Pattern Recognition 81 (2018) 240–253

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Keyword spotting in historical handwritten documents based on graph matching

Michael Stauffer[a,d,*], Andreas Fischer[b,c], Kaspar Riesen[a]

[a] University of Applied Sciences and Arts Northwestern Switzerland, Institute for Information Systems, Olten 4600, Switzerland
[b] University of Fribourg, Department of Informatics, Fribourg 1700, Switzerland
[c] University of Applied Sciences and Arts Western Switzerland, Institute of Complex Systems, Fribourg 1705, Switzerland
[d] University of Pretoria, Department of Informatics, Pretoria, South Africa

ARTICLE INFO

Article history:
Received 4 September 2017
Revised 6 January 2018
Accepted 2 April 2018
Available online 6 April 2018

Keywords:
Handwritten keyword spotting
Graph representation
Bipartite graph matching
Ensemble methods

ABSTRACT

In the last decades historical handwritten documents have become increasingly available in digital form. Yet, the accessibility to these documents with respect to browsing and searching remained limited as full automatic transcription is often not possible or not sufficiently accurate. This paper proposes a novel reliable approach for template-based keyword spotting in historical handwritten documents. In particular, our framework makes use of different graph representations for segmented word images and a sophisticated matching procedure. Moreover, we extend our method to a spotting ensemble. In an exhaustive experimental evaluation on four widely used benchmark datasets we show that the proposed approach is able to keep up or even outperform several state-of-the-art methods for template- and learning-based keyword spotting.

1. Introduction

In order to bridge the gap between availability and accessibility of ancient handwritten documents *handwriting recognition* is often employed for an automatic and complete transcription. In the case of historical documents this process is inherently an *offline* task, and as such, more complex than *online* handwriting recognition where temporal information is available [1]. Moreover, the recognition rates of handwriting recognition systems applied to ancient documents is often negatively affected by both the degenerative conservation state of scanned documents [2] and different writing styles [3].

In order to overcome the prior obstacles of automatic full transcriptions of historical handwritten documents, *Keyword Spotting (KWS)* as a more error-tolerant, flexible, and suitable approach has been proposed [4–7]. KWS refers to the task of retrieving any instance of a given query word in a certain document. The concept of KWS was originally proposed for speech documents [8] and later adapted for printed [9] and handwritten documents [4].

To date, KWS is still highly relevant in different application domains. This is particularly due to the global trend towards digitalisation of paper-based archives and libraries in both private and public institutions. Clearly, KWS techniques can make these documents accessible for searching and browsing [5]. Similar to handwriting recognition, textual KWS can be divided into *online* and *offline* KWS. The focus of this paper is on historical documents, and thus, offline KWS, referred to as KWS from now on, can be applied only.

Most of the KWS methodologies available are based on *template-based* or *learning-based* algorithms (similar to the corresponding subfields in handwriting recognition). Early approaches of template-based KWS are based on pixel-by-pixel matchings of word images [4] by either Euclidean distance measures or affine transformations by the *Scott and Longuet–Higgins* algorithm [10]. More elaborated and error-tolerant approaches to template-based KWS are based on the matching of feature vectors that numerically describe certain characteristics of the word images like projection profiles [5,11,12], gradients [11], contours [13], or geometrical characteristics [14]. Also more generic image feature descriptors have been used like *Histograms of Oriented Gradients* [15–17], *Local Binary Patterns* [17,18], or *Deep Learning* features [19], to name just a few. Regardless the features actually employed, *Dynamic Time Warping (DTW)* is the most frequently used algorithm for matching two sequences of features in KWS [12–16,19,20].

Learning-based KWS is based on statistical models that have to be trained *a priori* on a (relatively large) training set of word or character images. Many approaches of learning-based KWS are based on *Hidden Markov Models (HMM)* [6,7,21–26]. Early approaches are based on *generalised Hidden Markov Models* that are

[*] Corresponding author.
E-mail address: michael.stauffer@fhnw.ch (M. Stauffer).

https://doi.org/10.1016/j.patcog.2018.04.001
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

trained on character images, i.e. images of Latin [21] or Arabic [23] characters. However, character-based approaches are negatively affected by an error-prone segmentation step [7]. More elaborated approaches rely on feature vectors of word images [22], for example by means of *Continuous-HMM* [6] or *Semi-Continuous-HMM* [6], i.e. HMMs with a shared set of *Gaussian Mixture Models*. Furthermore, the use of a *Fisher Kernel* has been employed in conjunction with HMMs in [24], while a line-based and lexicon-free HMM-approach is proposed in [7]. Recently, HMMs have been used in combination with *Bag-of-Features* [25] or *Deep Neural Networks* [26].

Further learning-based approaches are based on *Recurrent Neural Networks* [20,27], *Support Vector Machines (SVM)* [28–30], or *Latent Semantic Analysis* [31–33]. Moreover, we observe a clear shift towards *Convolutional Neural Network (CNN)* in the last years [34–38]. In most cases, CNNs are used to learn a certain word string embedding like *Pyramid Histogram of Characters (PHOC)* [35–38] or *Discrete Cosine Transform of Words* [36–38] that allows the retrieval of visual and textual queries in the same feature space.

It is known that template-based matching algorithms generally result in a lower recognition accuracy when compared to learning-based approaches. Yet, this advantage is accompanied by a loss of flexibility, which is due to the need for learning the parameters of the actual model. In particular, learning-based methods are depending on the acquisition of labelled training data by means of human experts. This is a costly and time-intensive process, especially in case of handwritten historical documents. In contrast with learning-based approaches, template-based algorithms are independent from both the actual representation formalism and the language of the underlying document. Thus, only a single instance of a keyword image is needed for the whole retrieval process.

1.1. Related work

The vast majority of KWS algorithms are based on statistical representations of words by certain numerical features (regardless whether template- or learning-based approaches are used). However, in recent years a clear tendency towards structural pattern representation formalisms can be observed in various domains [39,40]. *Structural Pattern Recognition* is based on more sophisticated data structures than feature vectors such as strings, trees, or graphs (whereby strings and trees can be seen as special cases of graphs). Graphs are, in contrast with feature vectors, flexible enough to adapt their size to the size and complexity of individual patterns. Moreover, graphs are capable to represent binary relationships that might exist in different subparts of the underlying patterns.

In the last four decades various procedures for evaluating the dissimilarity of graphs, commonly known as *graph matching*, have been proposed [41,42]. Although graphs gained noticeable attention in various fields, we observe only limited attempts where graphs have been used for the analysis and recognition of handwriting [43–45]. This is particularly interesting as graphs offer a natural and comprehensive way to represent handwritten characters or words. Moreover, in the last decade substantial progress has been made in speeding up different graph matching algorithms [42]. These facts build the main motivation of the present paper that researches the benefits of graph and template-based KWS.

A first approach to graph-based KWS has been proposed in [43], where certain keypoints are represented by nodes, while edges are used to represent strokes between these keypoints. The matching of words is then based on two separate procedures. First, assignment costs between all pairs of connected components (represented by graphs) are computed by means of a bipartite graph matching algorithm [46]. Second, optimal assignment costs between all pairs of connected components are found by means of a DTW implementation. The same matching procedure is improved by a so-called *coarse-to-fine approach* in [47].

Another framework for graph-based KWS has been introduced in [44], where nodes represent prototype strokes (so-called invariants), while edges are used to connect nodes which stem from the same connected component. The same matching procedure as in [47] is finally used for computing graph dissimilarities.

A third graph-based KWS approach has been proposed in [45], where nodes represent prototype stokes (so-called graphemes), while edges are used to represent the connectivity between graphemes. The matching is based on a coarse-to-fine approach. Formally, potential subgraphs of the query graph are determined first. These subgraphs are subsequently matched against a query graph by means of a similar graph matching procedure as used in [44,46,47].

1.2. Contribution

In the present paper we employ four novel approaches for the representation of handwritten words by means of graphs. A first approach is based on the representation of characteristic points by nodes, while edges represent strokes between these points. Another approach is based on a grid-wise segmentation of word images, where each segment is eventually represented by a node. Finally, two representation formalisms are based on vertical and horizontal segmentations of word images by means of projection profiles. For matching graphs we adopt the concept of graph edit distance which can be approximated in cubic time complexity by means of the Bipartite graph edit distance algorithm [46].

When compared to existing graph-based KWS approaches [43–45], the present approach distinguishes manifold. First, our graph representations results in a single graph for every word image. Hence, no additional assignment between graphs of different connected components is necessary during the matching process. Second, no prototype library (as used in [44,45]) is necessary for our graph representations. Thus, the risk of losing the main characteristics of handwriting is mitigated in our approach. Last but not least, besides single matchings we make use of ensemble methods [48] to combine the graph dissimilarities resulting from the the different graph representations.

The present article combines several lines of research and substantially extends three preliminary conference papers [49–51]. Moreover, in the empirical evaluation we use two additional datasets of a very recent KWS benchmark [52] and thoroughly present and discuss the evaluation of all parameters.

The remainder of this paper is organised as follows. In Sections 2 and 3, the proposed graph-based KWS approach is introduced. An experimental evaluation against template- and learning-based reference systems is given in Section 4. Finally, Section 5 concludes the paper and outlines possible future research activities.

2. Graph-based word representation

The proposed system for KWS is based on representing ancient handwritten documents by means of a set of single word images, which are in turn represented by graphs. Thus, a keyword can be retrieved in a document by matching a corresponding query graph against the complete set of document graphs. More formally, a specific graph matching algorithm computes the dissimilarities between the questioned keyword graph and all document graphs. Based on these graph dissimilarities a retrieval index can be derived. In the best case, this index represents all n instances of a given keyword as the final top-n results.