ETHEM FATIH CAN

A LINE-BASED REPRESENTATION FOR MATCHING WORDS

CONTENTS

INTRODUCTION MOTIVATION

- The excessive number of documents in digital environment has increased →
  - Efficient access to historical documents
- Word Spotting techniques, alternative to OCR systems based systems
- Poor quality of historical documents, and variety of scripts

INTRODUCTION RELATED STUDIES

- Manmatha et al. (1996, 2003a, 2003b)
  - Projection profiles, word profiles, and background/ink transitions
  - Dynamic time warping (DTW)
- Rath and Manmatha, 2007
  - Employ clustering to recognize words

INTRODUCTION RELATED STUDIES

- Adamek et al., 2007
  - Contour-based approach to match the words
  - Contours are extracted after several processes, including binarization, and removing artifacts.
  - Multi-scale convexity concavity (MCC) representation with DTW

INTRODUCTION RELATED STUDIES

- Ataer and Duygulu, 2006
  - Extract interest points from word images using scale invariant feature transform (SIFT) operator (Lowe, 2004)
  - A codebook obtained by the vector quantization of SIFT descriptors is then used to represent and match the words.
  - The method is tested on Ottoman documents

INTRODUCTION $_{\text{APPROACH}}$

- Line-based representation
- Two matching criteria
  - Word matching task
    - Word Image matching using Line Descriptors (WILD)
  - *Redif* Extraction task
    - *Redif* Extraction using Contour Segments (RECS)

INTRODUCTION CONTRIBUTION

- Effective and efficient representation of words images based on line descriptors
- Two approaches for word matching and *redif* extraction task (WILD and RECS)
- Tested on different languages; English and Ottoman,
- Provides promising results without the need of pre-processing steps in word matching task
- Multi-scale analysis in word matching task
- A pioneering image-based automatic *redif* extraction method, first in the literature (RECS)

- Introduction
- Line-based Word Representation
  - Binarization
  - Extraction of Contour Segments
  - Line Approximation
  - Line Description
- Word Matching Task: Word Image Matching using Line Descriptors (WILD)
- *Redif* Extraction Task: *Redif* Extraction using Contour Segments (RECS)
- Experimental Results
- Conclusion

LINE-BASED WORD REPRESENTATION

- Motivation
  - Words consist of lines and curves
  - Success of using line segments as descriptors for object recognition (Ferrari et al., 2008)
- Words are described using line segments extracted from the contours of images

LINE-BASED WORD REPRESENTATION

- BINARIZATION

LINE-BASED WORD REPRESENTATION

- EXTRACTION of CONTOUR SEGMENTS
  - Connected components → 8-neighbors

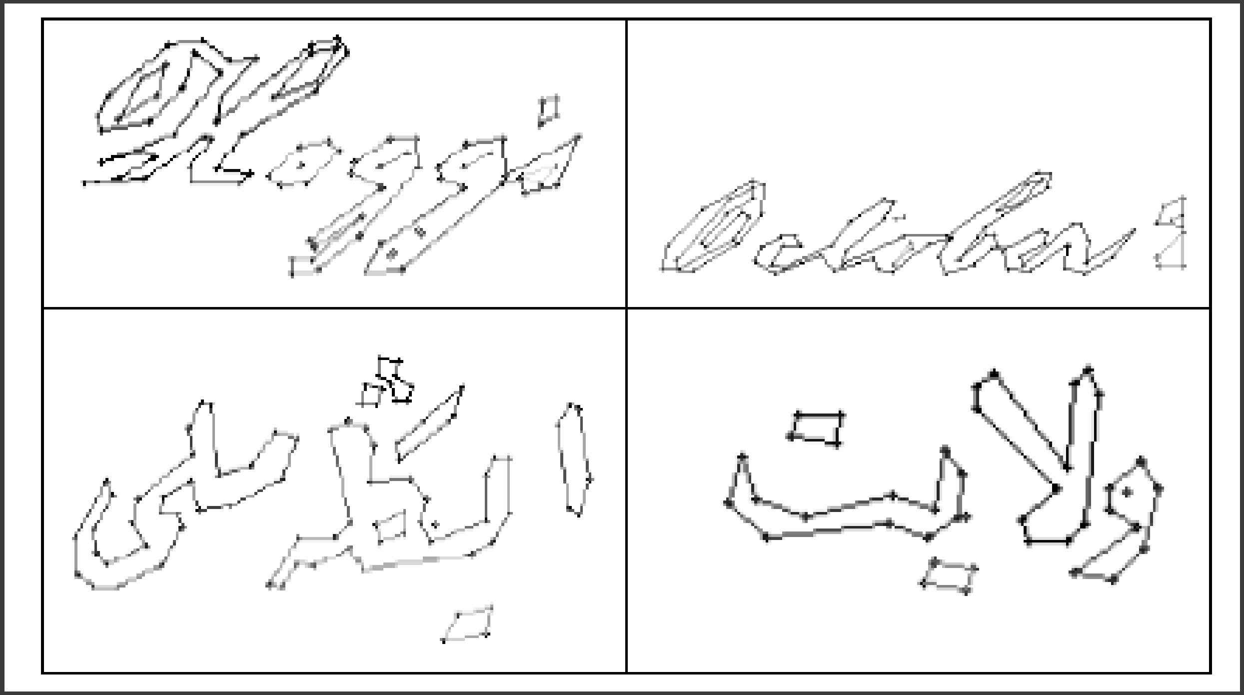
LINE-BASED WORD REPRESENTATION

- LINE APPROXIMATION
  - Fit lines to the points of the contour segments, instead of using the contour itself.
  - Line approximation is performed using Douglas-Peucker[1,2] algorithm
  - The parameter τ, as approximation accuracy

[1] Douglas and Peucker, 1973,
[2] Hershberger and Snoeyink, 1992

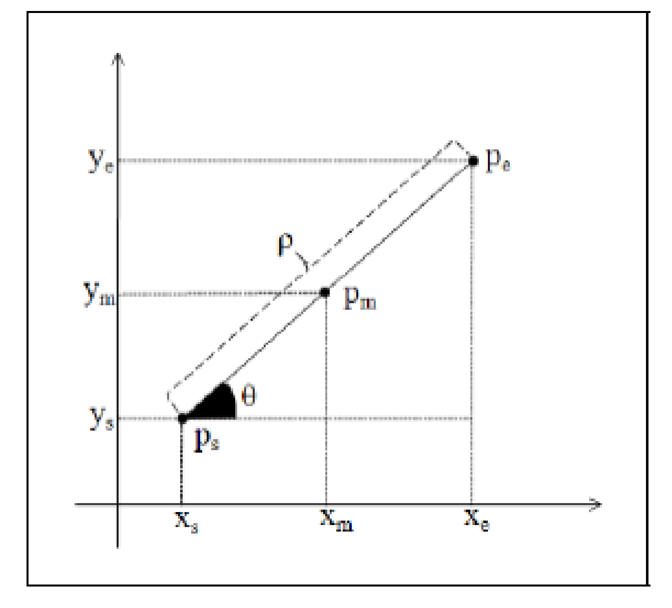
LINE-BASED WORD REPRESENTATION

- LINE APPROXIMATION



LINE-BASED WORD REPRESENTATION

- LINE DESCRIPTION
  - A line is described using the position, orientation, and length information as in (Ferrari et al., 2008)
  - $\ell = \{p_s, p_m, p_e, \theta, \rho\}$



- Introduction
- Line-based Word Representation
- Word Matching Task: Word Image Matching using Line Descriptors (WILD)
  - Line Matching
  - Word Matching
- *Redif* Extraction Task: *Redif* Extraction using Contour Segments (RECS)
- Experimental Results
- Conclusion

Word Matching Task: Word Image matching using Line Descriptors (WILD)

- Each word image I is represented as a set of line descriptors as:
  - $\mathrm{I} = \{\ell_1, \ell_2, \ldots, \ell_N\}$, N is the number of line descriptors.

Word Matching Task

- Representative points of each line descriptor are re-arranged as:

$$X_I = \frac{\sum x_m^i}{N},\ Y_I = \frac{\sum y_m^i}{N}, i = 1 \ldots N,$$

$$p_s' = (x_s - X_I, y_s - Y_I)$$
$$p_m' = (x_m - X_I, y_m - Y_I)$$
$$p_e' = (x_e - X_I, y_e - Y_I)$$

- We prefer to use only the mid-point and refer to it as r.

Word Matching Task

- LINE MATCHING
  - The distance between two line descriptors ℓ_a and ℓ_b are computed as:
  - $d(\ell_a , \ell_b) = 4d_r + 2d_\theta + d_\ell$

  where $d_r = |r_a - r_b|$, $d_\theta = |\theta_a - \theta_b|$,

  and $d_\ell = |\log(\rho_a, \rho_b)|$

  r_a and r_b →mid-points of the line descriptors

  θ_a and θ_b →orientations of the line descriptors[0,Π]

  ρ_a and ρ_b → lengths of the lines

Word Matching Task

- WORD MATCHING
  - Based on the distances between line descriptors
  - I_a and I_b are the word images having N_a and N_b line descriptors
  - For each line descriptor in I_a , we search for the best matching line in I_b .

That is; (ℓ_i^a, ℓ_j^b) is a matching pair, if $d(\ell_i^a, \ell_j^b) < d(\ell_i^a, \ell_k^b)\ \forall_k$, $j \neq k,\ k = 1, 2, ..., N_b$.

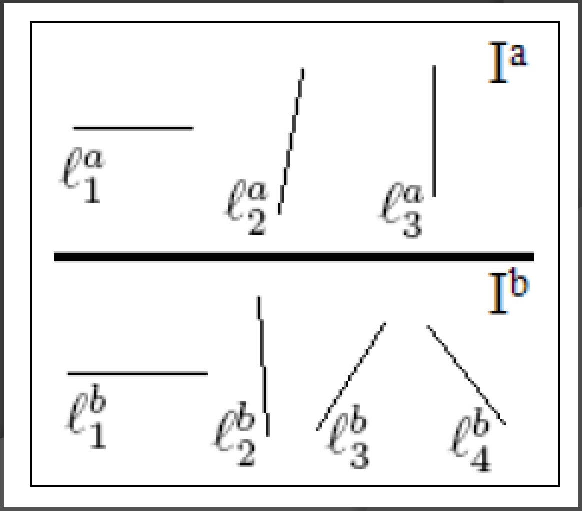
Word Matching Task

- WORD MATCHING
  - If two or more line descriptors in I_a match to a single line in I_b then we choose the one with the minimum distance and eliminate the others.

For example; $I^a = \{\ell_1^a, \ell_2^a, \ell_3^a\}$ $\quad I^b = \{\ell_1^b, \ell_2^b, \ell_3^b, \ell_4^b\}$

the minimum matches are $\{(\ell_1^a, \ell_3^b), (\ell_2^a, \ell_2^b), (\ell_3^a, \ell_2^b)\}$,

$$D_{a,b} = d(\ell_1^a, \ell_3^b) + min(d(\ell_2^a, \ell_2^b), d(\ell_3^a, \ell_2^b)).$$



Word Matching Task

- WORD MATCHING
  - To compute the final score
    - In addition to distance of matched pairs of line descriptors
    - The number of hits $h_{a,b}$ as the number of matches between two images
    - The number of line descriptors in the images N_a and N_b

$$f(I^a, I^b) = (D_{a,b})\left(\frac{(N_a - h_{a,b})^2 + (N_b - h_{a,b})^2}{\sqrt{[(N_a)^2 + (h_{a,b})^2][(N_b)^2 + (h_{a,b})^2]}}\right)$$

Word Matching Task

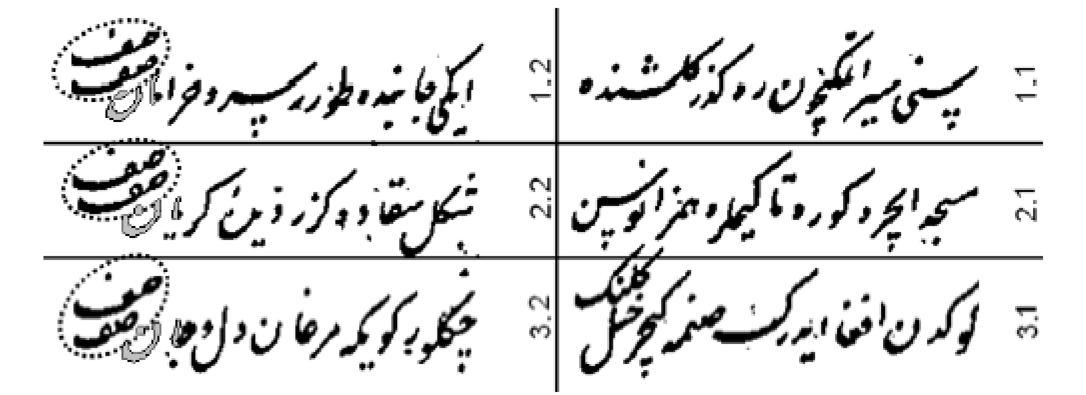
- WORD MATCHING
  - a global distance matrix **F (Q by Q)**
  - **F**(a,b) = f(I_a , I_b), Q is the number of images in test bed
  - F(1,3) is the dissimilarity value between the first and third image in the data set.

- Introduction
- Line-based Word Representation
- Word Matching Task: Word Image Matching using Line Descriptors (WILD)
- *Redif* Extraction Task: *Redif* Extraction using Contour Segments (RECS)
- Experimental Results
- Conclusion

Redif Extraction Task: *Redif* Extraction using Contour Segments (RECS)

- In Ottoman (Divan) poetry, most of the
  - poems are based on a pair of lines, couple ir distich.
  - Distich: two hemistichs (lines)
  - Hemistichs of the same distich completes each other
  - The rhyme and *redif* are used to provide the integrity of the distichs of a poem and provide a melody to its voice.
  - The *redif* can be explained as the repeated patterns following the rhyme in a poem.

Redif Extraction Task



1.1 Seni seyr itmek içün reh-güzer-i gülşende
1.2 îki cânibde durur serv-i hırâmân saf saf

2.1 Mescid içre göre tâ kimlere hemzânûsın
2.2 Şekl-i sakkada gezer dide-i giryân saf saf

3.1 Gökde efgân iderek sanma geçer hayl-i küleng
3.2 Çekilür kûyune mürgân-ı dil ü cân saf saf

Redif Extraction Task

- In the method, we automatically extract the *redifs* in the handwritten literary Ottoman text images.
- Un-segmented images are used
- Unlike word spotting studies, the most basic unit is not the word image but contour segment C which may represent a word, a character, and a sequence of characters.

Redif Extraction Task

- Method can be summarized as:
  - Normalization of line descriptors with reference line

$$X_C = \frac{\sum x_m^i}{n}, Y_C = \frac{\sum y_m^i}{n}, i = 1, 2, ..., n$$

$$\ell^r = (p_m^r, \theta^r, \rho^r)$$

$$\ell' = (p'_m, \theta', \rho') \qquad \begin{aligned} x'_m &= x_m - x_m^r \\ y'_m &= y_m - y_m^r \\ \theta' &= \theta - \theta^r \\ \rho' &= \rho / \rho^r \end{aligned}$$

Redif Extraction Task

- Contour segment descriptors are defined C′

$$C' = \{\ell'_1, \ell'_2, \ldots, \ell'_n\}$$

- Construction of codebook B

$$B = \{b_1, b_2, \ldots, b_k\}$$

- Represent contour segment descriptors as a sequence of elements of codebook

$$C' = \{b_{\ell'_1}, b_{\ell'_2}, \ldots, b_{\ell'_n}\} \quad \text{where } b_{\ell'_i} \in B,\ b_{\ell'_i} \text{ is the code of } \ell'_i, \text{ and } i = 1, 2, \ldots, n.$$

Redif Extraction Task

- We compute the distance between two contour segment descriptors each consisting of elements of the codebook
  - The difference is the sum of insertions, deletions, and substitutions.

$$C'_i = \{b_1, b_2, b_3, b_4\} \qquad C'_j = \{b_1, b_3, b_3\}$$

- Second code of the C'_i should be deleted
- b_4 should be substituted with b_3

Redif Extraction Task

- *REDIF* EXTRACTION
- Rules of the *redif*
- A redif must appear
  - At the end of the second hemistich –line- of a distich –couple- (constraint 1)
  - In every distich (constraint 2)

Redif Extraction Task

- *REDIF* EXTRACTION
  - According to constraint 1
    - The x positions of the redifs should roughly be the same and they should be close to the left border (end of the last hemistich)
    - We eliminate the contour segments that do not appear in the left (last) part of the distichs
      - $X < \alpha_1(w)$, w is the width of the image α_1 in [0,1]
    - Among the remaining ones, a contour segment and its matches are need to be vertically aligned to be counted as a *redif*
    - For a contour segment, we check each of its matches whether they are vertically aligned. We ignore the rest of the matches for the segment if not.

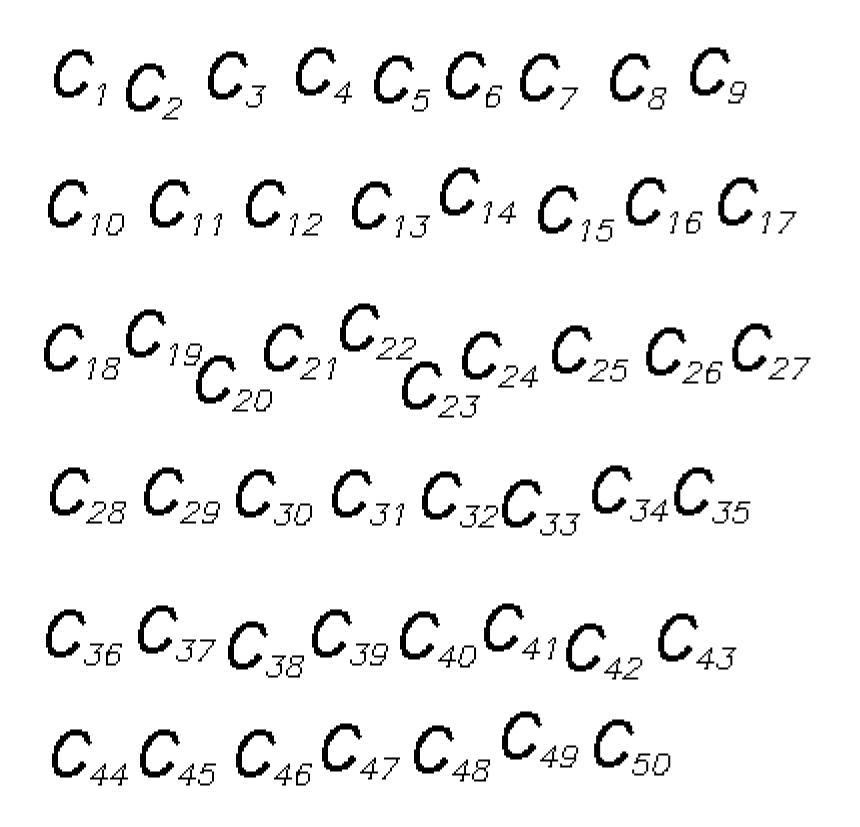
Redif Extraction Task

- *REDIF* EXTRACTION
  - According to constraint 1
    - Two contour segments are vertically aligned;
      - If the distance in x positions $\alpha_2(w)$, α_2 in [0,1]
    - α_1 and α_2 are emprically determined; 0.25 and 0.15
  - We check the remaining contour segments and their matches, we check the number of matches for each remaining contour segment to satisfy constraint 2.
  - Minimum number of matches should be 5

Redif Extraction Task

- We search for the contour segments that have one or more common matches and take the union of the matches of those contour segments, and we perform this operation until any pair of contour segments has a common match.
- Having combined the common matches not to extract the same contour segment as *redif*, we check whether the contour segment extracted as *redif* has more than five matches, and if so we count that contour segment as *redif*

Redif Extraction Task



$C_1 \rightarrow \{ C_1, C_{10}, C_{18}, C_{28}, C_{44}, C_{50}, C_{36}, \ldots\}$

$C_2 \rightarrow \{ C_2, C_{11}, C_{19}, C_{29}, C_{37}, C_{45}, C_9, C_{17}, \ldots\}$

$C_3 \rightarrow \{C_3, C_{12}, C_{20}, C_{30}, C_{35}, C_{38}, C_{46}, \ldots.\}$

$C_4 \rightarrow \{C_4, C_5, C_{12}, C_{41}, C_{42}, C_{43}, \ldots.\}$

$C_7 \rightarrow \{ C_7, C_{15}, C_{25}, C_{34}, C_{42}, C_{49}, C_{24}, C_{32}, \ldots\}$

...

$C_{18} \rightarrow \{ C_{18}, C_{10}, C_1, C_{28}, C_{36}, C_{44}, C_{50}, \ldots\}$

$C_{19} \rightarrow \{ C_{19}, C_{37}, C_{45}, C_{29}, C_2, C_{11}, C_{17}, \ldots\}$

...

Redif Extraction Task

- According to constraint 1
  - We do not consider the contour segment descriptors like C_{50}, C_{26}, C_9, … as a *redif* candidate. Since they do not satisfy
    - $X < \alpha_1(w)$
  - After checking contour segments should be vertically aligned
    - $\alpha_2(w)$

Redif Extraction Task

- The candidate list turns out to be

$C_1 \rightarrow \{ C_1, C_{10}, C_{18}, C_{28}, C_{44}, C_{50}, C_{36}, \ldots\}$

$C_2 \rightarrow \{ C_2, C_{11}, C_{19}, C_{29}, C_{37}, C_{45}, C_9, C_{17}, \ldots\}$

$C_3 \rightarrow \{C_3, C_{12}, C_{20}, C_{30}, C_{35}, C_{38}, C_{46}, \ldots.\}$

$C_4 \rightarrow \{C_4, C_5, C_{12}, C_{41}, C_{42}, C_{43}, \ldots.\}$

...

$C_7 \rightarrow \{ C_7, C_{15}, C_{25}, C_{34}, C_{42}, C_{49}, C_{24}, C_{32}, \ldots\}$

...

$C_{18} \rightarrow \{ C_{18}, C_{10}, C_1, C_{28}, C_{36}, C_{44}, C_{50}, \ldots\}$

$C_{19} \rightarrow \{ C_{19}, C_{37}, C_{45}, C_{29}, C_2, C_{11}, C_{17}, \ldots\}$

...

Redif Extraction Task

- Then we search for the contour segment descriptors having common matches, and take the union of them.

$C_1 \rightarrow \{ C_1, C_{10}, C_{18}, C_{28}, C_{44}\}$ and $C_{18} \rightarrow \{ C_{18}, C_{10}, C_1, C_{28}, C_{36}, C_{44}\}$

$C_2 \rightarrow \{ C_2, C_{11}, C_{19}, C_{29}, C_{37}, C_{45}\}$ and $C_{19} \rightarrow \{ C_{19}, C_{37}, C_{45}, C_{29}, C_2, C_{11}\}$

then we have

$C_1 \rightarrow \{ C_1, C_{10}, C_{18}, C_{28}, C_{44}, C_{36}\}$

$C_2 \rightarrow \{ C_2, C_{11}, C_{19}, C_{29}, C_{37}, C_{45}\}$

$C_3 \rightarrow \{C_3, C_{12}, C_{20}, C_{30}\}$

$C_4 \rightarrow \{C_4, C_5, C_{12}, C_{41}\}$

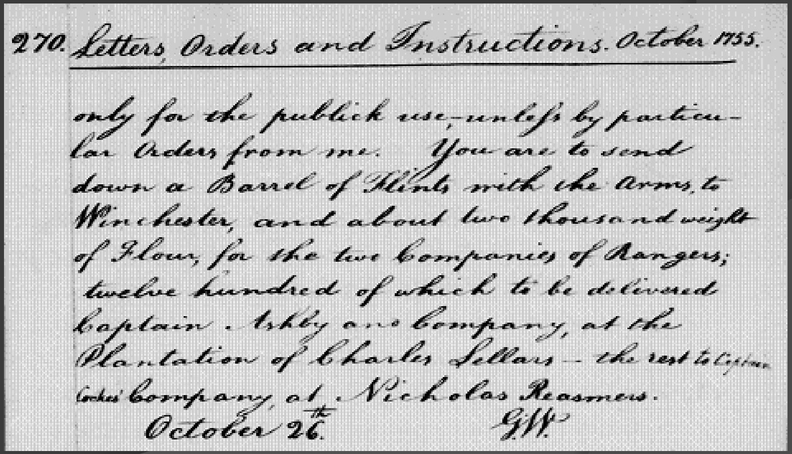
Redif Extraction Task

- According to constraint 2
  - Each candidate should have at least 5 or more matches
    - Then we only have C_1 and C_2
    - $R = \{C_1, C_2\}$

- Introduction
- Line-based Word Representation
- Word Matching Task: Word Image Matching using Line Descriptors (WILD)
- *Redif* Extraction Task: *Redif* Extraction using Contour Segments (RECS)
- Experimental Results
  - Data sets
  - Evaluation Criteria
  - Results
- Conclusion

EXPERIMENTAL RESULTS

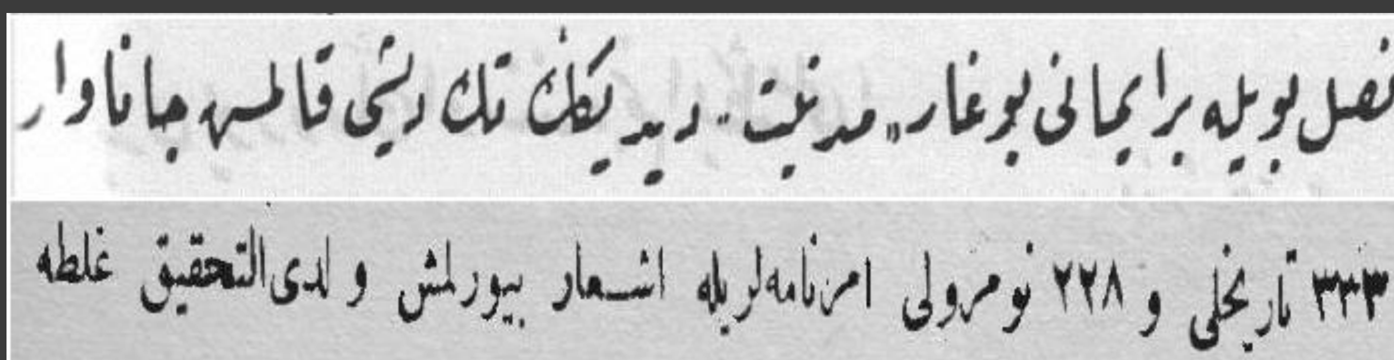
- Word Matching
  - DATA SET
    - Images from George Washington (GW) Collection in Library of Congress, which is used as a benchmark data set in word spotting studies
      - Ten pages of GW (GW10)
        - 2381 words
      - Twenty pages of GW (GW20)
        - 4860 words


270. Letters Orders and Instructions. October 1755.

only for the publick use,—unless by particu-
lar Orders from me. You are to send
down a Barrel of Flints with the Arms, to
Winchester, and about two thousand weight
of Flour, for the two Companies of Rangers;
twelve hundred of which to be delivered
Captain Ashby and Company, at the
Plantation of Charles Sellars—the rest to Captain
Cockes' Company, at Nicholas Reasmens.
October 26th. G.W.

EXPERIMENTAL RESULTS

- Word Matching
  - DATA SET
    - Ottoman sets used in Ataer and Duygulu, 2006 - 2007
      - Three pages – 257 words OTM1
      - Six pages – 823 words OTM2
      - Combination of OTM1 and OTM2 (OTM1+2)



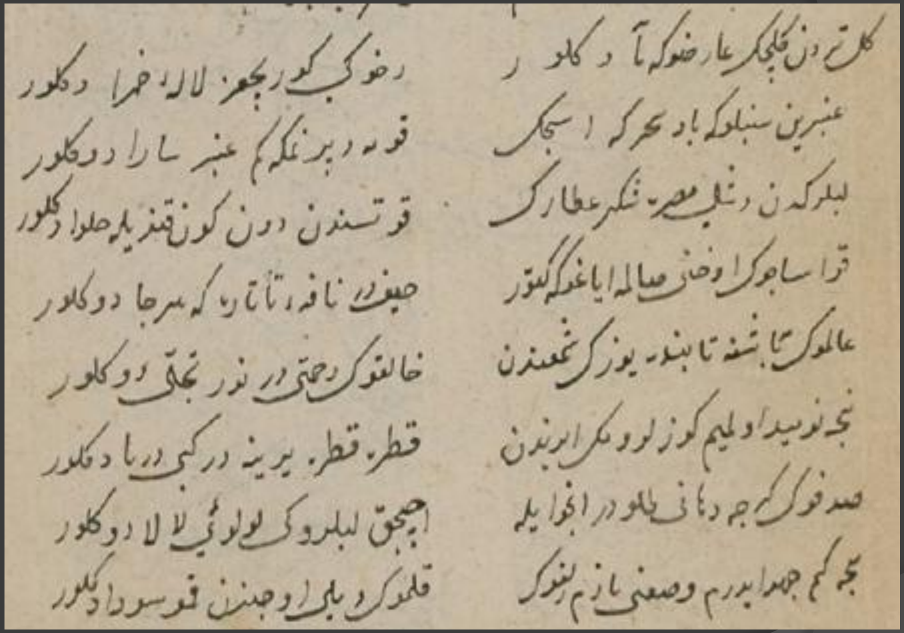
EXPERIMENTAL RESULTS

- Word Matching
  - EVALUATION CRITERIA
    - Retrieval
      - trec_eval package – precision-recall
    - Recognition
      - Word Error Rate

$$WER = 1 - \left(\frac{\#correct\ matches\ in\ test\ page}{\#words\ in\ test\ page}\right)$$

EXPERIMENTAL RESULTS

- *Redif* Extraction
  - DATA SET
    - 100 poems from
      - 15th to 19th centuries
    - Turkey Manuscripts[1]
    - Ottoman Text Archive Project (OTAP)[2]



[1] www.yazmalar.gov.tr
[2] courses.washington.edu/otap/

EXPERIMENTAL RESULTS

- *Redif* Extraction
  - EVALUATION CRITERIA

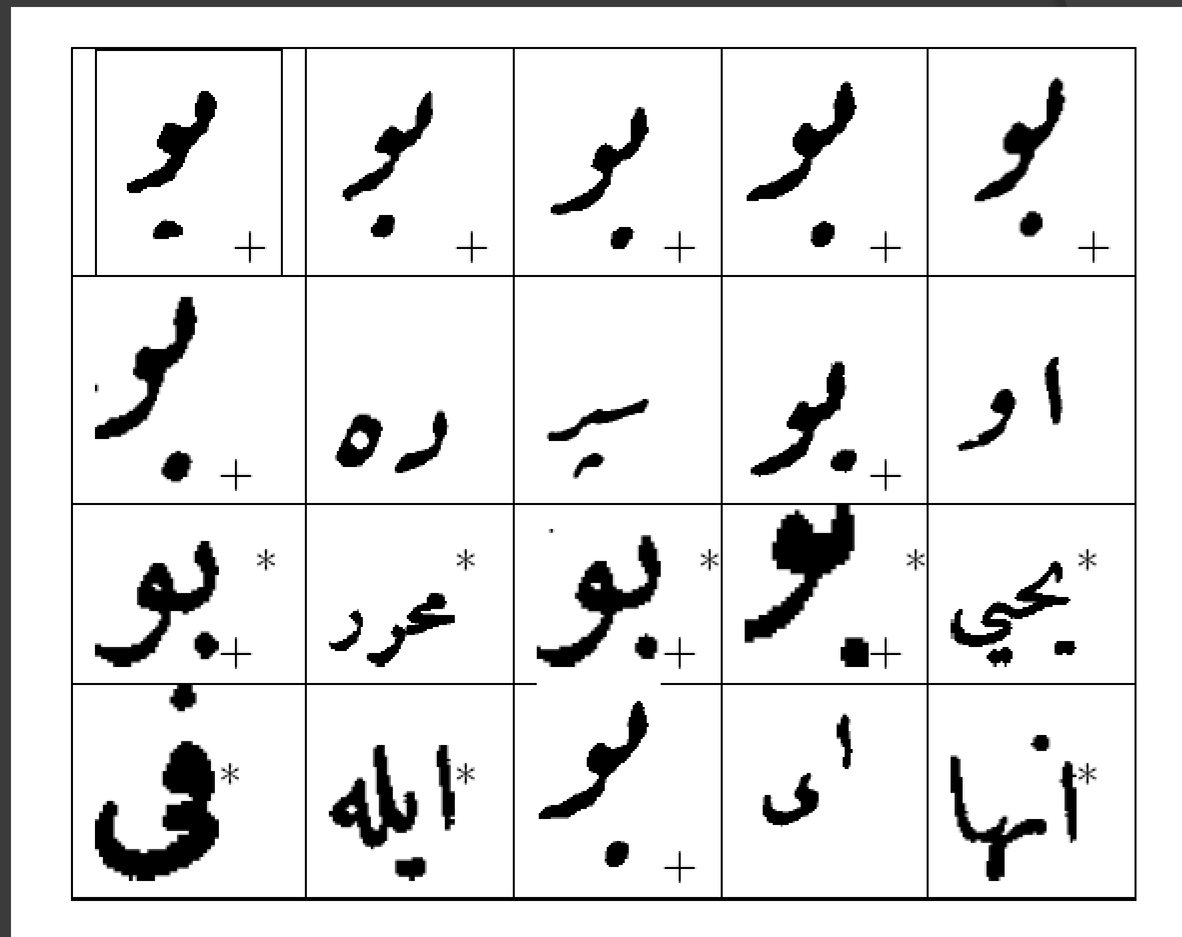
$$R : extracted\ redifs$$
$$R_{gt} : ground\ truth\ for\ redifs$$
$$ER : extraction\ rate\ \in\ [0,1]$$
$$ER = \frac{\#correct\ extractions\ in\ R}{max(sizeof(R_{gt}), sizeof(R))}$$

EXPERIMENTAL RESULTS

- Word Matching (GW data sets)

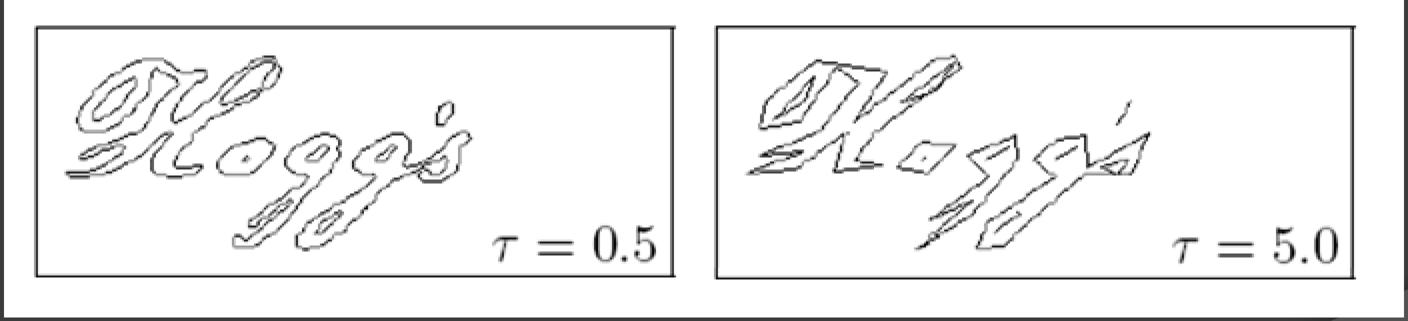
EXPERIMENTAL RESULTS

- Word Matching (OTM data sets)



EXPERIMENTAL RESULTS

- Word Matching
  - Considering results of different τ values
  - Simply adding the distance matrices of different τ values



EXPERIMENTAL RESULTS

Method	Data set	Precision	Recall
our approach	GW10	0.688	1.000
our approach	GW10	0.774	0.770
DTW (Rath and Manmatha [23])	GW10	0.653	0.711
DTW (Rath and Manmatha [22])	GW10	0.726	0.652
our approach	GW20	0.566	1.000
our approach	GW20	0.667	0.673
DTW (Rath and Manmatha [22])	GW20	0.518	0.550
our approach	OTM1	0.987	1.000
bag-of-words (Ataer and Duygulu [4])	OTM1	0.910	1.000
DTW (Ataer and Duygulu [4][1])	OTM1	0.940	1.000
our approach	OTM2	0.944	1.000
bag-of-words (Ataer and Duygulu [4])	OTM2	0.840	1.000
our approach	OTM1+2	0.957	1.000
bag-of-words (Ataer and Duygulu [4])	OTM1+2	0.810	1.000

EXPERIMENTAL RESULTS

Method	WER	WER w/o OOV words	Language model post-processing
our approach	0.303	0.189	-
Adamek et al. [1]	0.306	0.174	-
Lavrenko et al. [14]	0.449	0.349	+


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

EXPERIMENTAL RESULTS

- *Redif* Extraction
  - In the collection of 100 poems
  - We obtain a score of 0.682, when k is set to 45

EXPERIMENTAL RESULTS

- *Redif* Extraction
  - For large values of k,
    - our method is able to extract more complicated *redifs*
    - it misses more number of *redifs*
  - If we decrease the minimum number of matches to be counted as *redif* , our method extracts
    - more number of correct contour segments as redifs; however,
    - the number of false matches also increases
  - For higher values of the same parameter,
    - the number of false matches decreases;
    - the number of misses increases.

- Introduction
- Line-based Word Representation
- Word Matching Task: Word Image Matching using Line Descriptors (WILD)
- *Redif* Extraction Task: *Redif* Extraction using Contour Segments (RECS)
- Experimental Results
- Conclusion

CONCLUSION

- Line-based representation schema
  - Two matching criteria using the proposed representation schema; WILD, and RECS
- In most of the cases, WILD outperforms the results of the existing studies
- RECS provides promising results and motivation for future studies.