Annotation-free keyword spotting in historical Vietnamese manuscripts using graph matching

Anna Scius-Bertrand[1,2,3], Linda Studer[1,2], Andreas Fischer[1,2], and Marc Bui[3]

[1] iCoSys, HES-SO, Fribourg, Switzerland
{anna.scius-bertrand,linda.studer,andreas.fischer}@hefr.ch
[2] DIVA, University of Fribourg, Switzerland
[3] Ecole Pratique des Hautes Etudes, Paris, France
marc.bui@ephe.psl.eu

Abstract. Finding key terms in scanned historical manuscripts is invaluable for accessing our written cultural heritage. While keyword spotting (KWS) approaches based on machine learning achieve the best spotting results in the current state of the art, they are limited by the fact that annotated learning samples are needed to infer the writing style of a particular manuscript collection. In this paper, we propose an annotation-free KWS method that does not require any labeled handwriting sample but learns from a printed font instead. First, we train a deep convolutional character detection system on synthetic pages using printed characters. Afterwards, the structure of the detected characters is modeled by means of graphs and is compared with search terms using graph matching. We evaluate our method for spotting logographic Chu Nom characters on the newly introduced Kieu database, which is a historical Vietnamese manuscripts containing 719 scanned pages of the famous Tale of Kieu. Our results show that search terms can be found with promising precision both when providing handwritten samples (query by example) as well as printed characters (query by string).

Keywords: Historical documents · Keyword spotting · Annotation-free · Kieu database · Chu Nom characters · Character detection · Handwriting graphs · Hausdorff edit distance.

1 Introduction

Despite strong progress in the past two decades, automated reading of historical handwriting remains a challenging open problem in the field of pattern recognition [5]. One of the main obstacles is the large variety of scripts, languages, writing instruments, and writing materials that need to be modeled. The current state of the art follows the paradigm of learning by examples using machine learning techniques, leading to high accuracy for automatic transcription under the condition that a large amount of annotated handwriting samples are available for some manuscript collections. Especially for ancient languages, which are only known by few human experts, obtaining annotated samples is time-consuming and thus hinders an automatic transcription of historical documents at large scale.

As an alternative to producing a full transcription, keyword spotting (KWS) has been proposed to find specific search terms in historical manuscripts [6]. Although the number of annotated learning samples can be reduced when focusing only on a few search terms, it has become evident over the past decade that the learning by examples paradigm is still needed to achieve high precision for KWS. Prominent examples include the use of annotated word images for training word embeddings with deep convolutional neural networks (CNN) [13] and the use of annotated text line images for training character models with hidden Markov models (HMM) or long short-term memory networks (LSTM) [15].

For historical Vietnamese manuscripts, transfer learning from printed fonts to handwritten Chu Nom characters has recently been shown to be feasible for automatic transcription [9], reducing the amount of annotated handwriting samples required during training. For the tasks of character detection (localizing characters, without recognition) and transcription alignment (localizing characters of a known transcription), it was even possible to completely replace handwritten learning samples with printed ones by means of self-calibration and unsupervised clustering algorithms, leading to fully annotation-free detection [11] and alignment [12] methods, respectively.

In this paper, we go a step further and investigate to what extent annotation-free KWS is feasible for historical Vietnamese manuscripts when considering only printed fonts during training. The proposed method consists of two components. First, a deep convolutional neural network is trained on synthetic printed pages to detect characters. Secondly, a handwritten sample of the search term (query by example) or a printed sample (query by string) is compared with the detected characters to retrieve similar instances. Such an approach is highly valuable for an initial exploration of historical document collections, because it operates directly on the scanned page images and only requires printed fonts of the Chu Nom characters to make the handwritten content searchable. Furthermore, it allows to gather basic statistics about the document collection (number of columns, characters, etc.) and facilitates ground truth creation, especially the annotation of handwritten characters.

To compare images of the logographic Chu Nom characters, we leverage methods from structural pattern recognition [2, 14] to focus on the core structure of the writing and thus supporting the transfer from printed to handwritten characters. Specifically, we consider keypoint graphs extracted from character skeleton images [3] and compare them efficiently by means of the Hausdorff edit distance (HED) [4], a quadratic-time approximation (lower bound) of the graph edit distance. A similar approach has already proven successful for signature verification [8] and for spotting handwritten words in Latin script [1]. When compared with [1], the proposed method has two main advantages. First, it is segmentation-free, i.e. it does not require pre-segmented word images. Secondly, it is fully annotation-free, i.e. even the parameters of graph extraction and HED are optimized on a set of printed characters instead of using a human-annotated validation set. It is also noteworthy that it is the first time that this structural approach is investigated for non-Latin characters. Hopefully, the publication of

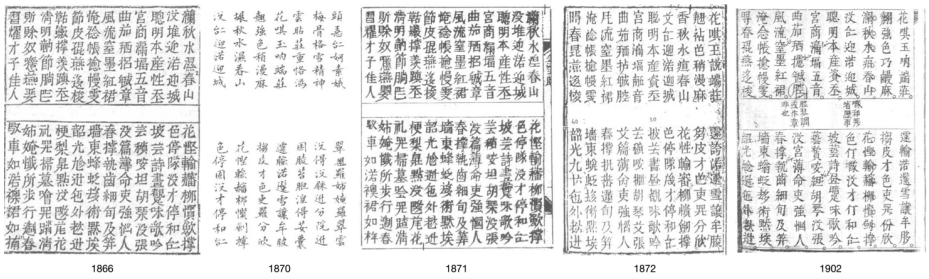

Fig. 1. Kieu database

the Chu Nom character dataset from the books of Kieu used in this work will be a valuable contribution to the community.

The remainder of the paper is structured as follows. Section 2 describes the Kieu database, a newly introduced research dataset used for experimental evaluation. Section 3 details the proposed KWS method, Section 4 reports the experimental results, and Section 5 draws some conclusions.

2 Kieu database

The Kieu database consists of five books with versions of the Tale of Kieu, one of the most famous stories of Vietnam. The versions date from 1866, 1870, 1871, 1872 and 1902, respectively, and are written in Chu Nom. This logographic script was used in Vietnam from the 10th to the 20th century before it was replaced with a Latin-based alphabet. Figure 1 illustrates an example page for each of the five books. The images and transcriptions used to create the database were kindly provided by the Vietnamese Nom Preservation Foundation[4].

We used the transcription alignment method from [12] to align the transcriptions with the page images and manually verified the results, discarding pages that contained errors. The resulting Kieu database[5] consists of 719 pages with a total of 97,152 characters annotations, i.e. bounding boxes and unicode encodings, for 13,207 unique characters. The images retrieved from the webpage have a relatively low resolution, introducing an additional challenge for KWS when combined with the large number of unique characters.

3 Annotation-free keyword spotting (KWS)

An overview of the proposed annotation-free KWS method is provided in Figure 2. First, a YOLO-based deep convolutional character detection system is trained on synthetic pages with printed Chu Nom characters. Therefore, the training does not require human annotations of real page images. The trained

[4] http://www.nomfoundation.org
[5] Available here: https://github.com/asciusb/Kieu-database

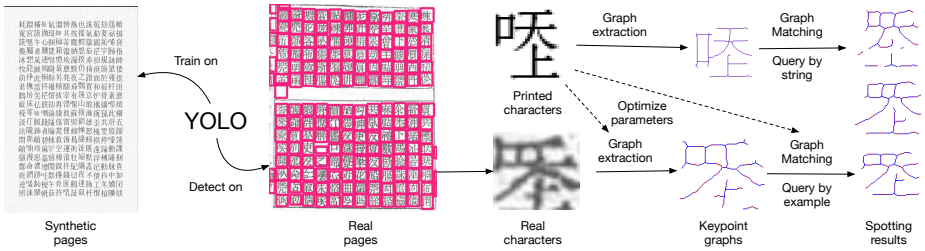

Fig. 2. Overview of the workflow of our proposed keyword spotting method

network is then applied to real page images to extract characters. Typical errors at this stage include missing character parts and false positives, especially in the border regions as illustrated in Figure 2.

Next, the character images are represented by means of keypoint graphs, and HED-based graph matching with query graphs is performed to retrieve the most similar samples. Both the parameters of graph extraction and graph matching are optimized with validation experiments that take only graphs from printed Chu Nom characters into account. Therefore, the parameter optimization does not require human annotations of real character images.

The query graphs can be obtained either from real character images (query by example) or printed characters (query by string). The former approach requires the user to manually select one or several templates of the query term on real page images. The latter allows to enter the search term in form of plain text and is thus completely independent of the scanned manuscripts.

In the following, we provide more details on the different steps of our method.

3.1 Synthetic dataset creation

We use five fonts to generate images of printed characters, namely Nom Na Tong Light, Han Nom A et B, Han Nom Gothic, Han Nom Minh, and Han Nom Kai. The synthetic training pages are created by writing columns of random printed Chu Nom characters on a white background surrounded by a black border. Afterwards, a series of image transformations are applied for data augmentation, including changes in brightness, applying Gaussian blur, and adding salt and pepper noise. An example is illustrated in Figure 2.

3.2 Character detection

For character detection, we consider the YOLO [10] (You Only Look Once) architecture for one-stage object detection with deep convolutional neural networks. More specifically, we employ the latest version YOLOv5 [7], which integrates several improvements over the original architecture that are important for character detection, including a higher resolution, multi-scale features, and multi-scale anchor boxes that allow to detect also small characters in low-resolution images.

3.3 Graph extraction

Following the procedure proposed in [3], keypoint graphs are extracted from the detected character images, real and synthetic, as illustrated in Figure 2. First, a local edge enhancement by means of a Difference of Gaussians (DoG) is performed before applying a global threshold for binarization. Afterwards, the binary image is thinned to one pixel width and three types of keypoints are located on the skeleton image: endpoints, junction points, and a random point on circular structures. To obtain a labeled keypoint graph $g = (V, E)$, all keypoints are added to the set of nodes V with coordinate labels (x, y). Afterwards, the skeleton is sampled at distance D pixels between the keypoints, adding further nodes to V. Unlabeled edges are added to the set of edges E for each pair of nodes that is directly connected on the skeleton.

We introduce another parameter for graph extraction to cope with low-resolution images, i.e. a scaling factor $S > 1$, which is applied to resize (upscale) the character images at the beginning of the process, before applying DoG and binarization. This super-resolution allows to insert even more nodes between two keypoints than the number of pixels in the original image, thus emphasizing the structure even for very small strokes, which are important to distinguish similar Chu Nom characters.

3.4 Graph matching

We use the graph edit distance (GED) to compare Chu Nom characters based on their keypoint graphs. GED calculates the minimum transformation cost between two graphs with respect to cost of node deletion $(u \to \epsilon)$, node insertion $(\epsilon \to v)$, node label substitution $(u \to v)$, edge deletion $(s \to \epsilon)$, and edge insertion $(\epsilon \to t)$. To overcome the computational constraints of exact the GED, which is NP-complete, we use the Hausdorff edit distance (HED) [4] to compute an approximation (lower bound) in quadratic time:

$$HED_c(g_1, g_2) = \sum_{u \in V_1} \min_{v \in V_2 \cup \{\epsilon\}} f_c(u, v) + \sum_{v \in V_2} \min_{u \in V_1 \cup \{\epsilon\}} f_c(u, v) \tag{1}$$

where c is the cost function for the edit operations and $f_c(u, v)$ the cost for assigning node u to node v, taking into account their adjacent edges as well.

We use the Euclidean cost function, i.e. constant costs c_V and c_E

$$\begin{aligned} c(u \to \epsilon) = c(\epsilon \to v) = c_V \\ c(s \to \epsilon) = c(\epsilon \to t) = c_E \end{aligned} \tag{2}$$

for node and edge deletion and insertion, and the Euclidean distance

$$c(u \to v) = ||(x_u, y_u) - (x_v, y_v)|| \tag{3}$$

for node label substitution.

3.5 Keyword spotting (KWS)

To create a keyword, one or several template graphs $\mathcal{T} = \{t_1, \ldots, t_m\}$ are extracted either from real character images or from synthetic images using printed fonts. Then, for spotting a keyword, the template graphs are compared with each character graph $c_1, \ldots, c_n$ in the document collection, and scored according to the minimum HED, such that the most similar character have the lowest score.

$$score(c_i) = \min_{t \in \mathcal{T}} d(c_i, t) \tag{4}$$

4 Experimental evaluation

To evaluate the proposed annotation-free KWS method, we conduct a series of experimental evaluations on the five books of the Kieu database. In a first step, we optimize the different meta-parameters without using human annotations. In a second step, we perform an an ablation study to explore the limitations of the method by gradually rendering the task more difficult, while keeping the parameters fixed.

For the performance evaluation, the character graphs of the document collection are sorted according to Equation 4 to compute recall and precision for each possible score threshold. For each keyword, the average precision is computed (AP) and the mean average precision (mAP) is considered as the final performance measure when spotting N keywords:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i. \tag{5}$$

4.1 Task setup and parameter optimization

We compare the following three scenarios in terms of what data (real vs. synthetic font characters) is used for parameter and model selection:

- **Annotated-only**. As a baseline approach, we use the first 5 pages of each book to select keyword templates, the next 5 pages as a validation set to optimize the parameters, and the next 5 pages as a test set to evaluate the spotting performance with the best parameter configuration. All keywords are considered that have at least 3 templates and that appear at least once in the validation and test set, respectively. Table 1 shows the size of the test sets for each book and the number of keywords that are spotted.
- **Font-validation**. To avoid the need for expert annotations for parameter optimization, we replace the validation set with synthetic printed characters. 20 random characters are printed in 5 Chu Nom fonts to obtain keyword templates. Another 900 random non-keyword characters are printed with a random font, leading to a synthetic validation set with a total of 1,000 characters.

Table 1. Overview of the test sets of the Kieu database.

	1866	**1870**	**1871**	**1872**	**1902**
Pages	5	5	5	5	5
Characters	840	490	840	700	700
Keywords	80	36	93	68	63

Table 2. Chosen meta-parameters after optimization.

Parameter	**Annotated-only**	**Font-validation**
Scale	3	4
Norm	z-score	z-score
Node cost c_V	1.0	0.1
Edge cost c_E	1.0	2.0

– **Annotation-free**. To fully avoid expert annotations, we use the best parameters obtained by font-validation and perform an automatic character detection on the 5 test pages instead of using the ground truth bounding boxes. This setup corresponds to the KWS method proposed in Section 3.

The parameters optimization and selection is performed as follows:

– **Character detection parameters**. One fixed setup is tested for YOLO-based character detection. We use the default configuration[6] of the medium-sized YOLOv5m model with COCO-pretrained weights, and an initial learning rate of 0.0032. A total of 30,000 synthetic pages using printed Chu Nom characters (see Section 3.1 and 3.2) are used for training over 25 epochs until convergence.
– **Graph extraction parameters**. Six scaling factors $S \in \{1, 2, \ldots, 6\}$ are tested in combination with a fixed node distance of $D = 3$ pixels to explore different degrees of super-resolution (see Section 3.3). Furthermore, three setups are tested for normalizing the node labels, i.e. using the raw coordinates, centering to zero mean, and normalizing to zero mean and unit variance (z-score).
– **Graph matching parameters**. 25 combinations of node and edge costs (c_V, c_E) are tested (see Section 3.4). For the raw and centered coordinates, we investigate the range of $c_V, c_E \in \{1.0, 5.0, 10.0, 15.0, 20.0\}$ and for the z-score normalized ones we consider $c_V, c_E \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$.

4.2 Results

Table 2 indicates the optimal parameter values obtained with respect to the mAP on the annotated validation set (annotation only), and on the synthetic

[6] github.com/ultralytics/yolov5, commit cc03c1d5727e178438e9f0ce0450fa6bdbbe1ea7

Table 3. Mean average precition (mAP) on the test set after parameter optimization. The best results are highlighted in bold font.

	1866	**1870**	**1871**	**1872**	**1902**	**Average**
Annotated-only	**0.79**	0.93	0.67	0.68	**0.74**	0.76
Font-validation	**0.79**	**0.97**	**0.73**	**0.70**	**0.74**	**0.78**
Annotation-free	**0.79**	0.94	**0.73**	0.66	0.71	0.77

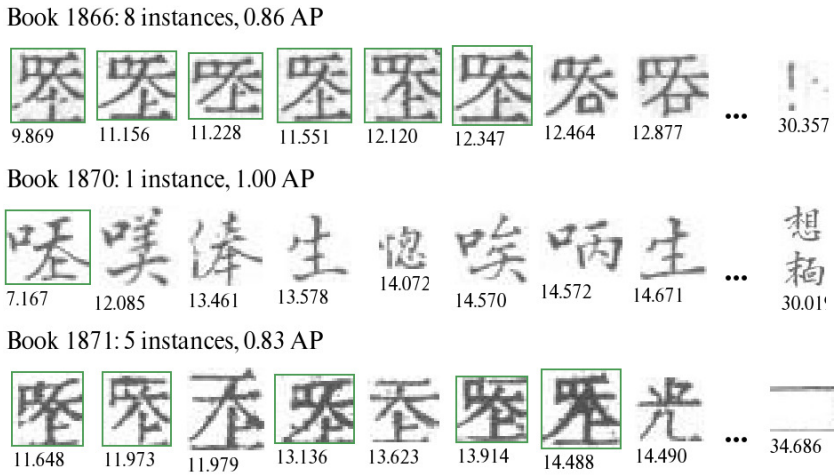

Fig. 3. Exemplary spotting results for the character "word". Correct retrieval results are marked in green. The top-8 results as well as the character with the worst score are shown, ranked according to the Hausdorff edit distance (indicated below).

validation set with printed characters (font-validation). In both cases, a super-resolution is preferred, highlighting details of small strokes in the Chu Nom characters by inserting additional nodes. Also, normalizing the coordinates to zero mean and unit variance is beneficial (z-score), removing small variations in position and scale among the characters. The optimal values for node and edge deletion and insertion are a different when optimizing on printed characters instead of real ones. The font-validation suggests an emphasis on the edges, thus increasing the importance of the character structure.

Table 3 shows the mAP results achieved on the test set for each book individually, and on average over all books. For annotated-only, we used the 5 annotated validation pages of book 1866 for parameter optimization. When compared with font-validation, we observe a slight overfitting to this book, whereas the font-validation parameters generalize better to all five books, achieving 0.78 mAP. When using the same parameters as for font-validation, but detecting the characters automatically instead of using the ground truth bounding boxes, we report between 0.66 mAP (book 1872) and 0.94 mAP (book 1870), resulting in 0.77 mAP on average for fully annotation-free KWS.

Table 4. Ablation study. Mean average precision (mAP) on the test set for different keyword spotting tasks with increasing difficulty.

Query by	# Pages	# Templ.	Book version					Average
			1866	**870**	**1871**	**1872**	**1902**	
Example	5	3	0.79	0.94	0.73	0.66	0.71	0.77
	10	3	0.77	0.92	0.73	0.60	0.74	0.75
	20	3	0.73	0.92	0.70	0.56	0.74	0.73
	5	3	0.77	0.93	0.71	0.63	0.66	0.74
	5	1	0.70	0.92	0.64	0.54	0.62	0.68
String	5	1	0.65	0.73	0.59	0.61	0.59	0.63

Figure 3 illustrates exemplary spotting results on the test sets when searching for the character "word". For each book, the number of instances of the character and the achieved average precision (AP) is reported. For example, in book 1870, the only present instance is retrieved with the smallest HED, thus leading to a perfect spotting result of 1.0 AP. The retrieved character with the lowest HED is typically a detection error, such as elements of the page border, partial characters, merged characters, or page background. Thus, errors at the character detection stage do not have a negative impact on the KWS.

4.3 Ablation study

After obtaining strong results with the initial setup of three real keyword templates, we gradually increase the difficulty of the KWS task in an ablation study. Table 4 shows the results. When increasing the number of test pages from 5 to 20, we include more characters, which may be similar to the keywords. Nevertheless, the spotting performance is only reduced by 0.04 mAP on average. When using only a single keyword template, however, the decrease of 0.09 mAP is already more significant.

Remarkably, our proposed method is also able to perform query by string KWS using printed templates of the keyword, highlighting the effectiveness of the structural representation. However, there is a decrease in mAP of 0.14, illustrating the need to model the variability of the handwriting in order to achieve the best results.

5 Conclusions

We have introduced a new approach to spot keywords in historical Vietnamese manuscripts, which is directly applicable to a collection of scanned page images without requiring human annotations. The structural pattern recognition method for KWS is able to achieve a promising performance of 0.77 mAP for query by example and 0.63 mAP for query by string on the Kieu database. Thus,

our method is ideally suited for an initial exploration of manuscript collections, especially because of its ability to perform knowledge transfer from printed to handwritten characters.

Future research includes improvements of the spotting method by means of geometric deep learning for graph-based representations, data augmentation for modeling variations in the handwriting, and a possible generalization of the method to other scripts and languages.

References

1. Ameri, M.R., Stauffer, M., Riesen, K., Bui, T.D., Fischer, A.: Graph-based keyword spotting in historical manuscripts using Hausdorff edit distance. Pattern Recognition Letters **121**, 61–67 (2019)
2. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. Journal of Pattern Recognition and Artificial Intelligence **18**(3), 265–298 (2004)
3. Fischer, A., Riesen, K., Bunke, H.: Graph similarity features for HMM-based handwriting recognition in historical documents. In: Proc. Int. Conf. on Frontiers in Handwriting Recognition. pp. 253–258 (2010)
4. Fischer, A., Suen, C.Y., Frinken, V., Riesen, K., Bunke, H.: Approximation of graph edit distance based on Hausdorff matching. Pat. Rec. **48**(2), 331–343 (2015)
5. Fischer, A., Liwicki, M., Ingold, R. (eds.): Handwritten historical document analysis, recognition, and retrieval — state of the art and future trends. World Scientific (2020)
6. Giotis, A.P., Sfikas, G., Gatos, B., Nikou, C.: A survey of document image word spotting techniques. Pattern Recognition **68**, 310–332 (2017)
7. Jocher, G., et al.: ultralytics/yolov5: v4.0 - nn.silu() activations, weights & biases logging, pytorch hub integration (2021). https://doi.org/10.5281/ZENODO.4418161
8. Maergner, P., Pondenkandath, V., Alberti, M., Liwicki, M., Riesen, K., Ingold, R., Fischer, A.: Combining graph edit distance and triplet networks for offline signature verification. Pattern Recognition Letters **125**, 527–533 (2019)
9. Nguyen, K.C., Nguyen, C.T., Nakagawa, M.: Nom document digitalization by deep convolution neural networks. Pattern Recognition Letters **133**, 8–16 (2020)
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 779–788 (2016)
11. Scius-Bertrand, A., Jungo, M., Wolf, B., Fischer, A., Bui, M.: Annotation-free character detection in historical Vietnamese stele images. In: Proc. 16th Int. Conf. on Document Analysis and Recognition (ICDAR). pp. 432–447 (2021)
12. Scius-Bertrand, A., Jungo, M., Wolf, B., Fischer, A., Bui, M.: Transcription alignment of historical Vietnamese manuscripts without human-annotated learning samples. Applied Sciences **11**(11), 4894 (2021)
13. Sudholt, S., Fink, G.A.: PHOCNet: A deep convolutional neural network for word spotting in handwritten documents. In: Proc. 15th Int. Conf. on Frontiers in Handwriting Recognition. pp. 277–282 (2016)
14. Vento, M.: A one hour trip in the world of graphs, looking at the papers of the last ten years. In: Proc. Int. Workshop on Graph-Based Repr. pp. 1–10 (2013)
15. Vidal, E., Toselli, A.H., Puigcerver, J.: A probabilistic framework for lexicon-based keyword spotting in handwritten text images. CoRR abs/2104.04556 (2021)