of extracted window images a pixel belongs to. It impacts the vertical overlapping of the sampling windows and the distance between each consecutive window.

Both parameters affect the chance that words are spotted correctly, hopefully leading to better relevance probabilities. However increasing that chance might also affect the speed and memory consumption of the whole process.

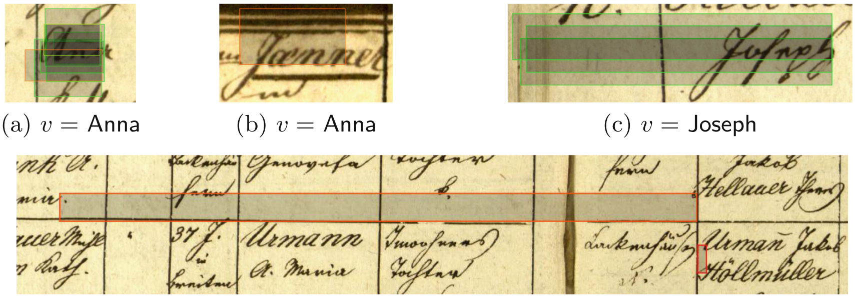
From what we have tested so far, this approach seems to be good for the dataset we are using. However, in datasets where the text orientation highly differs from the horizontal, this process would not be suitable.

3.3 Consolidating Multiple Spots of the Same Word Region

Figure 3 shows different kinds of BBs obtained with LSF-PKWSI. In a typical case, we obtain many high relevance-probability BBs which significantly overlap with each other around a true-positive spot (Fig. 3a). Conversely, false-positive spots often correspond to “lone” BBs with no other BBs for the same word in their neighborhood (Fig. 3b). We also observe very elongated BBs in areas of the image where there is no text, generally with low relevance-probability (Fig. 3d). However, elongated BBs containing a wide area without text, followed by a word on their right side may also appear (Fig. 3c). They appear to have high relevance-probability for the true-positive word. In future works we will try to avoid this kind of spots, which we believe are mostly due to a mismatch between the shapes and sizes of the line (shaped) image regions considered for training and testing.

Users generally want a single spot for each keyword. It is also expected to have a high difference in the relevance probability between true-positives and false-positives. Based on the previous observations (Fig. 3), we should group all the BBs that are overlapping into a single spot with a high relevance-probability, while either discarding or lowering the relevance-probability of lone BBs.

Let x be again a full image and b the true BB of a word-sized image region for the keyword v. In LSF-PKWSI, b is unknown, but there are several word-sized regions β, associated with b, where v is likely to be written. $P(R \mid x, b, v)$ can be computed by considering all possible BBs, β, for which v may be relevant:



(a) $v =$ Anna (b) $v =$ Anna (c) $v =$ Joseph

(d) $v =$ Anna

Fig. 3. Different situations arising with our method: (a) a true-positive with many overlapping BBs, (b) a lone false-positive, (c) a true-positive with elongated BBs, and (d) a very elongated false-positive.

$$
\begin{aligned}
P(R \mid x, b, v) \equiv P(v \mid x, b) = \sum_{\beta} P(v, \beta \mid x, b) &= \sum_{\beta} P(\beta \mid x, b) P(v \mid x, b, \beta) \\
&\approx \sum_{\beta \sqcap b} P(\beta \mid x, b) P(v \mid x, \beta) \qquad (2)
\end{aligned}
$$

We say that two BBs β_1 and β_2 (viewed as sets of pixels of x) *θ-significantly overlap* (written as $\beta_1 \sqcap \beta_2$) if $|\beta_1 \cap \beta_2| \,/\, |\beta_1 \cup \beta_2| \geq \theta$, where θ, $0 < \theta \leq 1$, is a fixed parameter for the minimum fraction of pixels that β_1 and β_2 must share (a typical value for θ is 0.5). In this case we assume that v is conditionally independent of b given β, otherwise (i.e. $\beta \not\sqcap b$), $P(\beta \mid x, b) = 0$.

In plain words, the relevance probability of b for v is computed as a weighted average of the relevance probabilities of all the BBs associated with v (obtained as explained in Sect. 2). Therefore, a best BB $\hat{b}$, for some image region where v is written should be one that maximizes $P(v \mid x, b)$. According to Eq. (2):

$$
\hat{b} \;=\; \arg\max_{b} \sum_{\beta \sqcap b} P(\beta \mid x, b)\, P(v \mid x, \beta) \qquad (3)
$$

An algorithmic solution to this optimization problem does not seem easy. Thus, we leave it for future studies. Here, we instead adopt a simple but effective heuristic approach, as discussed later in this section.

The weights, $P(\beta \mid x, b)$, of Eqs. (2) and (3) can be considered as prior probabilities of the different BBs relevant for v, conditioned by the position and shape of the unknown true BB, b. For a BB β, this probability should be high if it shares significant parts with b (and also with other nearby BBs), and low for lone BBs. In addition, this prior should be high if the shape (size) of β is adequate to hold v, and should be low if it is too large or too small for v. Based on that we built Algorithm 1.

Algorithm 1 has two parameters: θ (explained above) and τ. τ is the minimum overlapping fraction between all the BBs $\in g$ and a new BB β in order to be

Algorithm 1. Create groups of Bounding Boxes of a given word v

```
Start with no groups; i.e. $G = \emptyset$
for all BBs $\beta$ of $v$ do
    for all $g \in G$ do
        if $\beta$ overlaps with more than $\tau$ of the BBs $\in g$ then Insert $\beta$ in $g$
    if $\beta$ was not inserted in any group then Create $g = \{\beta\}; G = G \cup \{g\}$
```

added into g. Algorithm 1 returns groups of overlapping BBs. Each BB β in a group g share at least a fraction θ of its pixels with at least a fraction τ of the other BBs in the same group g.

For each relevant BB group g produced by Algorithm 1, a single merged BB $\hat{b}$ (as in Eq. (3)) has to be obtained, for which an estimate of $P(\beta \mid x, \hat{b})$ is initially needed. We explored several approximations to this single-BB prior probability. Based on these, the simple heuristic presented in Eq. (4) (which ignores the dependence on b) gave the best empirical results:

$$P(\beta \mid x, \hat{b}) \;\approx\; P(\beta) \;=\; F\Bigg(\sum_{\beta' \in g : \beta' \neq \beta} \frac{|\beta \cap \beta'|}{|\beta \cup \beta'|} \Bigg) \tag{4}$$

For each BB β, we sum the relative overlap for each other BB β' and apply a customized sigmoid function, $F(x) = 1 \,/\, (1 + e^{-ax+b})$, with parameters tuned.

Then, we compute the merged BB (our heuristic approximation to $\hat{b}$) as a weighted sum of the coordinates of all overlapping β's in g; that is:

$$\hat{b} \;\equiv\; \hat{\boldsymbol{b}} = \sum_{\beta \in g} P(\beta) P(v \mid x, \beta)\, \boldsymbol{\beta} \tag{5}$$

with $\hat{\boldsymbol{b}}$ and $\boldsymbol{\beta}$ the 4-dimensional vectors of the coordinates (center and size) of $\hat{b}$ and β, respectively.

Finally, to obtain $P(v \mid x, \hat{b})$ following Eq. (2), we should do a weighted average, as in Eq. (5). However, additional experiments suggest that somewhat better results are obtained by using the simpler maximum approximation:

$$P(v \mid x, \hat{b}) \;\approx\; \max_{\beta \in g} P(\beta) P(v \mid x, \beta) \tag{6}$$

Examples of merged BBs could be seen in Fig. 4b and c or by using both Raw and Consolidated demonstrators which are explained in Sect. 5.

4 Experiments and Results

Assessment measures, data set and partitions, query sets, experimental setup and results are presented in this section.

4.1 Assessment

PKWSI performance is assessed in terms of standard *recall* vs. *interpolated precision* curves [5], from which the *average precision* (AP) [6] and the mean AP (mAP) [9,16] are obtained. While the AP is computed from a *global* ranked list containing all the results from all queries, the mAP is the mean of the APs of the individual queries. For the results presented in this section, AP and mAP have been computed using a publicly available tool called KwsEvalTool.[3]

In line-based PKWSI, these scores are computed at line level. Yet the transcripts of the image windows extracted by the LSF-PKWSI approach are not (precisely) known. To compare with previous results, we then decided to use a fair evaluation that could be used in both line-based PKWSI and LSF-PKWSI, without incurring the high cost of creating a detailed BB-based ground truth.

To this end, a simple idea is to evaluate the performance at the page level. This amounts to ask whether each keyword is written on a page or not. We compute the relevance probability at the page level for each keyword v, by taking the maximum of the $P(v \mid x, \hat{b})$ from Eq. (6) according to Eq. (1). It should be noted that, however with this approximation, we ignore repeated occurrences of keywords in each given page, which may be rather likely for some keywords.

4.2 Dataset and Experimental Partition

The 289 images of the dataset used in this work were selected from a subset of 57 222 scans of more than 800 000 sacramental register images belonging to the Passau Diocesan Archives[4]. The images show a great variety in the evolution of handwriting, record keeping and more and more standardized table forms introduced in the early 19th century. For more details about this collection and dataset, refer to [4].

Table 1 shows relevant details of the dataset used in this work. 179 images were selected for training, 21 for validation and the remaining 89 for testing. The large number of test-set image windows needed in our approach typically requires relatively important amounts of computing time and space. Hence, we decided to select a small subset of 10 test-set images from which we obtained the first encouraging results. This subset was later used as a further validation set to tune the parameters of the methods discussed in Sect. 3. It includes 4 images with large tables and 6 pages without. We will refer to this set as TestVal.

4.3 Query Set

The query set used in this work was adopted according to the most common criteria, where most of the words seen in the test set are chosen as keywords. Besides being a meaningful choice from an application point of view, it ensures that all the keywords are relevant (appear in one or more test images), thereby

[3] https://github.com/PRHLT/KwsEvalTool.git.

[4] Openly available at http://data.matricula-online.eu/de/deutschland/passau.

Table 1. The Passau experimental dataset. All the figures correspond to a transliterated version where all letters were case and diacritics folded.

Number of:	Train+Val	TestVal	Test	Total
Pages	200	10	89	289
Lines	29 314	2 053	16 376	45 690
Running words	72 848	5 204	37 354	110 207
Running words excluding punctuation	–	3 712	26 709	15 141
Different words	11 160	1 191	5 801	16 169
Different characters	99	48	87	102
Query words	–	1 043	5 725	–

allowing mAP to be properly computed. All the test-set words longer than 1 character are used, making a total of 1 043 and 5 725 transliterated keywords for the TestVal and the Test partitions respectively (see Table 1).

4.4 Experimental Setup

As discussed in Sects. 2 and 3, a primary step in the LSF-PKWSI approach is to obtain the $P(v \mid x, b)$ (as a byproduct of computing $P(R \mid x, v)$), see Eq. (1)). As shown in [8,12–14], a very appropriate way to obtain this relevance probability is by using previously trained optical and language models, similar to those employed in handwritten text recognition. In this work, as in [4], we use a Convolutional Recurrent Neural Network (CRNN) [7] for character optical modeling and a 6-gram character language model. Details about the (meta-)parameter settings employed for training/decoding with these models and producing the required relevance probabilities can be seen in [4,7].

Most of the experiments have been carried out with the TestVal set and in all of them, performances are measured at the page level. For the 89 pages of the entire Test set, the process requires 72h on a GeForce GTX 1080 and a 2 core Intel Core i3-6100 CPU. This time can be drastically reduced in several ways, but we believe that this is a secondary goal, since our main aim is to prove that LSF-PKWSI can bring competitive results.

Regarding the pre-processing of the page images (required to obtain an estimate of the line height $\hat{h}$) as described in Sect. 3.2, we used a kernel size of 25 for the Gaussian blur, a local thresholding covering areas of 201 × 201 pixels and a value of 20 for the RLSA parameter. We obtained the two best line-height estimates using a fast Fourier transform.

Then, we optimized the vertical sampling parameters (Height Factor and Vertical Sampling Rate). Table 2 shows the results we obtained on the TestVal set by using LSF-PKWSI without any consolidation of the BBs. Despite the fact that after a value of 2, the Vertical Sampling Rate does not seem to impact a lot the AP, we believe that it matters more after the consolidation process. Based on the above reason, the AP obtained and the resources usage, we extracted lines of height $1.25 \cdot \hat{h}$, with a Vertical Sampling Rate of 3 windows per unit of height $\hat{h}$. We leave as future works the continuation of these experiments, as it might have an important impact.

Table 2. AP measured when changing the parameters of the line extraction process. Both Height Factor and Vertical Sampling Rate depend of the estimated line height $\hat{h}$. The results are obtained on the TestVal set at the page level.

Height factor	Vertical sampling rate			
	1	2	3	4
1.0	–	–	0.658	–
1.25	0.641	0.712	**0.714**	0.722
1.5	–	–	0.660	–

Then, we tuned the vertical sampling parameters. First, we dealt with the parameters for Algorithm 1. For θ, which is the minimum fraction of pixels that two BBs have to share to be considered as overlapping BBs, we used a value of 0.5. Concerning the parameter τ, we selected a value of 0.2. It is used in Algorithm 1 as the minimum fraction of BBs in a group g a new BB β has to overlap with to be inserted in g.

Finally, for the value of the customized sigmoid, $F(x) = 1 / (1 + e^{-ax+b})$ we used $a = 8$ and $b = 2.75$. It allows keeping most of the BBs overlapping with each other, while lowering the probability of lone BBs.

Table 3. Comparison between the results obtained with line-based PKWSI method (Baseline), and our method before (LSF-PKWSI Raw) and after being consolidated (LSF-PKWSI Consolidated) as described in Sect. 3.3.

Experiment	TestVal		Test	
	mAP	AP	mAP	AP
Baseline	0.90	0.84	0.73	0.73
LSF-PKWSI Raw	0.88	0.72	0.62	0.54
LSF-PKWSI Consolidated	**0.89**	**0.76**	**0.64**	**0.60**

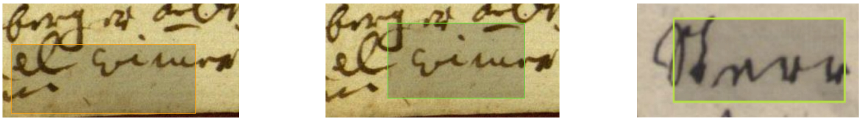
4.5 Results

Table 3 shows a comparison of our new approach and the previous results obtained with manually extracted lines. We compare the line-based PKWSI (Baseline) with our approach before consolidation (i.e. with overlapping BBs, referred to as LSF-PKWSI Raw) and after the consolidation, as described in Sect. 3.3 (LSF-PKWSI Consolidated). The consolidated LSF-PKWSI test set results are 0.13 points behind the baseline in terms of AP and 0.09 points in terms of mAP.

The LSF-PKWSI approach obtained three times more spots on the test set than the baseline method, with many of them being false-positives. We believe that it could be explained by the number of lines extracted and also their width.

Using a relevance probability threshold for which precision is equal to recall, our new approach obtained 17% less true-positives (*hits*) and 44% more false-positives than the baseline. Therefore, to further improve the LSF-PKWSI results, we should focus on reducing the number of false-positives. However, at this point, the number of spots detected by the baseline is similar to those detected by LSF-PKWSI (less than 0.1% difference). This significant improvement with respect to the previous three-fold difference shows that, although we do obtain many raw spots, most of them are false-positives with a very low probability. They can be easily filtered out for queries.

It is worth noting that our approach is often capable to outperform the baseline. In cases where the provided text-lines were not correct (e.g. because of mistakes in the ground truth), the baseline either fails or obtains low probabilities (Fig. 4a), whereas LSF-PKWSI is able to obtain a correct spot with a high probability (Fig. 4b and c).



(a) $P(R \mid \mathsf{Wimer}) = 0.027$ (Baseline)

(b) $P(R \mid \mathsf{Wimer}) = 0.99$ (LSF-PKWSI)

(c) $P(R \mid \mathsf{Sterr}) = 0.94$ (LSF-PKWSI)

Fig. 4. Two examples where LSF-PKWSI performs better than the baseline. In (b) LSF-PKWSI leads to better score and BB compared to the baseline (a), where the provided text line was wrong. (c) is an example where LSF-PKWSI spotted the correct keyword (Sterr) while the baseline failed.

Moreover, with our hands-on experience using the demonstrators (links in Sect. 5), we feel that the practical performance is really better than the results of Table 3 suggest. Hence, we believe that improving the results by a fair amount should not be difficult.

5 Conclusions

A new, line segmentation-free approach to probabilistic keyword spotting and indexing has been introduced and tested in a series of experiments with a difficult dataset of historical handwritten images. Despite having results a bit lower than the baseline, they are still promising and we are confident about their possible improvements. To allow practical testing of three different approaches (baseline, raw and consolidated spot BBs) three demonstrators have been implemented for the TestVal set and are publicly available: baseline,[5] where text lines were manually detected; line segmentation-free, raw spot BBs,[6] without post-processing; and line segmentation-free, consolidated by merging overlapping BBs[7]. A demonstrator for the full Test set is also available.[8]

The similar idea can be used for other applications such as probabilistically indexing text in natural scene images which may include text regions, or even to probabilistically index other objects of interest.

Future works will be devoted to improve the consolidation of word BBs. Especially, we expect to improve the results by sticking to the formal development and avoiding heuristics and tunable parameters as much as possible. In addition, we also plan to carry out experiments to measure precision-recall performance at the word BB level, rather than the rough, page-image level assessment reported in this paper. Moreover, we want to compare our approach with state-of-the-art methods using automatic text-line extraction, and also with full segmentation free approaches instead of manually extracted text lines as in the current baseline results. Lastly, we might also consider improving the computing time and memory taken by our approach.

Acknowledgements. This work was partially supported by the BBVA Foundation through the 2017–2018 and 2018–2019 Digital Humanities research grants “Carabela” and “HistWeather – Dos Siglos de Datos Climáticos”, and by EU JPICH project “HOME – History Of Medieval Europe” (Spanish PEICTI Ref. PCI2018-093122). This is also partially supported by the École Normale Supérieure de Rennes and Univ Rennes through a funded internship grant.

References

1. Bluche, T., et al.: Preparatory KWS experiments for large-scale indexing of a vast medieval manuscript collection in the HIMANIS project. In: 14th ICDAR (2017)
2. Diem, M., Kleber, F., Fiel, S., Grüning, T., Gatos, B.: CBAD: ICADR 2017 competition on baseline detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1355–1360. IEEE (2017)
3. Giotis, A.P., Sfikas, G., Gatos, B., Nikou, C.: A survey of document image word spotting techniques. Pattern Recogn. **68**, 310–332 (2017)
4. Lang, E., Puigcerver, J., Toselli, A.H., Vidal, E.: Probabilistic indexing and search for information extraction on handwritten German parish records. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 44–49, August 2018
5. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)

[5] http://prhlt-carabela.prhlt.upv.es/passau-LSF-TestVal/Baseline.

[6] http://prhlt-carabela.prhlt.upv.es/passau-LSF-TestVal/Raw.

[7] http://prhlt-carabela.prhlt.upv.es/passau-LSF-TestVal/Consolidated.

[8] http://prhlt-carabela.prhlt.upv.es/passau-LSF/.

6. Perronnin, F., Liu, Y., Renders, J.M.: A family of contextual measures of similarity between distributions with application to image retrieval. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2358–2365, June 2009
7. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: Proceedings of 14th ICDAR (2017)
8. Puigcerver, J.: A probabilistic formulation of keyword spotting. Ph.D. thesis, Universitat Politècnica de València (2018)
9. Robertson, S.: A new interpretation of average precision. In: Proceedings of the International Conference on research and Development in Information Retrieval, SIGIR 2008, pp. 689–690 (2008)
10. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recogn. **33**(2), 225–236 (2000)
11. Toselli, A.H., Puigcerver, J., Vidal, E.: Two methods to improve confidence scores for lexicon-free word spotting in handwritten text. In: Proceedings 15th ICFHR, pp. 349–354 (2016)
12. Toselli, A.H., Vidal, E., Romero, V., Frinken, V.: HMM word graph based keyword spotting in handwritten document images. Inf. Sci. **370–371**, 497–518 (2016)
13. Toselli, A.H., Vidal, E., Puigcerver, J., Noya-García, E.: Probabilistic multi-word spotting in handwritten text images. Pattern Anal. Appl. **22**, 23–32 (2018)
14. Vidal, E., Toselli, A.H., Puigcerver, J.: A probabilistic framework for lexicon-based keyword spotting in handwritten text images. Technical report, arXiv (2017)
15. Wahl, F.M., Wong, K.Y., Casey, R.G.: Block segmentation and text extraction in mixed text/image documents. Comput. Graph. Image Process. **20**(4), 375–390 (1982)
16. Zhu, M.: Recall, Precision and Average Precision. Working Paper 2004-09 Department of Statistics & Actuarial Science - University of Waterloo (2004)