Pattern Anal Applic (2016) 19:647–663
DOI 10.1007/s10044-014-0420-8

THEORETICAL ADVANCES

Cross-document word matching for segmentation and retrieval of Ottoman divans

Pinar Duygulu · Damla Arifoglu · Mehmet Kalpakli

Received: 27 September 2013 / Accepted: 7 September 2014 / Published online: 9 October 2014

Abstract Motivated by the need for the automatic indexing and analysis of huge number of documents in *Ottoman divan poetry*, and for discovering new knowledge to preserve and make alive this heritage, in this study we propose a novel method for segmenting and retrieving words in Ottoman divans. Documents in Ottoman are difficult to segment into words without a prior knowledge of the word. In this study, using the idea that divans have multiple copies (versions) by different writers in different writing styles, and word segmentation in some of those versions may be relatively easier to achieve than in other versions, segmentation of the versions (which are difficult, if not impossible, with traditional techniques) is performed using information carried from the simpler version. One version of a document is used as the source dataset and the other version of the same document is used as the target dataset. Words in the source dataset are automatically extracted and used as queries to be spotted in the target dataset for detecting word boundaries. We present the idea of cross-document word matching for a novel task of segmenting historical documents into words. We propose a matching scheme based on possible combinations of sequence of sub-words. We improve the performance of simple features through considering the words in a context. The method is applied on two versions of *Layla and Majnun* divan by Fuzuli. The results show that, the proposed word-matching-based segmentation method is promising in finding the word boundaries and in retrieving the words across documents.

Keywords Segmentation · Retrieval · Matching · Historical documents · Ottoman divans

1 Introduction

UNESCO launched a programme for "Memory of the World" to promote the world's documentary heritage. The Ottoman Empire lasted for more than six centuries, spread over three continents and left a remarkable legacy behind. In this marvellous heritage, there are huge collections of documents (archival, literary, etc.) that are currently preserved in the archives, libraries, museums and private collections of almost forty nations, constituting an important part of the world's memory.

These historical documents, most of which are in handwritten, manuscript or in rare old printed editions, attract the interests of scholars from many disciplines (history, literary studies, sociology, etc.)[1]. On the other hand, access to these historical texts is severely limited. Recent attempts in digitisation of the archival material are important for the preservation and electronic access of

This work was done while the second author was a graduate student in the Department of Computer Engineering, Bilkent University, Ankara, 06800 Turkey.

P. Duygulu (✉)
Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey
e-mail: duygulu@cs.bilkent.edu.tr

D. Arifoglu
Computer Engineering Department, Sabanci University, Istanbul 34956, Turkey

M. Kalpakli
Department of History, Bilkent University, Ankara 06800, Turkey

[1] Ottoman Text Archive Project (OTAP), url: http://courses.washington.edu/otap/.

these documents[2], however there is a lack of resources for analysis and translation except a few recent attempts [6, 7, 12, 13, 52, 62, 63].

While Ottoman Empire is known as one of the most powerful and significant forces in its era, the Ottoman literature is almost invisible to the world [3]. The poetry of the Ottoman Empire, or *Ottoman Divan poetry*, and its literary tradition that lasted for nearly six centuries is rarely known today. As stated in [2], "Achieving a statistically accurate picture of the vocabulary of the Ottoman lyrics would demand a vast recording, sorting, and counting project, which, although far from impossible using modern computer techniques, would require resources beyond what is currently available".

Ottoman poetry is a highly sophisticated and symbolic art form, and therefore inherits many difficulties compared to other historical texts. It is built upon shared knowledge of previously employed themes and cultural motives using lexical tools. It was composed through the constant juxtaposition of many such images within a strict metrical framework, thus allowing numerous potential meanings to emerge.

Ottoman *divans*, which are collections of poems (around 500 poems in one single divan) written by the same poet, were copied with different copyist and scribers over the years. Copying process resulted in multiple copies of the same divan text with many errors, different versions of the same poem and missing lines or parts of the poems.

Today, multiple copies of divans are studied by scholars manually to find the variants between different manuscripts, and to reach the correct text. Editing a divan means transcribing the text to produce a text as close as possible to the authors' own copy (autograph copy). It is very important to help the transcriber for his or her decision on showing variants and correcting the errors of the manuscripts.

With the help of an automatic word segmentation process, scholars of historical text editing may be able to manage a big number of divans (for example, some of the sixteenth century divans have more than 100 copies in manuscript libraries) and it may be much easier for the transcribers to show all of the variants of the text and select the correct word for the final edition.

However, word segmentation in Ottoman documents is difficult, if not impossible, without prior knowledge of the words. In Ottoman, a word can be comprised of many sub-words, (a sub-word is a connected group of characters or letters, which may be meaningful individually or only meaningful when it comes together with other sub-words) and a space does not necessarily correspond to a word boundary. There is no explicit indication where a word ends and another begins. The intra-word gaps can be as large as inter-word gaps, or both gaps can be very small. The word boundaries can only be decided, especially in some handwritten documents, only by reading the word.

We make use of multiple copies (versions) of the same divan for word segmentation. Our approach is based on the idea that some versions are easier to segment than others, and the difficult versions can be segmented by transforming information from the easier ones. For this purpose, we propose a cross-document word-matching method. Prior knowledge obtained from a source document in the form of segmented words is carried to a target document by spotting the words across documents. We, therefore, "read" the target document words by the help of the source document words.

We melted segmentation and retrieval in the same pot: segmentation is performed through retrieving words, and retrieval performance is increased by segmentation. While the proposed method is language and script independent and can be applied to any pair of documents, in this study we focus on Ottoman divans with the appealing idea of helping to scholars to discover this barely touched area.

In this study, our main contributions are as follows.

1. We apply the word-matching idea for segmenting historical documents into words, (which is difficult, if not impossible, using classical word segmentation techniques), by carrying information from other sources. To the best of our knowledge, this is the first application of the word-matching for segmenting words across documents.
2. Words may be broken into different numbers of smaller units in different versions of historical documents. In this study, rather than using entire words that may produce unsuccessful results when cross-document word-matching (which is a more difficult task compared to word-matching in the same document) is considered, we consider the sub-words to be more robust, and propose a word-matching method based on possible combinations of ordered sequences of sub-words.
3. We use context information for word matching. When words are spotted across documents individually, it is possible to mismatch them. We consider words in a context, in the form of lines or sentences, with its consecutive and preceding words.

In the following, we first review the related studies on word segmentation and word matching in Sect. 2. Then, in Sect. 3.1, we discuss the challenges of Ottoman divans used in our study. The proposed approach is described in Sect. 3, followed by detailed experiments for evaluating segmentation and retrieval performances in Sect. 4. We conclude in Sect. 5 by discussing the results and possible future improvements.

[2] State Archives Office of Turkey, url: http://www.devletarsivleri.gov.tr/.

2 Related work

In recent years, interest in preserving and accessing historical documents has increased. While indexing and retrieval of these documents are desired, applying ordinary optical character recognition (OCR) techniques on them is nearly impossible due to deformations caused by faded ink or stained paper and noise because of deterioration [40].

As an alternative, word spotting techniques have been proposed for easy access and navigation of historical documents [37, 38, 49]. Most of these techniques require word segmentation before searching for a word [6, 7, 37]. Although there are some segmentation-free approaches [1, 17, 20, 28, 29], their computational cost is usually high. Thus, providing a word segmentation schema would be beneficial and make the searching processes easier and faster. On the other hand, word segmentation is difficult in historical documents, where words may touch each other due to handwriting style or high noise levels.

Majority of the proposed segmentation algorithms [44] may not be useful in historical documents, because of degradation due to printing quality and ink diffusion. For segmenting historical documents, generally methods that are based on the analysis and classification of the distance relationship of adjacent components are used [22, 30, 31, 36, 39, 44, 53, 55, 61]. These methods, however, are likely to fail with languages such as Ottoman, Persian, Arabic, etc., in which there are inter-word gaps as well as intra-word gaps in a document, and determining which is not easy.

In word spotting literature, dynamic time warping (DTW) is one of the most commonly used methods to calculate the similarity of words [9, 18, 33, 43, 46, 48]. DTW can tolerate spatial variations unlike other methods such as XOR, Euclidean Distance Mapping, Sum of Squared Differences [47]. Alternative to the methods matching words based on whole images or profile-based features [46], recently other features are also experimented, including word contours [57], gradients [34, 50, 55, 65], shape context descriptors [35], Harris corner detector outputs [51], line segments [12, 13], and interest points [7]. In [11, 21, 49, 57], the problem of writing style variations in multi-writer datasets is tackled, but these studies generally require isolated words. In a recent study [17], a method based on character HMMs is proposed as an alternative to template-based methods.

In [10], a method based on M-band packet wavelet transform is proposed for recognition of handwritten Farsi words. In [27], segmentation and word spotting techniques are compared on clean printed and handwritten Arabic documents. In [8], considering the errors in word segmentation on Arabic documents, alternatively a segmentation-free approach is proposed for word spotting.

While our approach is related to recognition-based character-segmentation studies in the literature [14, 59, 62, 64], to the best of our knowledge, there is no recognition-based word segmentation method for Ottoman documents.

Although the word spotting literature is dominated by single word matching, in [41] words are modelled as a concatenation of Markov models and a statistical language model is used to compute word bigrams. In our study, in a similar direction, initial matching is performed on sub-words, and then neighbouring sub-words are combined for word matching. The recent work of Khurshid et al. [25] is another study that uses the idea of comparing sequences of sub-words for word spotting. Beyond segmentation errors resulting in different sub-words and therefore consideration of sub-word sequences in word matching, we tackle a much harder problem of matching sub-words from different sources.

In a recent segmentation-free word spotting approach for Arabic documents [28, 29], the authors propose a learning-based word spotting system. For the first time in the literature of Arabic word spotting, language models were integrated with the partial segmentation of the words, to represent contextual information and reconstruct words. The aim was to search for lexicon words within Arabic handwritten documents. The method is based on the partial segmentation of the lexicon and the documents into pieces of Arabic words (PAWs) to overcome the lack of boundaries problem. The segmented PAWs are passed to a hierarchical classifier to perform the final classification or arrive at a rejection decision. The system is a learning-based word spotting system for which there are training data consisting of samples of the lexicon words (Words Database) and a separate set of Testing Documents. Each lexicon word in the Words Database is partially segmented into its constituent components or PAWs by first segmenting the word into its connected components.

Our work resembles to the studies in aligning historical manuscripts to their inaccurate transcripts [16, 42, 58], in the sense that matching across documents is considered. However, they require a transcript which is not available for Ottoman documents in most of the cases.

In a recent study [5], printed and handwritten documents in Arabic are aligned. They extract column features to compare components using string matching techniques. However, they assume that there is no touching component between constituting words resulting in clear segmentation of the components and the user semi-automatically selects the area to be aligned. This assumption fails for different versions of Ottoman divans where sub-words may have different numbers of overlapping and touching components. Besides, in Ottoman divans not only the words but also the entire lines may be omitted or their order may change. These challenges are addressed in this study in a fully automatic manner.

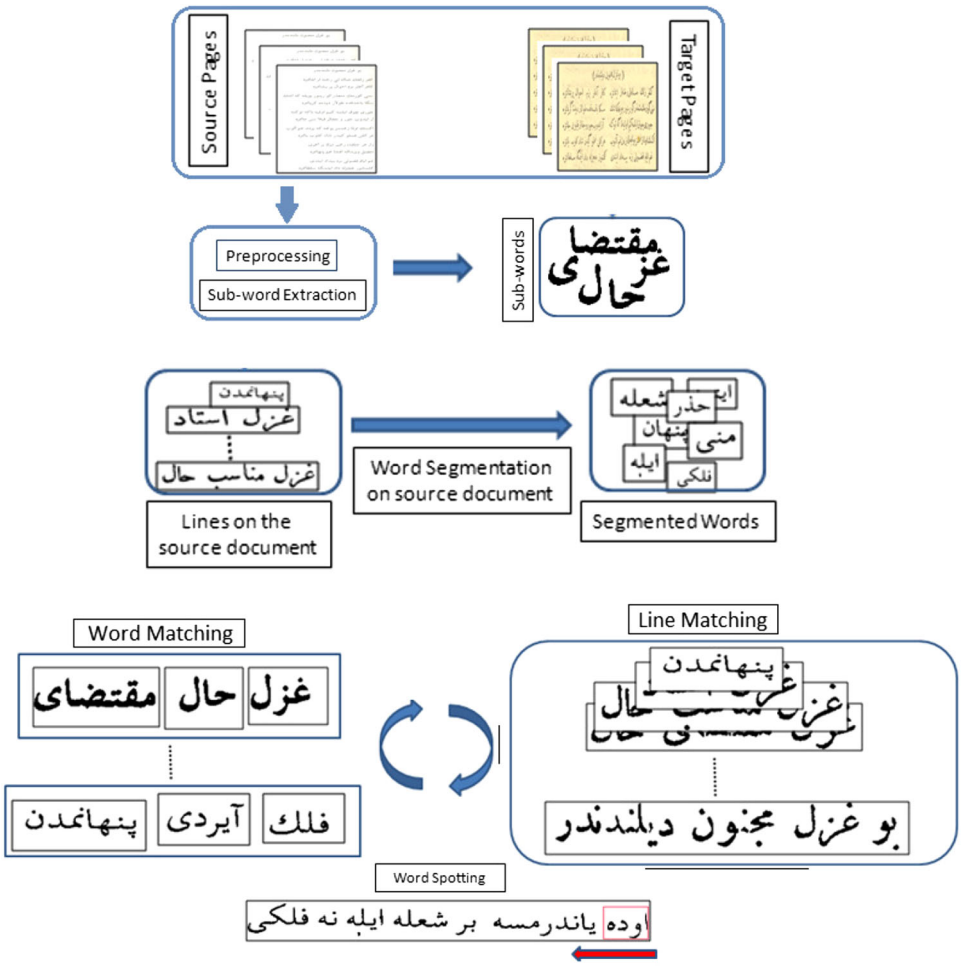


Fig. 1 Overview of the proposed approach

The idea of the proposed method also resembles to that of the domain adaptation and transfer learning fields in the sense of usage of source domain to label target domain. Domain adaptation (also referred to as transfer learning or cross-domain learning) is an emerging research topic in computer vision. The domain of interest (target domain) which contains very few or even no labelled samples is tried to be labelled using an existing domain (source domain) with a large number of labelled examples [24, 54, 56, 60]. Multitask learning or learning multiple related tasks simultaneously has shown a better performance than learning these tasks independently. Therefore, it is meaningful to study cross-domain representation learning which can transfer common knowledge structures from source domains to the target domain to help the tasks on the target test datasets.

3 Proposed approach

In this study, a cross-document word-matching-based method, which is comprised of the following steps is presented for segmenting documents into words when multiple copies (versions) are available (see Figure 1). (i) A version of a document that is easy to segment into words is chosen as the source dataset and version of the same document in a different writing style is used as the target dataset. (ii) All sub-words in the source and target datasets are extracted. (iii) From the source dataset, words are extracted by a simple word segmentation method. (iv) Extracted words are sought in the target dataset to determine the word boundaries by a method that performs matching of words and lines concurrently. In the following, first the challenges of Ottoman divans will be presented to discuss the requirement for the proposed method and then each step will be described in detail.

3.1 Challenges of multi-version divans

In different versions of a divan, although the content is generally the same, the documents may exhibit some differences (Fig. 2). For example, they may have different numbers of words and lines; some words and lines may have been omitted or new words and lines may have been

Fig. 2 Example pages **a** from the source dataset, which is machine printed, **b** from the target dataset, which is a lithograph. The *solid lines* indicate the correct word boundaries. In the source image, it is easier to find word boundaries, while in the target image it is harder to define intra- and inter-word gaps. Lines 10 and 11 of source page are missing in the target page. The *words in rectangles* are different or written in a different form between the two datasets. The sub-words *underlined* are the same in both images, but their characters have different shapes. Across documents, due to differences in writing **c** the same word may have different numbers of sub-components, and **d** sub-components may be different. The *top rows* of **c** and **d** show examples from the source dataset, and the *bottom rows* show the corresponding lines from the target dataset

added. Most importantly, the writing style (character shapes) may be different. For example, a character may have a long curve in one version, and a shorter curve, or additional curves in another version. Moreover, some historical documents may have broken characters in some versions because of deterioration resulting in different numbers of sub-components corresponding to a word.

In this study, the source dataset is chosen as a machine-printed version, and the target dataset is chosen as a lithography version of a divan. Lithography is a method in which a stone or a metal plate with a smooth surface is used to print text onto paper [26]. It was the first fundamentally new printing technology and was invented in 1798. In this method, letters or characters are not ordered by machine, they are placed on the stone or plate by humans. Spaces between characters are sometimes very large and sometimes very small. This was normally the case in handwritten documents, but this trend was also observed in lithography, showing that aspects of the handwriting culture are continued in lithography [26, 32]. In a machine-printed text, word boundaries can be easily determined by classifying the space between characters, but it is not easy to distinguish between inter-word and intra-word distances on lithographs. Thus, word segmentation methods based on gap distances are likely to fail in these documents. Lithography texts are chosen as being the best sources to transfer the segmentations from printed documents without tackling with the representation problems in handwritten documents.

3.2 Preprocessing

The datasets used in the experiments are relatively clean, therefore we use simple methods for preprocessing. First, the original documents are converted into grey scale, and they are binarised by an adaptive binarisation method [23]. Small noises such as dots and other blobs are cleaned by removing connected components smaller than a predefined threshold. Then, pages are segmented into lines by a run length smoothing algorithm [36]. Broken characters are

Fig. 3 All the black pixel groups are individual connected components (CC). There are 11 CCs in this image (six of them are major, and five of them are minor) that are merged to result in six sub-words. Sub-words are separated by *lines*

connected in 4-neighbourhood if their distances are smaller than a predefined threshold. Thresholds are learned from dataset samples.

3.3 Sub-word extraction

First, all connected components (CCs) in the source and target datasets are extracted by a boundary-detection algorithm. A CC is defined as a connected group of black pixels in the document image. Diacritics such as dots and zig zags are considered as minor components and other larger components such as letters and connected groups of characters are considered as major components. The width and height thresholds of minor/major components are learned on a small manually labelled set. After decision of major/minor components, minor components are connected to their closest major components to construct sub-words. If a minor component is inside the bounding box of a major component, then they are assumed to be connected (see Figure 3). These constructed sub-words may be individual words on their own or they may form words by joining with other sub-words.

3.4 Sub-word matching

Since historical documents are degraded, the same word may be split into different numbers of sub-words in source and target datasets. Therefore, word-matching methods that are based on the representation of entire words are likely to fail in cross-document word matching.

We use the sub-words, which are more robust than words, as basic units, and matching is performed across sub-words of source and target datasets (see Fig. 5a). Inspired by [19, 41, 46], we choose to use simple features (namely upper/lower vertical projection, background-to-ink transition, second-moment order, centre of gravity, number of foreground pixels between upper and the lower contours, and variance of ink pixels), for representing sub-words and Dynamic Time Warping (DTW) for matching words. In Fig. 4, a word image and its corresponding 10 features are given.

Let s be a sub-word in the source dataset and t be a sub-word in the target dataset. The similarity between s and t, $d(s,t)$ is defined as:

$$d(s_i,t_j)=\sum_{k=1}^{n}(f_k(s,i)-f_k(t,j))^2 \tag{1}$$

$$D(s,t)=\sum_{k=1}^{K}d(s_{ik},t_{jk}) \tag{2}$$

Here, $f_i(s)$ and $f_i(t)$ are the features extracted from the sub-word images, and n is the number of features, K is warping path length, i is image column of source sub-word s and j is image column of target sub-word and they are matched.

In DTW, the distance between two time series, which are lists of samples taken from a signal, ordered by time, is calculated with dynamic programming as in the equation (3), where $dist(x_i,y_i)$ is the distance between i^{th} samples.

$$DTW(i,j)=min\left\{\begin{array}{c}DTW(i,j-1)\\DTW(i-1,j)\\DTW(i-1,j-1)\end{array}\right\}+dist(x_i,y_i) \tag{3}$$

Without normalisation, DTW algorithm may favour the shorter signals. To prevent this, a normalisation is done based on the length of the warping path.

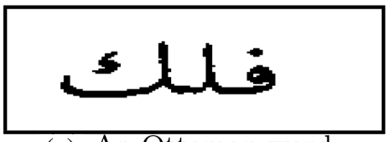
As will be shown by the experiments, the chosen features and similarity measure have major limitations, especially when documents with large variations are considered. We choose them to provide a baseline as they are commonly used in the literature. The main advantage of the chosen features is that, when it is needed to combine the sub-words for the proposed word-matching method described below, the features of the new word image can be obtained easily from its components without requiring additional feature extraction. The best characteristic of DTW algorithm is that the two samples do not have to be in the same size. In this way, the same words in different sizes can be easily matched and this is an important feature of DTW in cross-document word matching. In cross documents, the same characters may be in different sizes because of different writing styles and fonts.

3.5 Cross-document word matching based on sub-word sequence matching

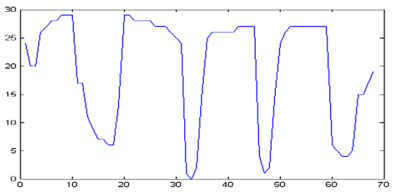
We consider words as ordered sequence of sub-words and propose a method for cross-document word matching based on sub-word sequence matching.

Let $S=(s_1,....s_{N_S})$ be a word in the source dataset with N_S sub-words, and $T=(t_1,....t_{N_T})$ be a word in the target dataset with N_T sub-words. Assume that a sub-word $s_i \in S$ is broken into a list of sub-words $(t_k,\ldots,t_{k+l}) \in T$ due to degradation (we assume that sub-words in source dataset

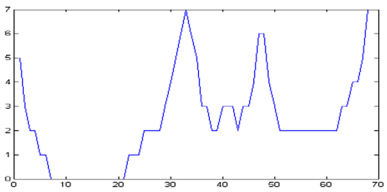
Fig. 4 **a** An Ottoman word (*"fate"*). (**b**–**k**) Its 10 features



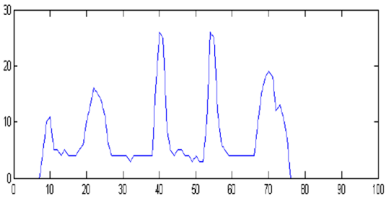
(a) An Ottoman word



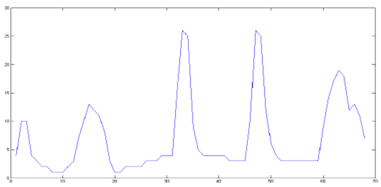
(b) Upper word profile



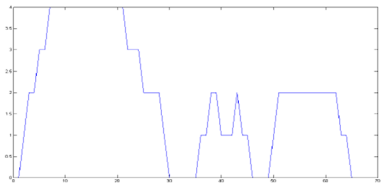
(c) Lower word profile



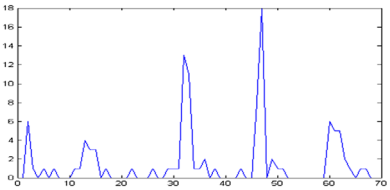
(d) Vertical projection



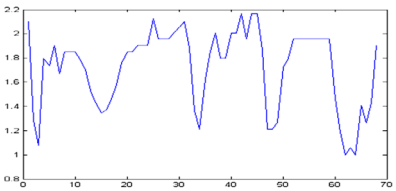
(e) Upper vertical projection profile



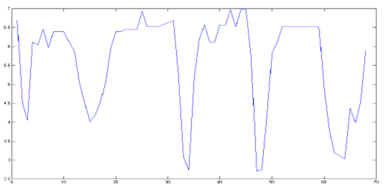
(f) Lower vertical projection profile



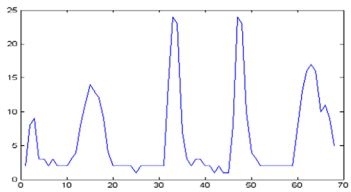
(g) Background-to-ink transition



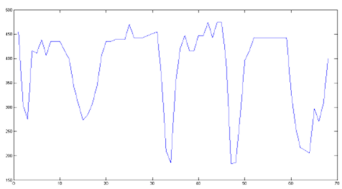
(h) Second-moment order



(i) Center of gravity



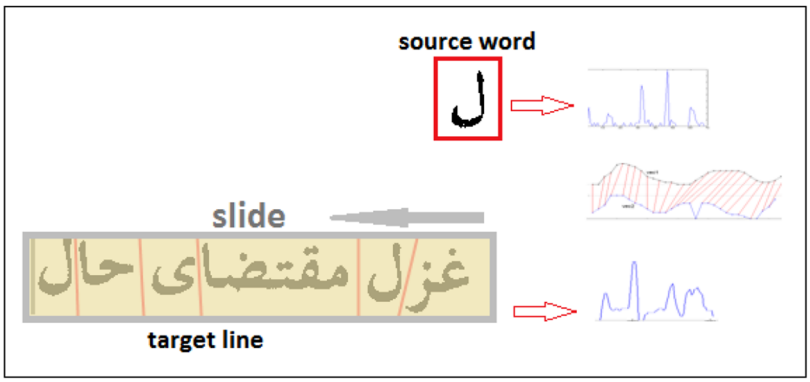
(j) Number of foreground pixels between upper and lower contours

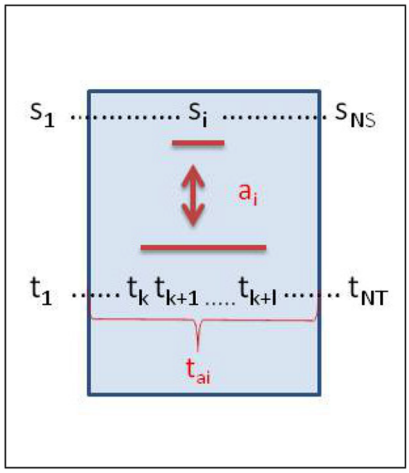


(k) Variance

are not broken since it is a printed version). The list of target sub-words $(t_k, \ldots, t_{k+l})$ aligned with the source sub-word s_i is defined as t_{a_i} (see Fig. 5b).

The similarity of the source word S and the target word T is found by summing the distances between source sub-words and the aligned target sub-words.





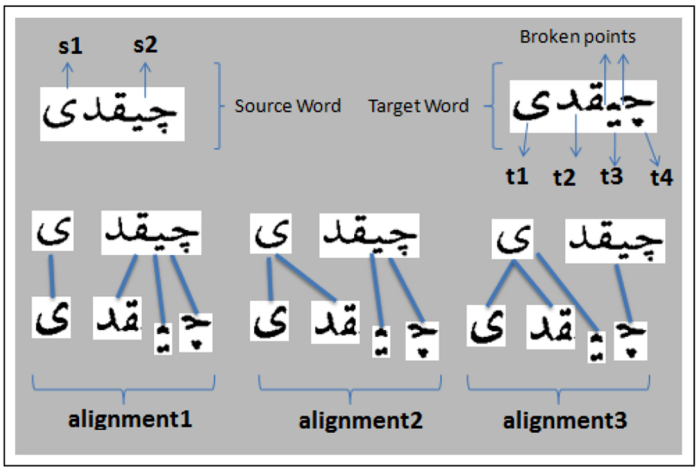


Fig. 5 **a** Subword matching. **b** An example alignment between a source sub-word s_i and a set of target sub-words t_{a_i}. **c** Word-matching example with three different sub-word alignments

$$D(S,T) = \sum_{i=1}^{N_S} \{d(s_i, t_{a_i})\} \tag{4}$$

Note that one source sub-word can be aligned with up to $|N_T - N_S| + 1$ target sub-words.

Let us assume that there can be N different alignments, and define $D_j(S,T)$ as the distance between S and T in a possible alignment j. The distance $D(S,T)$ is found as the minimum of all possible alignments.

$$D(S,T) = min(D_j(S,T)), \quad j = 1:N \tag{5}$$

To illustrate the proposed approach, consider the toy example in Fig. 5c where one of the sub-words in the source word is split into three sub-words in the target word because of deterioration. For the source word consisting of two sub-words $S = (s_1, s_2)$, and the target word consisting of four sub-words $T = (t_1, t_2, t_3, t_4)$, there are three possible alignments.

$D_1(S,T) = d(s_1, t_1) + d(s_2, t_{a_2})$, where $t_{a_1} = \{t_1\}$ and $t_{a_2} = \{t_2, t_3, t_4\}$

$D_2(S,T) = d(s_1, t_{a_1}) + d(s_2, t_{a_2})$, where $t_{a_1} = \{t_1, t_2\}$ and $t_{a_2} = \{t_3, t_4\}$

$D_3(S,T) = d(s_1, t_{a_1}) + d(s_2, t_{a_2})$, where $t_{a_1} = \{t_1, t_2, t_3\}$ and $t_{a_2} = \{t_4\}$

The similarity of the two words is then computed as the minimum of the three possible alignments:

$$D(S,T) = \min(D_1(S,T), D_2(S,T), D_3(S,T)) \tag{6}$$

In this example, the minimum score is obtained for the first alignment.

3.6 Line matching

In divans, the correct meaning of a word is captured through interrelationships of words. In this study, we make use of the context information provided by lines that are ordered sequences of words to increase the word-matching performance. For this purpose, we perform line matching prior to word matching.

The approach for matching lines resembles the approach for word matching. An entire line is assumed to be a single word consisting of sub-words. However, they differ in one point. While they may contain different numbers of sub-words, a source word and its corresponding target word are the same. On the other hand, when the lines are considered, there could be words omitted or added in a line in different versions of divans. Therefore, we allow *null* sub-words either in source or target line.

3.7 Selection of candidate lines and the best matching line

In Ottoman (Divan) poetry, most of the poems are based on a pair of lines, i.e., distich or couplets. A distich contains two hemistichs (lines). Ghazal is a poetic form consisting of rhyming couplets.

In different versions of the divans, some ghazals may be missing or their order may be different. Similarly, ghazals may have different numbers of couplets in different orders.

Before matching lines, some pruning is performed to reduce the number of candidate lines by finding the matching ghazals. First, the total number of sub-words in source and target ghazals is calculated separately. If the difference of the number of sub-words in a source ghazal and a target ghazal is smaller than a certain threshold, they are considered as candidate matches. Then, the same approach is applied to reduce the number of candidate lines in the reduced set of ghazals. We again allow *null* matching for either source or target line since some lines may be omitted or new lines may be added to the new version of a Divan.

Let $V = (G_1, G_2, ...G_K)$ be a version of divan consisting of ghazals, and $G_i = (L_i^1, L_i^2, ...L_i^L)$ be a ghazal consisting of lines. The set of candidate target ghazals $C(SG_i)$ for a source ghazal SG_i and the set of candidate lines $C(SL_k)$ for a source line SL_k are defined as follows:

$$TG_j \in C(SG_i), \quad if |TG_j| - |SG_i| \leq th1 \tag{7}$$

$$TL_l \in C(SL_k), \quad if |TL_l| - |SL_k| \leq th2 \tag{8}$$

Here, TG_j is a target ghazal, and TL_l is a target line.

The target line $TL_m \in C(SL_k)$ is considered as the matching line, $match(SL_k)$, for source line SL_k, if $D(SL_k, TL_l)$, the similarity of a source line SL_k and target line TL_l, is minimum. That is,

$$match(SL_k) = TL_m \quad if \quad D(SL_k, TL_m) \leq D(SL_k, TL_l), \quad \forall TL_l \in C(SL_k) \tag{9}$$

3.8 Segmentation of target dataset into words

Let $SL = (s_1, s_2, \ldots s_{N_S})$ be a source line with N_S sub-words, and $TL = (t_1, t_2, \ldots t_{N_T})$ be its best matching target line with N_T sub-words. We segment the target line into words by spotting the source words on the target line in order. Assume that a source word starts at s_i and includes n sub-words. Also, assume that we will search for the target word starting at t_j. Since a source sub-word may be broken into multiple target sub-words, we extend the search space with a window w. Therefore, the best alignment for the source word with n sub-words is searched among $n + w$ target sub-words. That is, we search for the alignments for source sub-words $(s_i, s_{i+1}, \ldots s_{i+n})$ in a range: $(t_j, t_{j+1} \ldots, t_{j+n}) - (t_j, t_{j+1} \ldots, t_{j+n+w})$. If the best alignment with minimum distance is found for sub-words $(t_j, \ldots t_k)$, where $(j + n) \leq k \leq (j + n + w)$, the boundary is set at sub-word t_k and the search for the next source word starts at the target sub-word t_{k+1}.

4 Experiments

4.1 Dataset and evaluation criteria

In this study, we choose to study two versions of *Layla and Majnun* divan, a famous work of Fuzuli who is considered one of the greatest contributors to the Divan tradition. The source dataset is a machine-printed version [15] which is of good quality, and is not noisy or degraded. There is no deformation on the pages, and intra- and inter-word distances can be easily distinguished, and it is relatively easy to segment it into words with a simple word segmentation method. The target dataset is nearly 100 years older and is a lithograph version [4]. It is not of good quality as it has some deformations and noise. Datasets were obtained by scanning books with a resolution of 300×300.

Both datasets consist of 26 ghazals. There are up to 29 lines and 226 words in these ghazals. On average, there are 10 lines and 102 words in a ghazal. A line may be as short as one word, and at most, there are seven words in a line (see Fig. 6).

Although both documents correspond to the same work, they have different numbers of lines and words (see Table 1). In total, there are 408 lines in the source dataset and 402 lines in the target dataset, with eight lines of the source dataset that are not in the target dataset and two lines are added to the target dataset that are not in the source dataset. Among 2688 words in the source dataset and 2,640 words in the target dataset, 124 words in the source dataset are not in the target dataset, while 76 words in the target set are not in the source dataset. The number of unique words is 1,379 and 1,357, and the number of sub-words is 5,964 and 6,186, respectively, for source and target datasets. Most of the words appear only a few times.

614 of the source dataset words have broken sub-words. Further some sub-words are not extracted

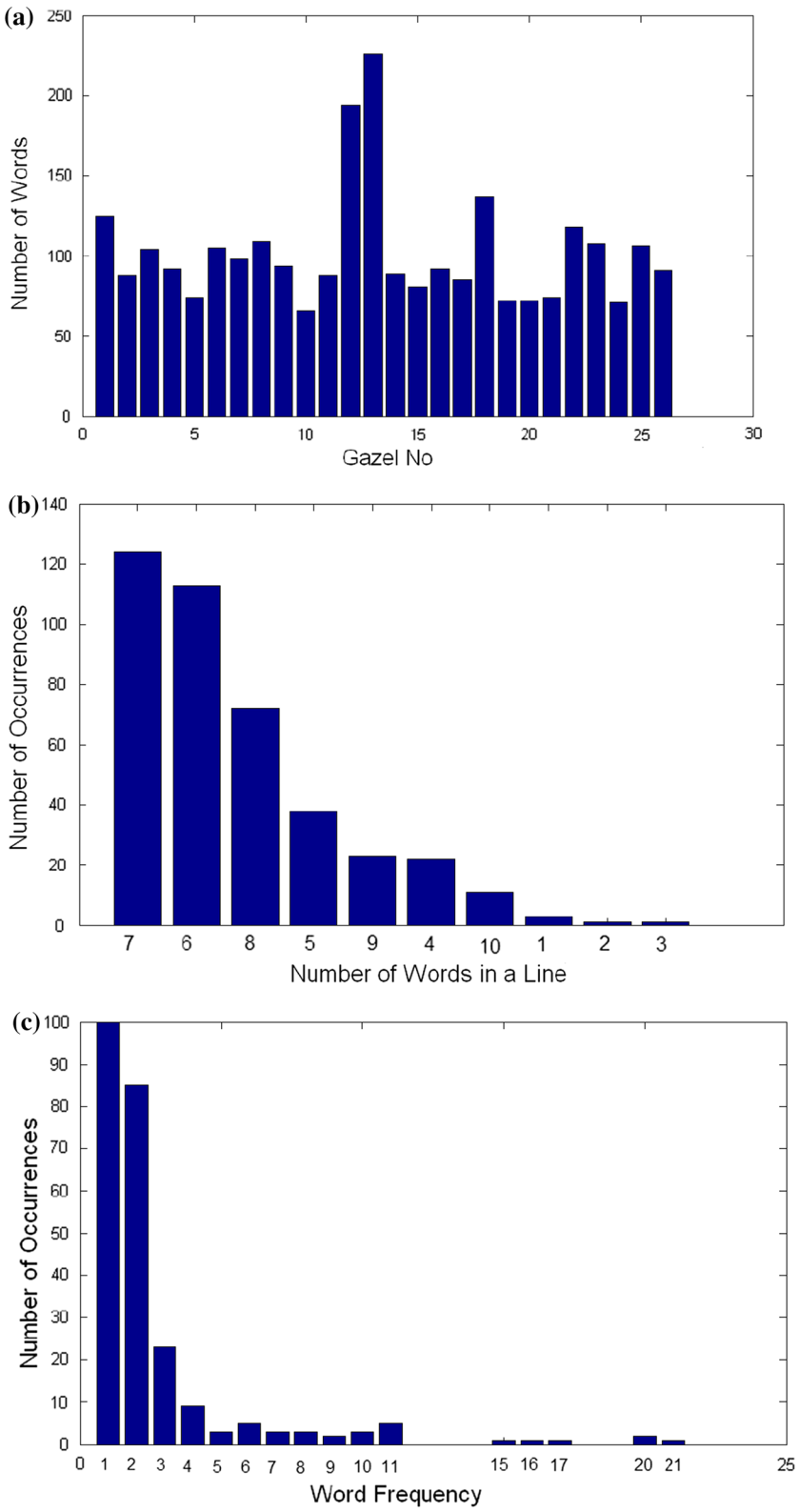


Fig. 6 In the source dataset **a** distribution of the number of words in each ghazal, **b** distribution of total words in lines **c** frequencies of words

correctly, which means some broken sub-word pieces are extracted as individual sub-words. For example, in the first 12 ghazals, 254 out of 2,937 sub-words are wrongly extracted. Also, 400 sub-words are wrongly divided into one more sub-word, 150 sub-words are divided into two more sub-words, while 35 sub-words are divided into

Table 1 Source and target datasets

	Source Machine-printed	Target Lithograph
Print Year	1,996	1,897
Number of Total Lines	408	402
Number of Total Words	2,688	2,640
Number of Unique Words	1,379	1,357
Number of Sub-words	5,964	6,186

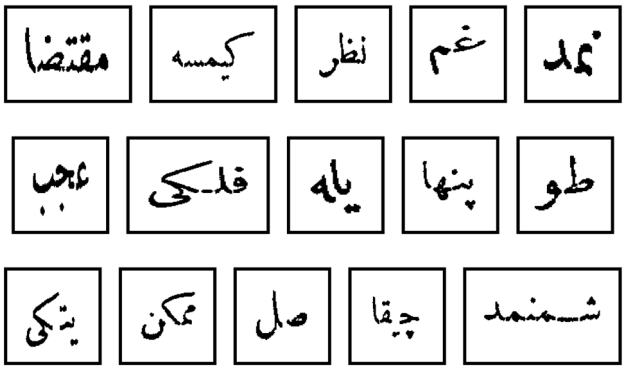


Fig. 7 Sub-word at *first row* can be easily connected by Manhattan distance approach; ones at *second row* can be connected by n-gram approach; and ones in *third row* cannot be connected

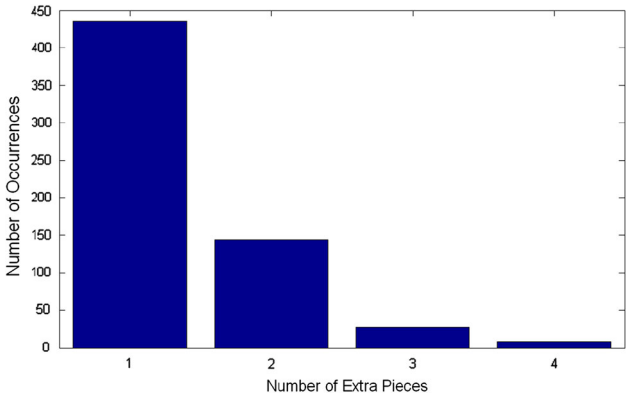


Fig. 8 Distribution of number of extra pieces that a sub-word is broken into

three extra sub-words because of broken characters. (see Figs. 7 and 8).

To evaluate the proposed method, the source and the target datasets are manually segmented into words to construct ground truth dataset. An automatically segmented word is counted as correct only if it is exactly the same as a word in the ground truth dataset. Precision (ratio of the number of correctly segmented words to the number of segmented words) and recall (ratio of the number of correctly segmented words to the number of words in the ground truth) values are used as matching scores.

Table 2 Results of vertical projection-based segmentation on source and target datasets for different threshold values

Threshold	Source		Target	
	Precision	Recall	Precision	Recall
6	0.81	0.83	0.55	0.44
7	0.87	0.86	0.59	0.44
8	0.87	0.85	0.59	0.44
9	0.85	0.83	0.59	0.43
10	0.82	0.77	0.59	0.43
11	0.80	0.74	0.60	0.43
12	0.76	0.68	0.47	0.36

4.2 Results of vertical projection-based word segmentation

The source dataset is automatically segmented into words with a simple vertical projection-based method. A vertical projection profile is obtained for each source line image. If the length of a white pixel group between two ink pixels is larger than a threshold *th*, it is assumed that this group of white pixels is an inter-word gap and used to define word boundaries.

To determine the best distance gap amount to use as the inter-word gap threshold, two pages of the source dataset are used and the words in them are manually segmented. The white pixel distances are calculated and the most frequent value is determined to be the inter-word distance. The best threshold is found as seven pixels in the source dataset. (see Table 2). Recall and precision values are found as 0.86 and 0.87, respectively, for the source dataset. Note that, a better word segmentation method is likely to increase the performance of the overall method. However, as will be discussed below, the differences between the manual and automatic segmentations are small when their effects on word matching are considered.

To provide a baseline, we also perform word segmentation using projection profiles on the target dataset. Highest recall and precision values for the target dataset are found as 0.43 and 0.60, respectively, for the threshold value of 11 pixels, and 0.44 and 0.59, respectively, for threshold value 7 which results in the best performance in source dataset (see Table 2). In both cases, the performance is significantly less than the performance on the source dataset. As seen, vertical projection is not successful in determining word boundaries in the target dataset.

4.3 Results of word segmentation in the target dataset

With the proposed method, after ghazal-based pruning step, the maximum number of candidate ghazals is reduced to 16 from 26, and the maximum number of candidate lines

Table 3 Word segmentation success rates on the target dataset for the proposed approach and the baseline methods in which vertical projection profile-based method or Run Length Smoothing Algorithm is applied on the target dataset (threshold 10). Queries are obtained from the source dataset either by manual segmentation (manual) or by a vertical projection-based segmentation method (automatic)

	Proposed approach		Baseline	
	Manual queries	Automatic queries	Vertical projection-based segmentation	Run length smoothing algorithm
Recall	0.70	0.65	0.43	0.50
Precision	0.74	0.68	0.60	0.53

The threshold value is chosen as 0.6 to define the similarity of two words

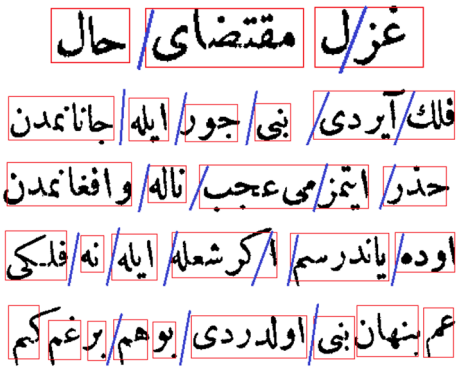


Fig. 9 RLSA segmentation results with a threshold 10. *Each box* shows groundtruth words and *lines* show the results by RLSA

is reduced to 140 from 402. Then, after line-based pruning, the maximum number of candidate lines for any line is further reduced to 90.

After finding the best matching lines among the remaining candidate lines, we segment the target lines into words by searching source words on the target line. We observed that a sub-word in the source dataset may be divided into at most four sub-words in the target dataset. We use this observation, and to limit the search space for word matching we search for the best alignment for a source word with n sub-words among at most $n + 4$ target sub-words.

As seen in Table 3, when automatically extracted source dataset words are used as query words, success rates decrease compared to manually segmented words because some query words may be extracted wrongly, which causes some words to be segmented wrongly, but the difference is not very large.

We compute the scores for perfect matches, that is a word is counted to be correctly segmented if it is exactly the same with the manual segmentation. This causes the relatively lower results, since there can be cases where small sub-words are attached to a wrong neighbour word, causing two words to be counted as wrongly segmented. For the line matching, these small mismatches are tolerated, since the overall similarity is considered for finding the best match. Also, in line matching, null word assignment is practically achieved by attaching the extra sub-words to its neighbours. When the overall line similarity is considered, this causes a negligible error. However, null words also may cause wrong word segmentations.

Note that, compared to the vertical projection profile-based method as discussed in Sect. 4.2, the increase in the performance is significant. We also use Run Length Smoothing Algorithm (RLSA) [45, 66] as another baseline to compare our proposed method, and have seen that RLSA is not able to capture the boundaries sufficiently as well (see Table 3 and Fig. 9).

Figure 10 shows word segmentation examples for some target lines when the vertical projection-based method and the proposed word segmentation method are used for the word segmentation. As seen, most word boundaries determined by the proposed method are correct, while the vertical projection-based method cannot detect word boundaries correctly.

4.4 Evaluation of word retrieval performances on the target dataset

When a user searches for a query word in an unsegmented collection, all dataset lines need to be searched by a sliding window approach. When a word segmentation schema is provided, however, a query word is searched for only in the set of segmented words, which speeds up the searching process. The word segmentation step takes time but done only once; thus, when a user wants to search thousands of query words, the time spent on word segmentation is negligible. However, wrongly segmented words effect word retrieval performance badly, proving the requirement for a good segmentation.

To understand the effects of word segmentation on word retrieval, we perform two sets of experiments. The first experiment is carried out on the source dataset and the second test is carried out on the target dataset.

In the first experiment, we analyse the intra-document word retrieval performances through searching a query word in the source dataset (i) among manually segmented words, and (ii) through a sliding window approach again in the source dataset. As shown in Table 4, when matching is performed over segmented words, the results are highly satisfactory. On the other hand, when segmentations are not available, and the search is done with a sliding window-based approach, precision decreases significantly. These results support the need for segmentation for a better retrieval.

Fig. 10 Example word segmentations in the target dataset. In *each box*, the *first row* shows the result of vertical projection-based segmentation method and the *second row* shows the result of proposed word-matching-based segmentation. *Lines* between the sub-words show the word boundaries found. For only the wrongly segmented ones, correct segmentations are shown in *rectangles*

In the second experiment, we test the performance of word retrieval on the target dataset for the following four different scenarios:

1. VPE: Target words are searched on a segmented dataset in which vertical projection-based method is used for segmentation. The number of segmented words is 1604.
2. UL: Target words are searched for in unsegmented target lines by a sliding window approach.
3. WME: Target words are searched on a segmented dataset in which the proposed word-matching-based method is used for segmentation. The number of segmented words is 2,200.
4. ME: Target words are searched for in the manually extracted target words set (comprising 2,640 words).

As seen in Table 5, the time required to search for a query word and the number of false matches increase when we use sliding window approach because the number of candidate words increases at each sliding iteration. On the other hand, when word boundaries are obtained with vertical projection-based segmentation method, many of the words are extracted wrongly, and therefore word retrieval success scores are not high. Note that, number of extracted words with vertical projection-based segmentation is less than the number of manually extracted words, since the vertical projection-based method cannot segment words correctly when there is little space between sub-words.

Table 4 Word retrieval success rates in the source dataset based on different matching score thresholds

Th	Segmented Dataset		Unsegmented Dataset	
	Recall	Precision	Recall	Precision
0.2	0.80	0.82	0.80	0.39
0.4	0.90	0.76	0.90	0.31
0.6	0.93	0.73	0.93	0.26
0.8	0.94	0.72	0.94	0.25

Words are tried to be spotted in both segmented and unsegmented source datasets

Table 5 Query retrieval success scores with a word-matching threshold of 0.4 and average query search times for four scenarios described in the text

Set	Recall	Precision	Average time (s)
VPE	0.50	0.43	151
UL	0.69	0.51	790
WME	0.76	0.70	233
ME	0.80	0.73	253

When word segmentation is done with the proposed word-matching-based method, the recall and precision values for retrieval are promising, and they are close to the performance of the manually segmented words, showing that the proposed cross-document word-matching-based method is able to provide good word boundaries.

Note that, even the upper limit for the retrieval performance (that can be achieved by manual segmentation) is relatively low, due to the limitations of the features used.

4.5 Cross-document word matching for handwritten Ottoman documents

We further analysed the proposed method for the cases where a printed document is used to segment its handwritten versions. For this purpose, we use three different versions of a page from an Ottoman divan by Fuzuli (see Fig. 11). First one is a printed version which is very clean and others are handwritten versions. The first handwritten version, referred to as easy handwritten, is relatively clean and easy to segment compared to the second handwritten version, referred to as hard handwritten. Note that, the dataset size is very small due to the difficulty of manual labelling, and this supports the motivation of the proposed study.

In the first experiment, printed version is used as a source document and manually segmented into words. There are 105 words in this version. Line matching is not performed in this small dataset but manually aligned (Note that some lines in printed version are missing in second version). Then, words are retrieved in the handwritten versions using these segmented words as queries. In total, out of 79, 69 words are correctly segmented from the easy handwritten. And in total, out of 79 words, 56 words are correctly segmented from the hard handwritten.

Then, easy handwritten version is used as a source document and these automatically extracted words are used as queries to be searched in the hard handwritten version. Out of 79 words, 61 words are correctly segmented in the hard version.

This experiment is important in proving the proposed segmentation transfer idea. Segmentation of a handwritten version which is not easy with the standard methods is handled by the help of a printed version, and then these segmentations are further utilised to segment more difficult versions. Segmentation on the difficult handwritten documents becomes possible with the information carried out from easier versions.

Note that, if we manually segment the easy handwritten version, and use it as a source document, without carrying

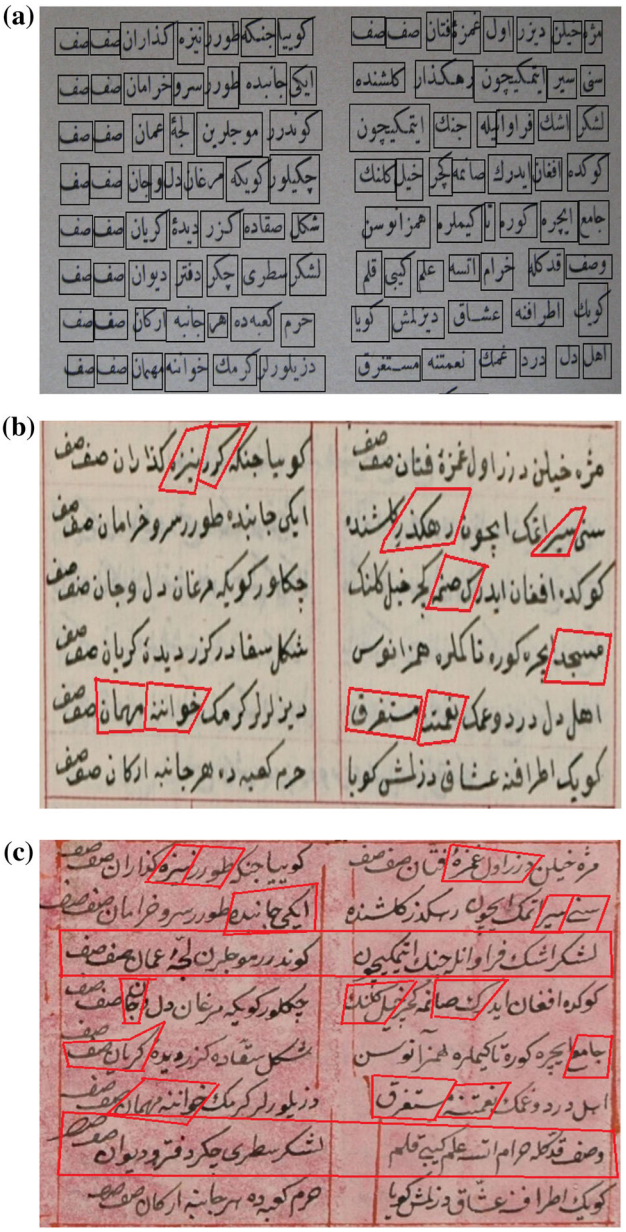


Fig. 11 First image is used a printed version of a poem, while second and third ones are handwritten versions which are from sixteenth century. Manually segmented words in printed version are used as query words to segment second version and these automatically extracted words in second version are used query words to segment third version into words this time. Mismatches are showed in *boxes* and reason for this is the connected sub-words and writing variations in different versions

information from printed documents, then out of 79 words, 8 of them cannot be segmented.

We give this toy example to show that cross-document word matching idea may also be used for word segmentation task across handwritten and printed documents. However, we noticed that DTW- based features are not robust enough to capture writing differences between handwritten documents; thus, better features should be considered for cross-document word matching on handwritten documents. Note also that we did not perform any correction for orientations, and this is one of the main reasons in mismatches. Further preprocessing will help to increase the performance, but this is out of scope for this study.

5 Summary and discussion

Addressing the requirements for vocabulary analysis and for finding the variants between versions and capturing the correct meaning of words, in this study we provide techniques for retrieving words in Ottoman divans. Word retrieval is more efficient and effective when segmented words are available. However, Ottoman documents are difficult to segment into words without a prior knowledge of the word. The prior knowledge, which is usually achieved by reading, is provided in this study by transforming the information from one version to another. An important outcome of the proposed method would be indexing and transcribing all copies through carrying manual labels provided for only a single copy.

In this study, simple profile-based features and DTW-based word matching method is used for finding the similarities of word matching. These features are chosen since they are commonly used in the word spotting literature, and they provide a baseline. We did not prefer to use features fine tuned to our datasets for . However, there are major drawbacks. Even within the same document these features are unable to provide satisfactory matching performances, and therefore it is the main bottleneck in the overall performance of the proposed method. In the future, we plan to focus on features that are robust to differences in writing styles and therefore can better capture the similarities among cross-document words.

Currently, we use machine-printed and lithograph versions. Lithographs are chosen since they are challenging for word segmentation: inter- and intra-word boundaries are not consistent and it is difficult to segment them into words based on spaces between components. However, the characters look alike to the ones in the machine-printed version, and therefore the features of the corresponding words in source and target datasets were relatively similar. Therefore, they provided a good testbed for our study. In the future, we plan to extend our approach to handwritten documents.

In this study, we use simple vertical projection profile-based method for segmenting words on the source dataset, and as a baseline for target dataset. There are a variety of word segmentation methods which are likely to increase the performance. However, this is not the focus of this study. Also, most of the available methods are likely to require parameter tuning for each different versions due to large variations between versions. Our main goal was to show that, detecting word boundaries in Ottoman documents is difficult since intra- and inter- gaps are not consistent and close to each other, therefore a prior information should be incorporated into word segmentation. In our study, this is achieved by matching words across documents.

References

1. Al-Badr BH (1995) A segmentation-free approach to text recognition with application to Arabic text, Ph.D. thesis. University of Washington, Seattle
2. Andrews WG (1985) Poetry voice, society song: ottoman lyric poetry. University of Washington Press, Seattle and London
3. Andrews WG, Black N, Kalpakli M (2006) Ottoman lyric poetry: an anthology. University of Washington Press
4. Anonymous (1897) Kulliyat-ı Divan-ı Fuzuli. Hurşid Matbaası, İstanbul
5. Asi A, Rabaev I, Kedem K, El-Sana J (2011) User-assisted alignment of Arabic historical manuscripts. In: International workshop on historical document imaging and processing
6. Ataer E, Duygulu P (2006) Retrieval of ottoman documents. In: Proceedings of the 8th ACM International workshop on Multimedia Information retrieval, pp. 155–162
7. Ataer E, Duygulu P (2007) Matching ottoman words: an image retrieval approach to historical document indexing. In: Proceedings of the 6th ACM International conference on Image and Video Retrieval, pp. 341–347
8. Ball G, Srihari SN, Srinivasan H (2006) Segmentation-based and segmentation-free methods for spotting handwritten Arabic words. In: 10th International Workshop on Frontiers in Handwriting Recognition
9. Brina CD, Niels R, Overvelde A, Levi G, Hulstijn W (2008) Dynamic time warping: a new method in the study of poor handwriting. Hum Mov Sci 27(2):242–255
10. Broumandnia A, Shanbehzadeh J, Varnoosfaderani MR (2008) Persian/Arabic handwritten word recognition using M-band packet wavelet transform. Image Vis Comput 26:829–842
11. Bulacu M, Schomaker L (2007) Text-independent writer identification and verification using textural and allographic features. IEEE Trans Pattern Anal Mach Intell 29:701–717
12. Can E, Duygulu P, Can F, Kalpakli M (2010) Redif extraction in handwritten Ottoman literary texts. In: Proceedings of the 20th International Conference on Pattern Recognition
13. Can EF, Duygulu P (2011) A line-based representation for matching words in historical manuscripts. Pattern Recognition Letters 32(8):1126–1138
14. Cheung A, Bennamoun M, Bergmann NW (2001) An Arabic optical character recognition system using recognition-based segmentation. Pattern Recognit 34(2):215–233

15. Dogan MN (1997) Mecnun ve Leyla Dilinden Siirler. Enderun Kitabevi (1997).
16. Fischer A, Indermuhle E, Frinken V, Bunke H (2011) HMM-based alignment of inaccurate transcriptions for historical documents. In: 11th Int. Conf. on Document Analysis and Recognition, p. 53
17. Fischer A, Keller A, Frinken V, Bunke H (2012) Lexicon-free handwritten word spotting using character HMMs. Pattern Recognit Lett 33:934–942
18. Fornés A, Lladós J, Sánchez G (2008) Old handwritten musical symbol classification by a dynamic time warping based method. Graph Recognit 5046:51–60
19. Fornes A, Llados J, Sanchez G, Karatzas D (2010) Rotation invariant hand-drawn symbol recognition based on a dynamic time warping model. Int J Doc Anal Recognit 13(3):229–241
20. Gatos B, Pratikakis I Segmentation-free word spotting in historical printed documents. In: Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, ICDAR '09, pp. 271–275
21. Howe NR, Rath TM, Manmatha R (2005) Boosted decision trees for word recognition in handwritten document retrieval. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 377–383. ACM
22. Huang C, Srihari SN (2008) Word segmentation of off-line handwritten documents. Document Recognition and Retrieval XV, Proc. SPIE 6815
23. Jain A (1986) Fundamentals of digital image processing. Prentice-Hall, Englewood Cliffs
24. Ji Y, Sun S (2013) Multitask multiclass support vector machines: model and experiments. Pattern Recognit pp. 914–924
25. Khurshid KCF, Vincent N (2012) Word spotting in historical printed documents using shape and sequence comparisons. Pattern Recognition 45:2598–2609
26. Kabacali A (1998) Cumhuriyet oncesi ve sonrasi matbaa ve basin sanayii. Cem Ofset
27. Kchaou MG, Kanoun S, Ogier JM (2012) Segmentation and word spotting methods for printed and handwritten arabic texts: a comparative study. In: International Conference on Frontiers in Handwriting Recognition
28. Khayyat M, Lam L, Suen CY (2012) Arabic handwritten word spotting using language models pp. 43–48
29. Khayyat M, Lam L, Suen CY (2014) Learning-based word spotting system for Arabic handwritten documents. Pattern Recognit 47(3):1021–1030
30. Kim S, Jeong S, Lee GS, Suen C (2001) Word segmentation in handwritten Korean text lines based on gap clustering techniques. In: Sixth International Conference on Document Analysis and Recognition, pp. 189–193
31. Konidaris T, Gatos B, Ntzios K, Pratikakis I, Theodoridis S, Perantonis SJ (2007) Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. Int J Doc Anal Recognit 9(2):167–177
32. Kut T, Ture F (1996) Yazmadan basmaya: muteferrika, muhendishane. Yapi Kredi Kultur Merkezi, Uskudar
33. Lados J, Rusinol M, Fornes A, Fernandes D, Dutta A (2012) On the influence of word representations for handwritten word spotting in historical documents. International J Pattern Recognit Artif Intell 26(05)
34. Leydier Y, Ouji A, LeBourgeois F, Emptoz H (2009) Towards an omnilingual word retrieval system for ancient manuscripts. Pattern Recognit 42(9):2089–2105
35. Lladós, J, Pratim-Roy P, Rodríguez JA., Sánchez G (2007) Word spotting in archive documents using shape contexts. In: Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part II, pp. 290–297. Springer-Verlag
36. Louloudis G, Gatos B, Pratikakis I, Halatsis C (2009) Text line and word segmentation of handwritten documents. Pattern Recognit 42(12):3169–3183
37. Manmatha R, Han C, Riseman E (1996) Word spotting: a new approach to indexing handwriting. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 631–637
38. Manmatha R, Han C, Riseman EM, Croft WB (1996) Indexing handwriting using word matching. In: Proceedings of the first ACM international conference on Digital libraries, pp. 151–159. ACM
39. Manmatha R, Srimal N (1999) Scale space technique for word segmentation in handwritten documents. Scale-Space Theories in Computer Vision. Lect Notes Comput Sci 1682:22–33
40. Marcolino A, Ramos V, Ramalho M, Pinto JC (2000) Line and word matching in old documents. In: Proceedings of the 5th IberoAmerican Symposium on Pattern Recognition, pp. 123–125
41. Marti UV, Bunke H (2001) Using a statistical language model to improve the performance of an HMM-Based cursive handwriting recognition system. Int J Pattern Recognit Artif Anal 15(1):65–90
42. Micah KE, Manmatha R, James A (2004) Text alignment with handwritten documents. In: DIAL '04: Proceedings of the First International Workshop on Document Image Analysis for Libraries, p. 195. IEEE Computer Society, Washington DC
43. Niels R (2004) Dynamic time warping: an intuitive way of handwriting recognition. Master's thesis
44. Nikolaou N, Makridis M, Gatos B, Papamarkos NSN (2010) Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. Image Vis Comput 28(4):590–604
45. Nikolaou N, Makridis M, Gatos B, Stamatopoulos N, Papamarkos N (2010) Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. Image Vis Comput. 28(4):590–604
46. Rath T, Manmatha R (2003) Word image matching using dynamic time warping. Proc IEEE Conf Computer Vis Pattern Recognit 2:521–527
47. Rath TM, Kane S, Lehman A, Partridge E, Manmatha R (2002) Indexing for a digital library of George Washington's manuscripts: a study of word matching techniques. Tech Rep
48. Rath TM, Lavrenko V, Manmatha R (2003) A statistical approach to retrieving historical manuscript images without recognition. Tech rep
49. Rath TM, Manmatha R, Lavrenko V (2004) A search engine for historical manuscript images. In: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval, pp. 369–376. ACM
50. Rodríguez-Serrano JA, Perronnin F (2009) Handwritten word-spotting using hidden Markov models and universal vocabularies. Pattern Recognit 42(9):2106–2116
51. Rothfeder JL, Feng S, Rath TM (2003) Using corner feature correspondences to rank word images by similarity. Comput Vis Pattern Recognit Workshop 3:30–36
52. Saykol E, Sinop A, Gudukbay U, Ulusoy O, Cetin A (2004) Content-based retrieval of historical Ottoman documents stored as textual images. IEEE Trans Image Process 13(3):314–325
53. Seni G, Cohen E (1994) External word segmentation of off-line handwritten text lines. Pattern Recognit 27:41–52
54. Sinno JP, Qiang Y (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
55. Srihari SN, Ball GR (2008) Language independent word spotting in scanned documents. In: Proceedings of the 11th International Conference on Asian Digital Libraries, pp. 134–143
56. Sun S, Xu Z, Yang M (2013) Transfer learning with part-based ensembles. Lect Notes Comput Sci 7872:271–282

57. Adamek TN, Smeaton A (2007) Word matching using single closed contours for indexing handwritten historical documents. Int J Doc Anal Recognit 9:153–165
58. Tomai CI, Zhang B, Govindaraju V (2002) Transcript mapping for historic handwritten document images. In: 8th International Workshop on frontiers in Handwriting Recognition
59. Tseng YH, Lee HJ (1999) Recognition-based handwritten Chinese character segmentation using a probabilistic viterbi algorithm. Pattern Recognit Lett 20(8):791–806
60. Tu W, Sun S (2012) Cross-domain representation-learning framework with combination of class-separate and domain-merge objectives. In: Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining, CDKD '12, pp. 18–25. ACM
61. Varga T, Bunke H (2005) Tree structure for word extraction from handwritten text lines. In: 8th International Conference on Document Analysis and Recognition , vol. 1, pp. 352–356
62. Yalniz I, Altingovde I, Gudukbay U, Ulusoy O (2009) Integrated segmentation and recognition of connected Ottoman script. Opt Eng 48(11):1–12
63. Yalniz I, Altingovde I, Gudukbay U, Ulusoy O (2009) Ottoman archives explorer: a retrieval system for digital Ottoman archives. J Comput Cult Herit 2(3):1–20
64. Zand M, Naghsh A, Monadjemi A (2008) Recognition-based segmentation in Persian character recognition. In: Proceedings of the Second International Conference on Advances in Pattern Recognition. World Academy of Science, Engineering and Technology 38
65. Zhang B, Srihari SN, Huang C (2003) Word image retrieval using binary features. Doc Recognit Retr XI 1:45–53
66. Zirari F, Ennaji A, Nicolas S, Mammass D (2013) A methodology to spot words in historical arabic documents pp. 1–4