Searching and Browsing in Historical Documents—State of the Art and Novel Approaches for Template-Based Keyword Spotting

Michael Stauffer, Andreas Fischer and Kaspar Riesen

Abstract In many public and private institutions, the digitalization of handwritten documents has progressed greatly in recent decades. As a consequence, the number of handwritten documents that are available digitally is constantly increasing. However, accessibility to these documents in terms of browsing and searching is still an issue as automatic full transcriptions are often not feasible. To bridge this gap, *Keyword Spotting (KWS)* has been proposed as a flexible and error-tolerant alternative to full transcriptions. KWS provides unconstrained retrievals of keywords in handwritten documents that are acquired either *online* or *offline*. In general, offline KWS is regarded as the more difficult task when compared to online KWS where temporal information on the writing process is also available. The focus of this chapter is on handwritten historical documents and thus on offline KWS. In particular, we review and compare different state-of-the-art as well as novel approaches for *template-based* KWS. In contrast to *learning-based* KWS, template-based KWS can be applied to documents without any a priori learning of a model and is thus regarded as the more flexible approach.

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

M. Stauffer (✉) · K. Riesen
Institute for Information Systems, University of Applied Sciences and Arts Northwestern Switzerland, Riggenbachstrasse 16, 4600 Olten, Switzerland
e-mail: michael.stauffer@fhnw.ch

K. Riesen
e-mail: kaspar.riesen@fhnw.ch

M. Stauffer
University of Pretoria, Pretoria 0083, South Africa

A. Fischer
Department of Informatics, University of Fribourg, 1700 Fribourg, Switzerland
e-mail: andreas.fischer@unifr.ch

A. Fischer
University of Applied Sciences and Arts Western Switzerland, 1705 Fribourg, Switzerland

R. Dornberger (ed.), *Business Information Systems and Technology 4.0*,
Studies in Systems, Decision and Control 141,
https://doi.org/10.1007/978-3-319-74322-6_13

1 Broad Perspective and Outline

In the last decades, handwritten documents have become increasingly available digitally in many fields and applications. However, automatic full transcriptions of handwritten documents are far from perfect, especially as recognition is often negatively affected by degraded documents and/or different writing styles (Wicht et al. 2016). Thus, accessibility to handwritten documents with respect to browsing and searching is still an open issue. In order to overcome the obstacles of a full transcription, *Keyword Spotting (KWS)* has been proposed as a more error-tolerant and flexible approach for speech (Rose and Paul 1990), printed (Agazzi 1994), and handwritten documents (Manmatha et al. 1996). KWS refers to the task of retrieving any instance of a given query word in a particular document. In the case of historical handwritten documents, KWS is inherently an *offline* task, and as such, more complex than *online* KWS where temporal information on the writing process is also available. Since the focus of this chapter is on historical documents, only offline KWS—referred to as KWS from now on—can be applied.

1.1 Template-Based Versus Learning-Based KWS

Most KWS approaches are either *template-based* or *learning-based* algorithms. The following paragraphs provide a brief survey of methods stemming from both categories.

The earliest template-based KWS approaches are based on pixel-by-pixel matchings of word images (Manmatha et al. 1996). That is, the pixels of the word images are matched on the basis of Euclidean distance measures or affine transformations by the *Scott and Longuet*-algorithm (Scott and Longuet-Higgins 1991). Likewise, *Zones of Interest*, rather than single pixels, are matched in Leydier et al. (2007). More recently, word images have been described by binary features, so called *Gradient, Structural and Convexity (GSC)* features, and matched by correlation-like measures (Zhang et al. 2003).

However, single features tend to be affected by noise, and thus, more recent approaches to template-based KWS are based on matching sequences of feature vectors. These sequences of feature vectors are often used to represent certain characteristics of word images, such as, for example, projection profiles (Manmatha and Rath 2003; Rath and Manmatha 2003; Zhang et al. 2003), contours (Adamek et al. 2006; Can and Duygulu 2011), or geometrical characteristics (Marti and Bunke 2001; Manmatha and Rath 2003). However, more generic image feature descriptors have also been applied, for example, *Gabor* (Cao and Govindaraju 2007), *Histograms of Oriented Gradients* (Rodríguez-Serrano and Perronnin 2008; Terasawa and Tanaka 2009; Kovalchuk et al. 2014), *Local Binary Patterns* (Kovalchuk et al. 2014; Dey et al. 2016) or *Scale-Invariant Feature Transform* (Konidaris et al. 2015), to mention just a few. In a recent paper (Wicht et al. 2016), features are extracted by means of

436992_1_En_13_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:17/1/2018 Pages: 16 Layout: T1-Standard

a *Convolutional Deep Belief Network*. Regardless of the employed feature descriptor, *Dynamic Time Warping (DTW)* is probably the most widely used method for matching sequences of features vectors and is actually used in various KWS publications (Marti and Bunke 2001; Manmatha and Rath 2003; Adamek et al. 2006; Frinken et al. 2012; Wicht et al. 2016).

In contrast to template-based approaches, learning-based KWS is based on statistical models that have to be trained a priori with respect to the actual spotting task on a (relatively large) training set of word or character images. Early approaches to learning-based KWS are based on *generalized Hidden Markov Models (gHMM)* that are trained on character images, i.e. images of Latin (Edwards et al. 2004) or Arabic (Chan et al. 2006) characters. However, character-based segmentations are often error-prone. Thus, more recent approaches are based on feature vectors of word images (Lavrenko et al. 2004), which are processed, for example, by means of *Continuous-HMM* (Rodríguez-Serrano and Perronnin 2009) or *Semi-Continuous-HMM* (Rodríguez-Serrano and Perronnin 2009, 2012), i.e. HMMs with a shared set of *Gaussian Mixture Models*. In Perronnin and Rodríguez-Serrano (2009), a *Fisher Kernel* is employed in conjunction with HMMs, while a line-based and lexicon-free HMM-approach is introduced in Fischer et al. (2012). In recent papers, HMMs were applied in combination with *Bag-of-Features* (Rothacker et al. 2013; Rothacker and Fink 2015), or *Deep Neural Networks* (Thomas et al. 2014; Wicht et al. 2016). Other learning-based KWS approaches are for example based on *Support Vector Machines* (Huang et al. 2011; Almazán et al. 2014), or *Neural Networks* (Aghbari and Brook 2009; Frinken et al. 2012), to name just two examples.

Generally, learning-based approaches result in higher KWS accuracy when compared to template-based approaches. However, this advantage is accompanied by a loss of flexibility, which is due to the need to learn the parameters of the actual model. In particular, template-based KWS is independent of both the actual representation formalism and the language of the underlying document.

1.2 Statistical Versus Structural Representation

All of the KWS methodologies mentioned so far are based on statistical representation formalisms (this accounts for both template-based and learning-based methods). That is, word images or subimages are represented by means of feature vectors or sequences of feature vectors encoding certain local or global characteristics. However, in recent years a tendency towards structural representation formalisms has been observed in various fields of pattern recognition (Conte et al. 2004; Foggia et al. 2014; Riesen 2015; Stauffer et al. 2017d). Structural representations such as strings, trees, or graphs (whereby strings and trees can be seen as special cases of graphs) are more sophisticated data structures when compared to vectorial formalisms. In contrast to feature vectors, graphs are able to adapt both their size and structure to the underlying pattern. Moreover, graphs are able to represent binary relationships that might exist

between the subparts of the represented pattern. This turns graphs into a natural and comprehensive way for representing handwriting.

Given the power and flexibility of graphs, it might be rather surprising that few graph-based KWS approaches have been proposed until now (Wang et al. 2014; Riba et al. 2015; Bui et al. 2015; Stauffer et al. 2016b). One possible reason for this observation is the general increase in the complexity of many algorithms that use graphs rather than vectors as their input.

The first graph-based KWS approach was introduced in Wang et al. (2014), where certain keypoints in word images are represented by nodes, while edges are used to represent strokes between selected keypoints. The matching procedure is then conducted in two stages. First, graph dissimilarities between pairs of subgraphs are computed by means of a fast approximation algorithm (Riesen and Bunke 2009). Secondly, an optimal cost assignment is found by means of DTW. In Bui et al. (2015) and Riba et al. (2015), two similar approaches are shown, where nodes represent prototype strokes, while edges are used to represent the connectivity between strokes. Finally, graph dissimilarities are computed by the same algorithm as in Wang et al. (2014). One of the most recent graph-based KWS approaches was proposed by Stauffer et al. (2016a, b), where four different graph representation formalisms are introduced and compared with each other.

1.3 Outline

The present chapter focuses on reviewing template-based approaches for offline KWS. In particular, we review different state-of-the-art and novel approaches for template-based KWS for both statistical and structural representation formalisms in Sects. 2 and 3, respectively. Section 4 deals with an empirical comparison of both representations of two historical benchmark documents. Finally, Sect. 5 concludes this chapter and outlines future trends and rewarding opportunities.

2 Statistical Template-Based Keyword Spotting

In this section we review four different DTW-based systems for template-based KWS based on statistical representations, viz. Marti and Bunke (2001) (termed DTW'01), Rodríguez-Serrano and Perronnin (2008) (termed DTW'08), Terasawa and Tanaka (2009) (termed DTW'09), and Wicht et al. (2016) (termed DTW'16).

Basically, the four reviewed KWS systems consist of three subsequent steps, as illustrated in Fig. 1. First, document images are preprocessed (A) in order to minimise variations caused, for instance, by noisy background images, skewed scanning, or degraded documents. Subsequently, document images are automatically segmented into word images. Based on preprocessed word images, sequences of feature vectors are extracted by means of different feature descriptors (B). Finally, a query word

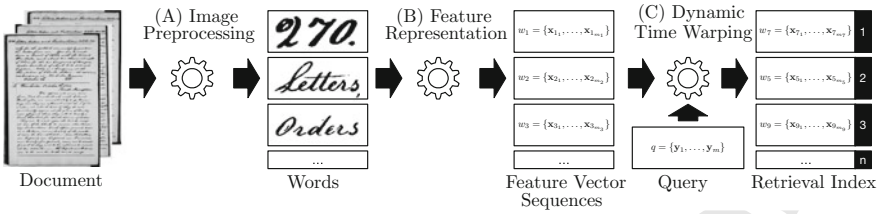


Fig. 1 Process of statistical template-based keyword spotting

(represented by a sequence of feature vectors) undergoes pairwise matching with a set of document words (represented by a set of sequences of feature vectors) (C). A retrieval index for a queried keyword can be derived on the basis of these dissimilarities. In the best possible case, this index represents all n instances of a given query word as the top-n results.

These three steps are described in greater detail in the following three subsections. It should be noted that the four systems only differ with respect to the extracted features. That is, the image preprocessing as well as the DTW-matching is conducted in quite a similar way in all four approaches.

2.1 Image Preprocessing

Image preprocessing aims at reducing variations caused by different writing styles (i.e. interpersonal variations) as well as the document itself (e.g. pixel noise, skewed scanning, or degraded documents). The reviewed systems rely on the following preprocessing steps.

The first preprocessing step addresses the issue of noisy background (e.g. by enhancing edges by a *Difference of Gaussians* (Fischer et al. 2010)). Next, document images are binarized by a global threshold and automatically segmented into single word images or word fragments. In addition, the skew, i.e. the inclination of the document, is also estimated on the lower baseline of a line of text and then corrected on the documents or single word images (Marti and Bunke 2001). Finally, the slant, i.e. the inclination of the handwriting, is also removed using a shear transformation (Marti and Bunke 2001).

2.2 Feature Representation

Based on preprocessed and segmented word images, sequences of feature vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ are extracted by means of a sliding window approach. In particular, a sliding window (with a user-defined width) is seamlessly moved over a word image

from left to right, and thus, one feature vector $\mathbf{x}_i$ is extracted at each window position i. The different DTW-based KWS systems differ with respect to the actual features extracted from the sliding window.

- **DTW'01** (Geometrical Features): In Marti and Bunke (2001), nine different geometrical features are defined for each window position. The first group of features describes the sliding window from a global perspective by the weight, center, and second order moment of the sliding window. Four features describe the position and orientation of the upper and lower contour in the sliding window, respectively. The two remaining features are used to characterize the number of black-white transitions in the vertical direction, as well as the number of black pixels between the upper and lower contour.
- **DTW'08 and DTW'09** (Histogram of Oriented Gradient (HoG) Features): In Rodríguez-Serrano and Perronnin (2008), HoG-features are locally extracted at each window position. In particular, the window is split into $M \times N$ cells of equal size. Based on the horizontal and vertical gradient components, the gradient magnitude m and angle θ are computed for each foreground pixel in the window cell. Thus, the gradient angles can serve to create a histogram with T radial bins. Angle θ determines the closest bin, while m sums up the corresponding bin. Hence, $M \times N \times T$ features are extracted for each window position. In Terasawa and Tanaka (2009), similar HoG-like features are extracted for overlapping blocks of cells rather than single cells.
- **DTW'16** (Deep Learning Features): In Wicht et al. (2016), a *Convolutional Deep Belief Network* based on two *Convolutional Restricted Boltzmann Machines (CRBM)* is used to extract features at each window position. In particular, the network is trained in an unsupervised manner in two subsequent steps. First, an image of the sliding window is used to train the first CRPM. The output of this layer is reduced by a pooling layer and used as input for the training of the second CRBM. Finally, the output of the second CRPM is again reduced by a pooling layer and used as a feature vector.

2.3 Dynamic Time Warping

All of the keyword spotting systems reviewed are based on matching a query word q with all document words $w_i \in \{w_1, \ldots, w_N\}$ by means of the dynamic programming approach DTW. In particular, DTW optimally aligns two sequences of features vectors $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ and $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ representing a query word q and a specific document word w_i along one common time axis using a dynamic programming approach. The alignment cost between each pair of feature vectors $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^k \times \mathbb{R}^k$ is given by the squared Euclidean distance. Formally,

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} (\hat{x}_i - \hat{y}_i)^2, \tag{1}$$

where k denotes the number of features, and $\hat{x}_i$ and $\hat{y}_i$ denote features normalized with a z-score. The DTW distance $D(q, w)$ between two sequences of feature vectors is then given by the minimum alignment cost found by dynamic programming. Formally,

$$D(X, Y) = \sum_{k=1}^{K} d(\mathbf{x}_{i_k}, \mathbf{y}_{j_k}), \tag{2}$$

where K is the length of the optimal warping path $((i_1, j_1), \ldots, (i_K, j_K))$ (Rath and Manmatha 2003). A *Sakoe-Chiba band* that constrains the warping path is often applied to speed up this procedure (Sakoe and Chiba 1978). Finally, a retrieval index can be created based on DTW distances between a query and all document words.

3 Structural Template-Based Keyword Spotting

In this section, we review two graph-based systems proposed by the authors of the present chapter for template-based KWS based on structural representations (Stauffer et al. 2016b, 2017a). Similarly to the statistical systems described in Sect. 2, the graph-based approaches consist of three subsequent steps, as illustrated in Fig. 2. First, document images are preprocessed and segmented into single word images (A). On the basis of preprocessed word images, graphs are extracted by means of a graph extraction algorithm (B). The actual keyword spotting is then based on a pairwise matching of a query graph with the set of all document graphs (C). A retrieval index is finally derived based on the resulting graph dissimilarities. In the following three subsections these steps are described in greater detail.

3.1 Image Preprocessing

The image preprocessing is based on similar steps as described in Sect. 2.1. That is, document images are filtered and binarized (Fischer et al. 2010), automatically segmented into word images, and manually corrected, if necessary. Next, the skew

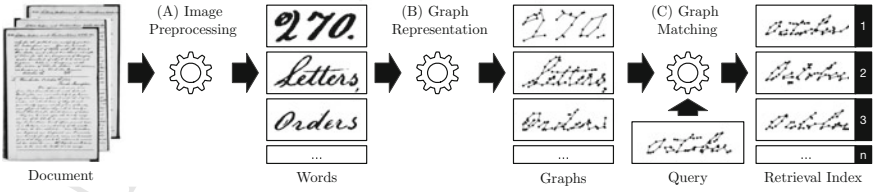


Fig. 2 Process of structural template-based keyword spotting

is estimated on lines of text and corrected on single words (Marti and Bunke 2001). However, in contradiction to the process described in Sect. 2.1, the slant is not corrected. Finally, word images are skeletonized by a 3×3 thinning operator (Guo and Hall 1989).

3.2 Graph Representation

In Stauffer et al. (2016b, 2017a), graphs serve to represent preprocessed and segmented word images. A graph g is defined as a four-tuple $g = (V, E, \mu, \nu)$ where V and E are finite sets of nodes and edges, and $\mu : V \rightarrow L_V$ as well as $\nu : E \rightarrow L_E$ are labeling functions for nodes and edges, respectively. All of the following four graph extraction algorithms (originally presented in Stauffer et al. (2016a)) result in graphs where nodes are labeled with two-dimensional numerical labels, while edges remain unlabeled, i.e. $L_V = \mathbb{R}^2$ and $L_E = \{\}$.

- **`Keypoint`**: The first graph extraction algorithm makes use of keypoints in word images such as start, end, and junction points. These keypoints are represented as nodes that are labeled with the corresponding (x, y)-coordinates. Between pairs of keypoints further intermediate points are converted to nodes and added to the graph at equidistant intervals. Finally, undirected edges are inserted into the graph for each pair of nodes that is directly connected by a stroke.
- **`Grid`**: The second graph extraction algorithm is based on a grid-wise segmentation of word images. For each segment, a node is inserted into the graph and labeled by the (x, y)-coordinates of the center of mass of this segment. Undirected edges are inserted between two neighboring segments that are actually represented by a node. Lastly, the inserted edges are reduced by means of a *Minimal Spanning Tree* algorithm (Kruskal 1956).
- **`Projection`**: The next graph extraction algorithm works in a similar way as `Grid`. However, this method is based on an adaptive segmentation of word images by means of horizontal and vertical projection profiles. A node is inserted into the graph for each segment and labeled by the (x, y)-coordinates of the corresponding center of mass. Undirected edges are inserted into the graph for each pair of nodes that is directly connected by a stroke in the original word image.
- **`Split`**: The last graph extraction algorithm is based on an iterative segmentation of word images. That is, segments are iteratively split into smaller subsegments until the width and height of all segments is below a certain threshold. A node is inserted into the graph and labeled by the (x, y)-coordinates of the point closest to the center of mass of each segment. For the insertion of the edges, a similar procedure as for `Projection` is applied.

Finally, the dynamic range of the (x, y)-coordinates of each node label $\mu(v)$ is normalized with a z-score (regardless the extraction algorithm). Formally,

$$\hat{x} = \frac{x - \mu_x}{\sigma_x} \text{ and } \hat{y} = \frac{y - \mu_y}{\sigma_y}, \tag{3}$$

where (μ_x, μ_y) and (σ_x, σ_y) represent the mean and standard deviation of all (x, y)-coordinates in the graph under consideration.

3.3 Graph Matching

The actual keyword spotting is based on pairwise matching of a query graph q with all document graphs $w_i \in \{w_1, \ldots, w_n\}$. Several approaches for graph matching have been proposed (Conte et al. 2004; Foggia et al. 2014). However, *Graph Edit Distance (GED)* is widely accepted as one of the most flexible and powerful paradigms available (Bunke and Allermann 1983). Given a query graph q and a document graph w, the basic idea of GED is to transform q into w using a sequence of edit operations. A standard set of edit operations is given by *insertions*, *deletions*, and *substitutions* of both nodes and edges. A set $\{e_1, \ldots, e_k\}$ of k edit operations e_i that transform q completely into w is referred to as an *edit path* $\lambda(q, w)$ between q and w.

To find the most suitable edit path, a domain-specific cost function $c(e)$ is usually introduced for each edit operation e. This cost function is used to measure the degree of deformation of a given edit operation. Given an adequate cost model, the graph edit distance $d_{\text{GED}}(q, w)$, or d_{GED} for short, between q and w is defined by

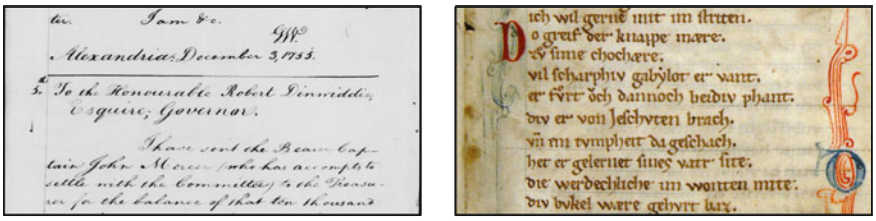
$$d_{\text{GED}}(q, w) = \min_{\lambda \in \Upsilon(q,w)} \sum_{e_i \in \lambda} c(e_i), \tag{4}$$

where $\Upsilon(q, w)$ denotes the set of all edit paths between q and w.

For the exact computation of d_{GED}, it is common to employ A*-based search techniques using heuristics (Fankhauser et al. 2011). However, these exhaustive search procedures are exponential with respect to the number of nodes of the involved graphs. Hence, in Stauffer et al. (2017a) the *Bipartite Graph Matching (BP)* algorithm (Riesen and Bunke 2009) is used, which approximates the GED in cubic time. Based on the resulting suboptimal graph edit distance $d_{\text{BP}}(q, w)$, a retrieval index is computed between query and document words.

3.3.1 Ensemble Methods

In Stauffer et al. (2017a), all graph representations as introduced in Sect. 3.2 are used in one KWS system at the same time. This approach is a well-known strategy from the field of *multiple classifier systems*, also referred to as *ensemble methods*. In particular, several query graphs (representing the same query word) are matched with several document graphs (representing the same document word). Next, different strategies are applied to combine the individual graph edit distances (derived



(a) George Washington (b) Parzival

Fig. 3 Exemplary excerpts of the two datasets

from the different representations). In Stauffer et al. (2017a), the minimal (termed min), maximal (termed max), or mean (termed mean) graph edit distance is used to condense the multiple distances to one retrieval index. Moreover, the most promising individual graph representations presented in Stauffer et al. (2016a), viz. `Keypoint` and `Projection`, are used to derive two weighted sums (termed sum_α and sum_{map}). The former sum makes use of a user-defined weighting value while the latter is based on a relative weighting that relies on the *Mean Average Precision* of the individual ensemble members. Eventually, the two normalized distances are summed up to form one single retrieval index.

4 Experimental Evaluation

In this section, we compare the reviewed statistical and structural approaches for template-based KWS with each other. The optimal parameters of the systems are taken from the corresponding papers. The comparison is carried out on two historical documents, viz. the *George Washington letters (GW)* and the *Parzival manuscript (PAR)* as shown in Fig. 3. GW is based on letters that are written in English and consists of twenty pages with a total of 4,894 handwritten words.[1] Variations caused by both degradation and writing style are low. PAR is based on a manuscript that is written in Middle High German and consists of 45 pages with a total of 23,478 handwritten words.[2] There are marked variations caused by degradation, while variations caused by writing style are low.

The performance of all KWS systems is measured by the *Recall (R)* and *Precision (P)*

$$R = \frac{TP}{TP + FN} \text{ and } P = \frac{TP}{TP + FP}, \tag{5}$$

[1] George Washington Papers at the Library of Congress, 1741–1799: Series 2, Letterbook 1, pp. 270–279 & 300–309, http://memory.loc.gov/ammem/gwhtml/gwseries2.html.

[2] Parzival at IAM historical document database, http://www.fki.inf.unibe.ch/databases/iam-historical-document-database/parzival-database.

436992_1_En_13_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:17/1/2018 Pages: 16 Layout: T1-Standard

which are both based on the number of *True Positives (TP)*, *False Positives (FP)*, and *False Negatives (FN)*.

Both recall and precision can be computed for two types of thresholds, viz. *local* and *global* thresholds. In the case of global thresholds, the quality of the KWS system is measured by *Average Precision (AP)*, which is the area under the *Recall-Precision (RP)* curve for all keywords given a single (global) threshold. In the case of local thresholds, the performance is indicated by *Mean Average Precision (MAP)*, that is the mean over the AP of each individual keyword query. Generally, global thresholds are regarded as the more realistic and challenging scenario.

For both benchmark datasets, the MAP and AP are given in Table 1. First, we compare the three individual approaches independently of each other (statistical, structural, and structural ensemble). In the case of statistical KWS, we observe that DTW'16 is the best approach in three out of four cases. However, DTW'09 also outperforms the two other statistical approaches, especially on PAR. In the case of structural KWS, we observe that `Keypoint` results in the highest KWS accuracy on GW, while `Projection` achieves the highest accuracy on PAR. On both datasets `Grid` and `Split` result in the lowest accuracy when compared to all other graph extraction methods. In the case of graph-based ensemble methods, we observe that the ensemble strategy mean achieves the best result in two out of four cases and the second and third best result in two cases.

Table 1 Mean average precision (MAP) using local thresholds and average precision (AP) using a global threshold for all DTW- and graph-based KWS systems. The first, second, and third best systems are indicated by (1), (2), and (3)

Method	GW		PAR	
	MAP	AP	MAP	AP
DTW				
DTW'01	45.26	33.24	46.78	50.67
DTW'08	63.39	41.20	47.52	55.82
DTW'09	64.80	43.76	73.49	69.10
DTW'16	68.64	56.98 (3)	72.38	72.71 (3)
Graph (Single)				
`Keypoint`	66.08	55.22	62.04	60.76
`Grid`	60.02	46.09	56.50	46.00
`Projection`	61.43	49.34	66.23	62.38
`Split`	60.23	48.08	59.44	56.25
Graph (Ensemble)				
min	70.56 (1)	56.82	67.90	62.33
max	62.58	47.94	67.57	50.59
mean	69.16 (3)	57.11 (2)	79.38 (1)	73.77 (1)
sum_{α}	68.44	55.78	74.51 (3)	68.12
sum_{map}	70.20 (2)	57.38 (1)	76.80 (2)	73.56 (2)

Comparing all systems with each other, we observe that the graph-based ensemble methods achieve the overall best results on both datasets and with both thresholds (with statistical significance (t-test, $\alpha = 0.05$)). In particular, the ensemble mean and sum_{map} outperform all other statistical and structural KWS approaches. This is particularly interesting as the DTW-based systems (Terasawa and Tanaka 2009; Wicht et al. 2016) use advanced feature sets, while the graph-based methods rely on coordinate labels only.

5 Conclusion and Outlook

In this chapter, different approaches for template-based Keyword Spotting (KWS) are reviewed. These methods basically differ in the formalism used to represent handwriting, viz. by means of statistical or structural representations. That is, preprocessed and segmented word images are either represented as sequences of feature vectors (in the case of statistical KWS) or graphs (in the case of structural KWS). The actual keyword spotting is then based on a matching of a query word with all document words by a dynamic programming approach or graph matching, respectively.

For the experimental evaluation both statistical and structural KWS approaches are compared with each other on two different benchmark datasets, viz. George Washington (GW) and Parzival (PAR). In the case of statistical KWS, DTW'16 is to favour on both datasets in most of the cases. In the case of structural methods, we observe that either `Keypoint` or `Projection` result in the highest accuracy on GW and PAR, respectively. Moreover, we observe that graph-based ensemble methods are able to clearly outperform all individual methods, as well as all statistical approaches.

One might argue that graph-based approaches are limited by the increased complexity of the matching procedure when compared to statistical approaches. However, recent papers (e.g. Stauffer et al. 2017b, c; Ameri et al. 2017) show that the complete KWS procedure with graphs can be substantially speeded up by filters and other heuristics. This makes graphs a versatile alternative for template-based KWS.

In future work, we see great potential in the combination of statistical and structural approaches. For instance, we plan to combine the matching scores derived by matching subgraphs of a sliding window with a DTW-based approach.

Acknowledgements This work has been supported by the Hasler Foundation Switzerland.

References

Adamek T, O'Connor NE, Smeaton AF (2006) Word matching using single closed contours for indexing handwritten historical documents. Int J Doc Anal Recogn 9(2–4):153–165

Agazzi O (1994) Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models. IEEE Trans Pattern Anal Mach Intell 16(8):842–848

Aghbari ZA, Brook S (2009) HAH manuscripts: a holistic paradigm for classifying and retrieving historical Arabic handwritten documents. Expert Syst Appl 36(8):10942–10951

Almazán J, Gordo A, Fornés A, Valveny E (2014) Segmentation-free word spotting with exemplar SVMs. Pattern Recogn 47(12):3967–3978

Ameri M, Stauffer M, Riesen K, Bui T, Fischer A (2017) Keyword spotting in historical documents based on handwriting graphs and Hausdorff edit distance. In: International graphonomics society conference

Bui QA, Visani M, Mullot R (2015) Unsupervised word spotting using a graph representation based on invariants. In: International conference on document analysis and recognition, pp 616–620

Bunke H, Allermann G (1983) Inexact graph matching for structural pattern recognition. Pattern Recogn Lett 1(4):245–253

Can EF, Duygulu P (2011) A line-based representation for matching words in historical manuscripts

Cao H, Govindaraju V (2007) Template-free word spotting in low-quality manuscripts. In: International conference on advances in pattern recognition, pp 1–5

Chan J, Ziftci C, Forsyth D (2006) Searching off-line arabic documents. IEEE Comput Soc Conf Comput Vis Pattern Recogn 2:1455–1462

Conte D, Foggia P, Sansone C, Vento M (2004) Thirty years of graph matching in pattern recognition. Int J Pattern Recogn Artif Intell 18(03):265–298

Dey S, Nicolaou A, Llados J, Pal U (2016) Local binary pattern for word spotting in handwritten historical document. Computing Research Repository

Edwards J, Teh YW, Bock R, Maire M, Vesom G, Forsyth DA (2004) Making latin manuscripts searchable using gHMM's. Int Conf Neural Inf Process Syst 17:385–392

Fankhauser S, Riesen K, Bunke H (2011) Speeding up graph edit distance computation through fast bipartite matching. In: Graph-based representations in pattern recognition, pp 102–111

Fischer A, Indermühle E, Bunke H, Viehhauser G, Stolz M (2010) Ground truth creation for handwriting recognition in historical documents. In: International workshop on document analysis systems, New York, USA, pp 3–10

Fischer A, Keller A, Frinken V, Bunke H (2012) Lexicon-free handwritten word spotting using character HMMs. Pattern Recogn Lett 33(7):934–942

Foggia P, Percannella G, Vento M (2014) Graph matching and learning in pattern recognition in the last 10 years. Int J Pattern Recogn Artif Intell 28(01)

Frinken V, Fischer A, Manmatha R, Bunke H (2012) A novel word spotting method based on recurrent neural networks. IEEE Trans Pattern Anal Mach Intell 34(2):211–224

Guo Z, Hall RW (1989) Parallel thinning with two-subiteration algorithms. Commun ACM 32(3):359–373

Huang L, Yin F, Chen QH, Liu CL (2011) Keyword spotting in offline chinese handwritten documents using a statistical model. In: International conference on document analysis and recognition, pp 78–82

Konidaris T, Kesidis AL, Gatos B (2015) A segmentation-free word spotting method for historical printed documents. Pattern Anal Appl

Kovalchuk A, Wolf L, Dershowitz N (2014) A simple and fast word spotting method. In: International conference on frontiers in handwriting recognition, pp 3–8

Kruskal JB (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. Proc Am Math Soc 7(1):48–48

Lavrenko V, Rath T, Manmatha R (2004) Holistic word recognition for handwritten historical documents. In: International workshop on document image analysis for libraries, pp 278–287

Leydier Y, Lebourgeois F, Emptoz H (2007) Text search for medieval manuscript images. Pattern Recogn 40(12):3552–3567
Manmatha R, Han C, Riseman E (1996) Word spotting: a new approach to indexing handwriting. In: Computer vision and pattern recognition, pp 631–637
Manmatha R, Rath TM (2003) Indexing of handwritten historical documents—recent progress. In: Symposium on document image understanding technology, pp 77–85
Marti UV, Bunke H (2001) Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition systems. Int J Pattern Recogn Artif Intell 15(01):65–90
Perronnin F, Rodríguez-Serrano JA (2009) Fisher kernels for handwritten word-spotting. In: International conference on document analysis and recognition, pp 106–110
Rath T, Manmatha R (2003) Word image matching using dynamic time warping. In: Computer vision and pattern recognition, vol 2, pp II–521–II–527
Riba P, Llados J, Fornes A (2015) Handwritten word spotting by inexact matching of grapheme graphs. In: International conference on document analysis and recognition, pp 781–785
Riesen K (2015) Structural pattern recognition with graph edit distance. In: Advances in computer vision and pattern recognition, Cham
Riesen K, Bunke H (2009) Approximate graph edit distance computation by means of bipartite graph matching. Image Vis Comput 27(7):950–959
Rodríguez-Serrano JA, Perronnin F (2008) Local gradient histogram features for word spotting in unconstrained handwritten documents. In: International conference on frontiers in handwriting recognition, pp 7–12
Rodríguez-Serrano JA, Perronnin F (2009) Handwritten word-spotting using hidden Markov models and universal vocabularies. Pattern Recogn 42(9):2106–2116
Rodríguez-Serrano JA, Perronnin F (2012) A model-based sequence similarity with application to handwritten word spotting. IEEE Trans Pattern Anal Mach Intell 34(11):2108–20
Rose R, Paul D (1990) A hidden Markov model based keyword recognition system. In: IEEE international conference on acoustics, speech, and signal processing, pp 129–132
Rothacker L, Fink GA (2015) Segmentation-free query-by-string word spotting with bag-of-features HMMs. In: International conference on document analysis and recognition, pp 661–665
Rothacker L, Rusinol M, Fink Ga (2013) Bag-of-features HMMs for segmentation-free word spotting in handwritten documents. In: International conference on document analysis and recognition, pp 1305–1309
Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech, Signal Process 26(1):43–49
Scott GL, Longuet-Higgins HC (1991) An algorithm for associating the features of two images. Proc Roy Soc B: Biol Sci 244(1309):21–26
Stauffer M, Fischer A, Riesen K (2016a) A novel graph database for handwritten word images. In: International workshop on structural, syntactic, and statistical pattern recognition
Stauffer M, Fischer A, Riesen K (2016b) Graph-based keyword spotting in historical handwritten documents. In: International workshop on structural, syntactic, and statistical pattern recognition
Stauffer M, Fischer A, Riesen K (2017a) Ensembles for graph-based keyword spotting in historical handwritten documents. In: International conference on document analysis and recognition
Stauffer M, Fischer A, Riesen K (2017b) Speeding-up graph-based keyword spotting by quadtree segmentations. In: International conference on computer analysis of images and patterns
Stauffer M, Fischer A, Riesen K (2017c) Speeding-up graph-based keyword spotting in historical handwritten documents. In: Graph-based representations in pattern recognition
Stauffer M, Tschachtli T, Fischer A, Riesen K (2017d) A survey on applications of bipartite graph edit distance. In: Graph-based representations in pattern recognition
Terasawa K, Tanaka Y (2009) Slit style HOG feature for document image word spotting. In: International conference on document analysis and recognition, pp 116–120
Thomas S, Chatelain C, Heutte L, Paquet T, Kessentini Y (2014) A deep HMM model for multiple keywords spotting in handwritten documents. Pattern Anal Appl 18(4):1003–1015

Wang P, Eglin V, Garcia C, Largeron C, Llados J, Fornes A (2014) A novel learning-free word spotting approach based on graph representation. In: International workshop on document analysis systems, pp 207–211

Wicht B, Fischer A, Hennebert J (2016) Deep learning features for handwritten keyword spotting. In: International conference on pattern recognition

Zhang B, Srihari SN, Huang C (2003) Word image retrieval using binary features. In: Document recognition and retrieval, p 45

436992_1_En_13_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:17/1/2018 Pages: 16 Layout: T1-Standard