The Iby and Aladar Fleischman Faculty of Engineering
The Zandman-Slaner School of Graduate Studies

Word Spotting Applications for Historical Documents

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science
in the School of Electrical Engineering, Tel Aviv University

by
Adi Silberpfennig

The research for this thesis has
been carried out under the supervision of
Prof. Lior Wolf

February 2017

Contents

Acknowledgments

I would like to express my gratitude to my advisor Prof. Lior Wolf for his guidance and support. I am very thankful for having the opportunity to learn from him, through his excellent courses, and our research.

A special and warm thanks goes to Prof. Nachum Dershowitz, who guided me during my research. Thank you for the help, support and advices through all the stages of this work.

I would like to thank Professor Yaron Tsur, the Jewish Historical Press Project and the National Library for giving me access to the JPRESS archive data and for giving me the support that was needed. I would also like to thank the Friedberg Jewish Manuscript Society for giving me the access to the Cairo Genizah ,and for integrating the word spotting engine in the website. I would like to thank Daniel Labenski for the joint work in the JPRESS chapter of my research.

Last but not least, I would like to thank my husband, Moti, for all his love and support.

This work was supported in part by the Israel Science Foundation (ISF grant 1330/14), the German-Israeli Foundation for Scientific Research and Development (GIF grant #I-145-101.3-2013), the Deutsch-Israelische Projektkooperation (DIP grant #01019841), the Friedberg Genizah Project, and the Blavatnik Family Fund.

Abstract

Historical documents have been undergoing large-scale digitization over the past years, bringing massive image collections available on-line. This provides scholars with a simple and fast access to cultural heritage documents and offers new opportunities for exploring these resources. Optical character recognition (OCR) quality for under-resourced scripts, for historical manuscripts and as well as for documents printed in old typefaces, is still lacking. As an alternative or in addition, one can perform an image-based search, a form of "query by example". A simple and efficient pipeline for word spotting in historical documents is utilized here for three different applications; An effective unsupervised pipeline for OCR betterment is proposed. It employs a baseline OCR engine as a black box plus a dataset of unlabeled document images. Given a new document to be analyzed, the black-box recognition engine is first applied. Then, for each result, word spotting is carried out within the dataset. The unreliable OCR outputs of the retrieved word spotting results are considered, and the word that is the centroid of the set of OCR words, measured by edit distance, is deemed a candidate reading. We also present an image based approach for the retrieval of related articles in a newspaper. As a preliminary stage, given a ready to use corpus, synthetic images are generated for every word in it, and each of these words is considered a query. Given a set of unlabeled documents they are first fed into the word spotting engine. Then, based on the spotting results, a normalized Tf-Idf vector representation is computed for every document and the articles retrieval is performed by a nearest-neighbor search. Another utility shown here is an operational word spotting engine. We developed, in collaboration with the Friedberg Genizah Project, a real-time word spotting engine, incorporated in a large scale historical manuscripts collection – The Cairo Genizah.

Chapter 1

Introduction

Recent large-scale digitization and preservation efforts have made huge numbers of images of historical manuscripts readily available over the Internet. Unfortunately, Optical Character Recognition (OCR) for handwritten and historical documents is still insufficient, making it quite challenging to search within those images for something in particular. As an alternative, one can perform an image-based search. Given a query image of a word, one seeks all sub-images that contain occurrences of that same word within the dataset of documents.

We utilize here an underlying word-spotting engine, based on the work of [18] for several applications in the field of documents analysis. The word spotting pipeline is both simple to implement and fast to run, making it extremely easy to incorporate in different systems and with various manuscripts datasets. Words in the images are not pre-segmented and not labeled; thus, the method works completely unsupervised. Queries are usually derived from the dataset itself and can even be synthetic and generated from digital text rendered.

We propose to alleviate the OCR quality problem for historical manuscripts by using an efficient bootstrapping method for OCR betterment, based on the results of applying unreliable OCR to a dataset of unannotated documents. The outline of the suggested process is as follows:

- In the preparatory step, all images of a dataset A of documents undergo OCR by the baseline OCR engine. The corresponding bounding boxes from the set of OCR results, B, are resized to a fixed size and represented by conventional image descriptors. For accuracy and efficiency, a concise representation is extracted by performing maximum pooling over random groups of exemplars bounding boxes, using standard cosine similarities.

- Given a new document, requiring OCR betterment, word-spotting within the images A is performed for each resultant word $u \in B$, and the set $C \subset B$ containing the m best word-spotting results is considered.

- We use textual edit distances to compute the distance between all pairs of words in the set $D = u \cup C$ of $m + 1$, formed from the query u and the spotted words C. The candidate for

improved OCR, r, is the centroid of the set D. A score is assigned to the reading r based on the mean edit distance to the other elements in D and on the visual similarity to the bounding box u.

The entire process is fully automatic and efficient. It uses only two parameters beyond those of the OCR and word-spotting engines: the size n of the set of retrievals returned by the word-spotting, and the threshold θ used to decide whether to prefer r over u.

Second, we present a novel image-based approach for the retrieval of related articles in a digitized historical newspaper archive. Finding articles that discuss the same event can help scholars to easily explore a sequence of historical stories and can help users by enabling a fast navigation.

We start by binarizing all the dataset images and extracting candidate target words. Each target is resized to a fixed-size rectangle and as before represented by image descriptors, followed by max-pooling. We use a pre-computed Hebrew corpus as our dictionary and generate synthetic images for each of its words. We treat these synthetic images as queries and feed them to the word-spotting engine, performed within the dataset images. Employing these spotting results, we represent each article in the dataset using a term vector model combined with the traditional tf-idf weighting scheme. Finally, we retrieve related articles using cosine similarity for nearest neighbor search.

This process, as the preceding one, is automatic, simple and unsupervised. Only one parameter needs to be handled – the threshold used, τ , to decide whether a word-spotting result is considered positive or not.

Finally, we show an operational word spotting engine, which we developed as a joint work with the Friedberg Genizah Project. The engine is incorporated in the digitized Cairo Genizah manuscript collection available over the internet. Necessary adjustments have been made to fit the unique nature of this enormous and diverse dataset.

This research was done in collaboration with several different groups. For the OCR betterment we collaborated with the Indian Statistical Institute in Kolkata, who provided us with Bangla documents and corresponding OCR results. This work was published in the International Conference on Document Analysis and recognition (ICDAR) 2015. In the section of related articles retrieval we collaborated with the Historical Jewish Press Project (JPRESS), who gave us access to articles from historical newspapers. The research in this section was also a joint effort with Daniel Labenski, who studied this problem from a text mining point of view, and helped with the results labeling task. As stated before, the operational word-spotting engine was researched and developed in collaboration with the Friedberg Genizah Project (FPG).

The next section presents the two preservation projects of historical documents and the large-scale datasets we worked with. Chapter 3 briefly surveys some recent approaches to word spotting

algorithm, OCR betterment and topic detection. In Chapter 4 we describe in detail the underlying word-spotting engine, used in all of our proposed methods. Then, in Chapters 5-7 we explain the details of our different methods step by step, and present experimental results on the various datasets. We conclude with a brief discussion of possible further improvements and extensions.

Chapter 2

Background

This work was done in collaboration with two databases of historical datasets, that were recently digitized for research use and for the use of the wide public. The datasets are very large and would require years of human effort to process and to exploit for purposes such as topic categorization, retrieving related articles, searching for a particular phrase etc. This is where computational tools come in handy; with them, one can scan the whole dataset and give scholars a list of findings ordered by their relevance.

2.1 The Historical Jewish Press Project

The Historical Jewish Press website (JPRESS) [26] is a unique source of information on the history, life and culture of world Jewry and on the countries of Jews' residence in the modern era. The project brings the digitization revolution to this field and offers the possibility to perform a full search of all the published text of a given newspaper throughout all the years of its publication.[1] The Jewish Press project includes 97 different newspapers in 10 various languages from a wide variety of Jewish communities from the eighteenth century onwards.

This project is a joint venture of the National Library of Israel and Tel Aviv University. Today the users of this project range from independent scholars to journalists, lawyers, students and many more.

The digitalization process is comprised of three main technologies: scanning the material, converting the scanned images to text using OCR, and article segmentation. Although these three technologies are mature and time-proven, they are still not perfect. Neither OCR nor segmentation has reached a 100% standard of accuracy, and both are highly affected by the poor quality of the material.

[1] JPRESS website: http://web.nli.org.il/sites/JPress/English/about/Pages/default.aspx

The Historical Jewish Press website works with old newspapers and therefore is compelled to deal with many phenomena that threaten to decrease accuracy. These phenomena include inferior quality of printing (which characterizes early publications), yellowing paper, marks or damage in the original printing, unique fonts, etc.[2] The Technological limitations and the raw materials' poor quality lead to two primary problems that can occur: word identification errors and segmentation errors.

For the purpose of topic detection, we can avoid those problems by taking an image-based approach. We will show that a word spotting based system for topic detection can achieve reasonably high accuracy comparing to traditional topic detection in the presence of noisy OCR.

2.2 The Friedberg Genizah Project

The Cairo Genizah, discovered at the end of the 19th century, is a collection of over 200,000 fragmentary Jewish medieval texts that were stored in the loft of the ancient Ben Ezra Synagogue in Cairo, Egypt between the 8th and 17th centuries. These manuscripts outline a 1,000-year continuum of Middle-Eastern history and comprise the largest and most diverse collection of medieval manuscripts in the world.[3] When discovered, the fragmented manuscripts were quickly acquired by European and North American universities and by private collectors. For this reason, even today there is no exact accounting of the Genizah's contents or their whereabouts.

The documents discovered in the Cairo Genizah include a combination of important scholarly works, community records and ledgers, business contracts and more. A big part of these manuscripts are classic Jewish texts such as the Hebrew Bible. The works were written in several languages. Among them are Arabic, Hebrew and Armanic.

The Friedberg Genizah Project (FGP) [25] was established in order to advance and breathe new life into Genizah research. It is achieving its goals by locating, identifying, cataloging, transcribing, translating, digitally photographing and publishing images of all the manuscripts online. In May 2008, FGP released a fully-operational version of its on-line research platform, where it is now possible to view over 100,000 digitized images of Genizah manuscripts.

Research on many of the topics found in the Genizah is being organized by the FGP. There has been an ongoing collaboration between the FGP and Tel Aviv University and this work has contributed to the establishment of an operational module of Word-Spotting for the Cairo Genizah.

[2] JPRESS website: http://web.nli.org.il/sites/JPress/English/about/Pages/ about_technology.aspx
[3] FGP website: http://www.genizah.org/About-ExecutiveSummary.aspx

Chapter 3

Related Work

3.1 Word Spotting

Much effort has been devoted to research on word spotting, few examples are [12], [28], [9], [24]. Other works, with their own set of problems and less relevant here, deal with words embedded in outdoor photographs, e.g.,[39]. Dynamic time warping (DTW) and hidden Markov models (HMMs) are two popular training techniques. An example of the former is [27] and of the latter is [10]. A more recent approach is using a neural networks based system, for example the work in [37] and [11]. Many recent systems are supervised and pre-segmented. Regrettably, these techniques are time consuming.

Two approaches to searching for occurrences of words in documents are possible: one can first segment the text images into words and then compare each target word with the query, or one can search for a match to the query using a sliding window of some sort. There is substantial literature on word segmentation, including, for example, [20]. An example of word spotting using segmented images is [5]; among the works that do not require segmentation are [29], [4], [30] and a more recent one is [31]. An in-between approach is to work with multiple overlapping target regions, as in [22]. Using multiple candidates for the purpose of reducing the number of false positives that sliding-window approaches can engender, is a current trend in computer vision; see [32], [38] among others.

A second dimension that distinguishes work in this area is whether training examples are used for learning or whether the method is unsupervised. Our method is unsupervised.

The word-spotting engine presented in this work is based on the work of [18], which is inspired by the work of Liao et al. [19] in the domain of face recognition. Our training data is unsupervised, no classifiers are used to learn similarities and we employ a mechanism for identifying candidate targets.

3.2 Improving OCR using unsupervised word-spotting

Character recognition of printed text in Roman-based scripts is considered a solved problem since - for fair quality documents - OCR accuracy reaches 99.5% at word level. However, accuracy falls substantially when the document is of inferior quality, when it is old, or when it is printed in obsolete fonts. OCR engines for some oriental scripts are also quite advanced and have similar performance. However, the situation is not satisfactory for Indic scripts, for which the development of OCR engines is still at the laboratory stage.

There are several reasons for this situation. First, Indic scripts, such as Bangla, are alpha-syllabic, compared to Roman based scripts, which are alphabetic, with many fewer characters. Indic characters are divided into three categories, viz. basic, modified and compound; the number of distinct shapes that need to be recognized is about 1000. Character shapes have undergone changes over the past 200 years of printing, and orthography has also undergone modifications over this period, making a dictionary-based correction approach less effective. Furthermore, there is a resource crunch (of database and scientific information) for doing research in Indian languages and scripts that could otherwise be helpful for advanced OCR research. All the same, some good work has been done recently, and a workable OCR system for printed Bangla script has been developed lately. However, this system is not flexible enough to handle poor-quality Bangla text, and new approaches are required.

To the best of our knowledge, our work is the first to use word spotting in order to improve OCR results in an unsupervised manner. The work of Sankar et al. [34] is the closest work we are aware of. However, it deals with partial OCR, which is accurate where available, while we deal with noisy OCR. As specified before this word-spotting engine is based on the work of [18]. Whereas [18] is an unsegmented word spotting work, our method requires a tight coupling between the OCR and word-spotting results. Therefore, multiple adjustments are required. These include query jittering ([4], [19]) and a post-processing re-ranking stage.

3.3 Operational word-spotting module

As stated before many systems were suggested to solve the word-spotting problem, but no operational module for this purpose has ever been established. The word spotting engine in this module is again based on the work of [18]. Several changes and adjustments have been made to the original engine due to the special nature of the manuscripts, the handwriting style and the poor quality of the documents. These include for example introducing PCA for dimensionality reduction in replacement of one of the components in the engines pipeline.

To the best of our knowledge, our work, is the first to build an operational word–spotting module which is implemented in a real-life situation. There have been some experimental modules but none of which were implemented. The work of Wieprecht et al. [40] is the most relevant to

our work that we are aware of. They proposed a method for querying digital archives of historical documents with online-handwritten queries. They also presented a short interactive demo but it only included a small amount of pages and was not implemented on a large scale database.

3.4 Topic Detection and Tracking (TDT)

Image based topic detection can be in some way compared to topic detection in the field of NLP and data mining. This NLP task is related to the research of Topic Detection and Tracking. The TDT study [2]was an initiative to investigate new technologies for finding events and topics in a stream of news stories. The study defined the term event as some unique thing that happens at some point in time. For example, the eruption of Mount Pinatubo on June 15, 1991 is consider to be an event, whereas volcanic eruption in general is considered to be a class of events. We will define a class of events as topic, for example sports events or political events.

The input to this process is a stream of stories. The TDT study defined four technical tasks:

1. The Segmentation task - segmenting a continuous stream of text into its constituent stories.
2. The Retrospective Event Detection grouping the stories in the corpus into clusters, where each cluster represents an event and where the stories in the cluster discuss the event.
3. The Tracking Task - a target event is given, and each successive story must be classified as to whether or not it discusses the target event
4. The On-line New Event Detection Task - deciding if a story in a stream of news stories is a first story of a new event or if it discusses an event already seen.

The most relevant task to our work is the retrospective detection task. The detection is an unsupervised learning task (without labeled training examples). Whereas the study tries to group the stories into clusters were each cluster should correspond to a particular predetermined target event, our goal is to retrieve for each predetermined story the stories that discuss the same event as it.

For solving the task, they used the conventional vector space model and each story was represented as a vector. They tested several typical term weighting schemes which combine the Term Frequency (TF) and the Inverse Document Frequency (IDF) in different ways [33]. In this paper we use this simple but powerful method for document representation and similarity.

Alan and Harding et al. [3] used one version of tf-idf and selected the top-weighted 1,000 terms from the document to create a vector representation. Other document representation and similarity measures were suggested, especially worth mentioning [35], [36], [42] used language model-based similarities. New event Detection has recently regained its popularity with the emerge of social media. Some recent works are [23], [21].

Whereas the task in the field of NLP is widely researched, we suggest a novel approach using images as our sole input. To the extent of our knowledge, this work is the first to take an image-based approach for solving the topic detection problem. The work of Wilkinson et al. [41] is in some way related to this work. They used an image-based approach, including the word-spotting engine, for solving a basic data mining problem. However, it deals with computing word clouds while we deal with Topic detection and categorization.

Chapter 4

Word Spotting

Given a query image of a word, the word-spotting engine seeks all sub-images that contain occurrences of that same word within the dataset of documents. In this chapter we present the word-spotting engine that will appear in various forms in all of the following chapters. This engine was presented in [18]. One of the strengths of this proposed method lies in its simplicity. The set of candidate targets is extracted by a straightforward process. Then, the same processing pipeline is applied to both the candidate targets and the query image. The process of max-pooling, relies on the unsupervised extraction of target candidates from a set of training images. In practice, the training set is the set of all dataset images. Identifying matching targets is performed via a nearest neighbor search.

4.1 Pre-processing dataset images

First, the dataset images are binarized by thresholding the input image at a value which equals 85% of the mean pixel intensity. Then, connected components are computed, and connected components that are either too small (noise) or too big (stains and page margins) are discarded.

4.2 Extracting candidate targets

The candidate targets are formed as groups (of arbitrary size) of the computed connected components that satisfy five criteria:

1. All the components forming the group should fall within width and height limits set by predetermined thresholds.

2. The projection of the centers of mass of each component to the y-axis does not have a gap larger than a threshold.

3. The projection of the pixels of all the components onto the x-axis does not have a gap larger than another threshold. This is used to exclude boxes containing more than one word.

4. There is no component within the group that its centroid lies inside the boundaries of the group but it has pixels outside the boundaries.

5. There is no component outside the group that lies between the top and bottom group's boundaries and to the right/left of the group with a gap smaller than a threshold.
   Since words tend to have spaces between them, this criterion is used to exclude boxes containing only a part of a word and therefore do not have spaces before/after them.

Given an image, the suitable groups of connected components are collected by iterating over all connected components. Each time, the connected component at hand is considered to be the top-right component of the group, and by scanning from right to left subsets are constructed and evaluated. The groups of the connected components that satisfy the first and second criteria are first considered and then smaller subsets are evaluated based on the next three criteria. The resulting candidate targets are typically dense and often overlapping.

4.3 Representing binary image patches

The process of representing an image patch is applied to the binarized version of the query and to each of the binary candidate targets. Each binary patch is cropped to the minimal bounding box containing all the pixels and embedded in a larger white patch, adding a fixed size margin to it. The extended patch is resized (by image interpolation) to a patch of a fixed size. Using a regular grid, the fixed sized patch is divided into multiple non overlapping cells, each of which is encoded by a HOG descriptor [8] of length 32 and by an LBP descriptor [1] of length 58. The HOG descriptors of all cells are concatenated and the resulting vector is normalized to a Euclidean norm of 1. The same process is applied to all LBP descriptors. The two vectors are then concatenated to form a single vector $v \in \mathbb{R}^d$.

A matrix $M \in \mathbb{R}^{n \times d}$, which consists of the vector representations (same as v) for n random candidate targets from the dataset is then considered. The vector v is transformed into a vector $s \in \mathbb{R}^n$ by means of a linear projection: $s = Mv$. In other words, the normalized descriptor vector is represented by its cosine similarities to a predetermined set of exemplars.

Then, a max-pooling step is carried out. The set of indices $[1 \dots n]$ is randomly split into groups I_i of size p. Given a vector s, this max-pooling is performed simply by considering one measurement per I_i that is the maximal value among the values in the vector s in the indices of I_i. Put differently, let x be the vector of length n/p that results from the max-pooling process as applied to the vector s. Then $x_i = \max_{j \in I_i} s_j$.

4.4 Query

Given a query image, it is treated as explained above. It is binarized, padded with margin and resized. Then the vector consisting of the multiple HOG and LBP descriptors is computed, multiplied by the matrix M, and max-pooling is employed using the same partition $\{I_i\}$. This vector is then compared, by means of $L2$ distance, to the similar vectors – computed in exactly the same manner – of all candidate targets. Note that the set of vectors associated with the candidate targets is precomputed, which supports scalability. Since the representation is compact and the nearest neighbor search can be performed by means of matrix multiplication, the entire search process is performed very efficiently in main memory.

All candidate targets are ranked in accordance with the computed distance. Recall that there are many overlapping candidate targets. In order to eliminate multiple occurrences of the same target word as it appears in multiple candidate targets, a heuristic step is employed. Out of all bounding boxes that contain the same connected component as their largest component, only the box with the highest rank (lowest distance) is considered. The rest of the bounding boxes that share the same maximal-area component are eliminated from the retrieved list.

There are various parameters used in the word-spotting system. In the following chapters the selected parameters will be given in detail for each system separately

Chapter 5

Improving OCR using Word-Spotting

In this chapter we will go over our proposed method's steps for improving OCR using unsupervised word-spotting engine. The proposed method improves the OCR of a new document by considering each OCR result u as a query and comparing it to a set of words B extracted automatically using the OCR engine from the dataset of all documents images A. The method handles both the documents of dataset A and the document that include the query u in a uniform fashion, and the same processing pipeline is applied to both.

5.1 Method description

As a first stage, OCR is applied to each document image in A. The OCR results B include both the bounding box and proposed text of each recognized word. Word segmentation is therefore an integral part of the OCR engine. We do not make use of the quality score returned by the OCR engine, and we expect the OCR results to be noisy and only partly reliable.

Next, we use the word spotting engine. Since all the document images in set A are already black and white, we can skip the pre-processing step. The bounding boxes are given to us by the OCR engine, and for each OCR result the binary image patch of the associated bounding box is considered. Therefore, the step of extracting candidate targets is skipped as well.

The patch is resized to a fixed size: 160×64 pixels and divided into a grid of 20×8 cells each 8×8 pixels, see Figure 5.1. The length of the vector v, denoted above, is therefore $20 \times 8 \times (31 + 58) = 14,240$. Since we rely on pre-segmented bounding boxes, which are not always exact, we apply a jittering process. Each bounding box is considered five times: the original bounding box, plus the bounding boxes that are obtained by shifting the original one 4 pixels in each of the four directions.

The matrix M is consisted of the vector representations for random OCR bounding boxes from the dataset B. n, the number of exemplars used, is set to 1000. We use a random partition $\{I_i\}$ that contains 250 subsets of the indices $[1 \dots 1000]$ of size 4, resulting in patch representation vectors $x \in \mathbb{R}^{250}$.

(a) (b)

Figure 5.1: The patch normalization process. (a) Image patches obtained using the bounding box of the OCR engine. (b) Resized patches with grid overlayed. All bounding boxes of the set of documents and the target document are resized to the same size regardless of their size and aspect ratio.

Given a new document requiring OCR, the black box OCR engine is applied to it. Then, each OCR result u is considered separately and represented as vector as explained above. This vector is then compared, by means of $L2$ distance, to the similar vectors - computed in exactly the same manner - of the OCR results of B set extracted from the documents dataset A. Note that the set of vectors associated with B is precomputed.

All elements of the set B are ranked in accordance with the computed $L2$ distance in $\mathbb{R}^{250}$. Unlike [18], we found the underlying encoding to be more reliable than the pooled similarity representation, and a re-ranking procedure is thus employed. The top $n_0 = 50$ query results are considered. For each, we compare the combined HOG+LBP encoding in $\mathbb{R}^{14240}$ by means of cosine distance to that of the word u.

Then, the set C containing the top $m = 9$ results is considered. The results in set C are fused to create a new OCR candidate r. We consider the set D of $m + 1$ words that is the union of the word u with the words of C. Textual edit distances with a fixed and equal insert/delete cost is computed between $\binom{m+1}{2}$ pairs of words in D. The candidate for improved OCR, r, is the centroid of the set D, i.e., the word with the least mean distances to the rest of the words in D:

$$r = \arg\min_{w \in D} S_a(w)$$

where

$$S_a(w) = \sum_{x \in D} d_{edit}(w, x)$$

The new candidate r is assigned a quality measure (lower is better) that is based on two factors.

The first $S_a(r)$, is the mean edit distance to the other elements in D. The second factor $S_b(u;r)$ is the visual similarity between the bounding box of u and the bounding box associated with r, as measured by the cosine distance of the joint HOG+LBP representation. The final combined quality score used is given by

$$S(r|u) = \log\left(e^{-S_a(r)/2} + S_b(u;r)\right)$$

The use of the exponent is done, as is often done, in order to convert a distance to a similarity. Finally, the text of r is used in lieu of u for the same bounding box of u if $S(r|u) > \theta$. The default value of θ in our experiments is 0.25.

5.2 Bangla OCR

An OCR system for Bangla has recently been developed at the Indian Statistical Institute, which works with about 98% accuracy for clean and recent documents containing text printed in the various fonts in modern character styles [6, 7]. The system begins with preprocessing, including noise removal and skew correction. This is followed by binarization, then text line, word and characters/sub-character segmentation in the upper, middle and lower zones, after which the characters/sub-characters are submitted to a two-stage tree classifier. The first stage is a group classifier, wherein each group may consist of one or more similarly-shaped character classes. The groups are then subjected to second-stage classifiers to recognize the character/sub-characters of each group. This approach improves speed and offers flexibility in choosing different sets of features at the second level. Then the recognizer outputs of the upper, middle and lower zones are combined to form characters, and the characters are combined into words in machine code, with some simple post-processingbased on orthographic positioning rulesemployed to correct a small amount of output errors. No dictionary or deeper linguistic information is utilized to improve results.

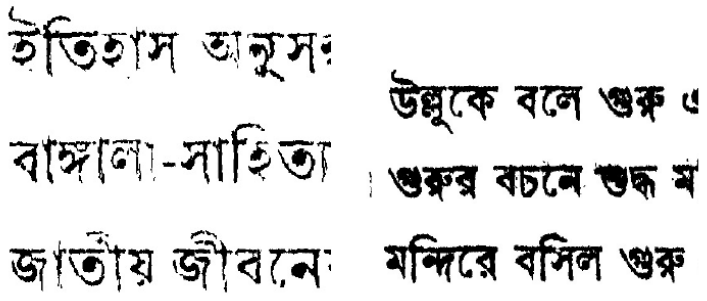


Figure 5.2: Samples of printed Bangla text. The printing is crude and the font is obsolete leading to poor OCR results.

5.3 Evalution

Our method is evaluated on the Bangla dataset we downloaded from the Digital Library of India. The dataset contains printed text that was printed nearly 100 years ago using crude technology and an obsolete font. See Figure 5.1 for sample text. We used 18 pages comprising 3576 words that were manually annotated for evaluation purposes only. For each query, we apply the jittering process, and all the elements in the set of the OCR results are ranked based on visual similarity. We then consider a set of 10 words, the top 9 results from the retrieved words and the query word. Out of this set a candidate is chosen to improve the OCR, see Figure 5.3 for examples. A quality score is then calculated for this candidate (see method description) and the text in the position of the querys bounding box is replaced with the candidates OCR if the score is higher than a threshold. To evaluate the OCR betterment process, we report OCR accuracy before and after implementing our system. We also evaluate independently the performance of the word spotting system. For this, the mean Average Precision (mAP) retrieval score is used, according to reporting standards in the literature.

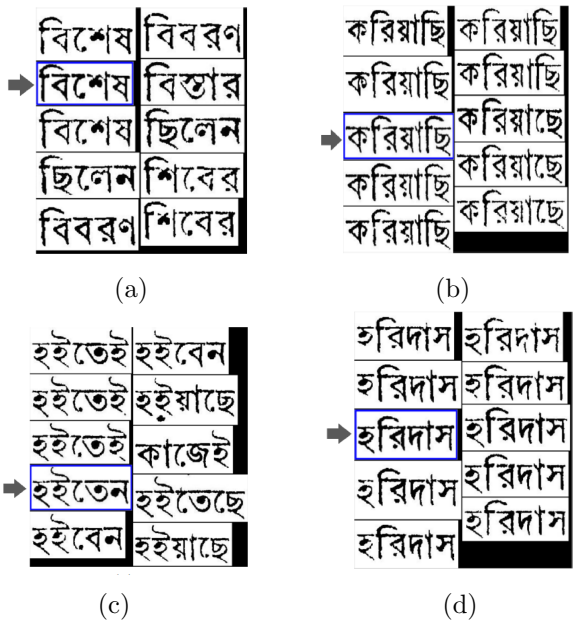


Figure 5.3: Samples of OCR replacements, where the original OCR result was replaced by the new OCR candidate. In all cases, the original word is at the top left. The OCR of the marked words was the one selected for the replacement. (a,b) are good results. In (c) the word that was chosen is not the same as the query word. In (d) the word that was chosen is the same as the query word, however its OCR is incorrect. The figure could be misleading: there are many more occurrences of good replacements than of harmful replacements.

Table 5.1 shows the results achieved by our OCR improvement system and variants of it. We present results for our complete pipeline using the edit distance two alternative systems: one using a Longest Common Subsequence (LCS) based text similarity, and the second using a bag-of-letters representation. The length of the LCS is normalized by the length of the two words, which improves performance considerably. The bag-of-letters method simply represents each word by a histogram of letter frequencies, compared, as it gives best performance, by the cosine similarity. The LCS variant seems to perform slightly better than both alternatives and gives a sizable improvement of 12.2% in the OCR accuracy (over 23% relative improvement).

The performance of the word spotting method by itself, applied only on the Bangla pages which have ground truth is reported in Table 5.2. Presented are results for the complete word spotting pipeline, and for the pipeline without the suggested modifications of query jittering and re-ranking. As can be seen, both help improve the overall word spotting quality.

Method	OCR accuracy
without implementing our method	52.0%
Complete pipeline using bag of letters	61.8%
Complete pipeline using LCS	64.2%
Complete pipeline using the edit distance	64.0%

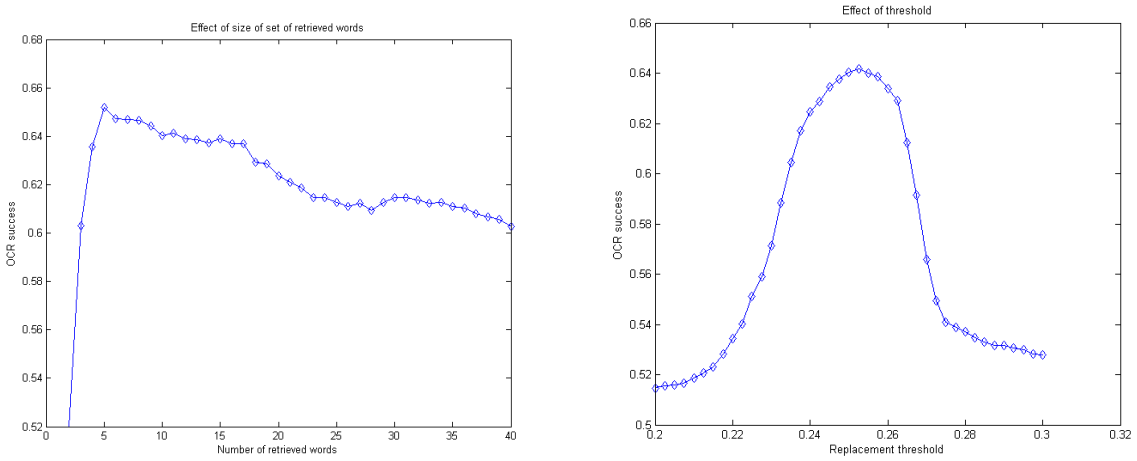
Table 5.1: OCR accuracy for the baseline OCR method, the suggested pipeline and the same pipeline where the edit distance, used to compare OCR results, is replaced by the Longest Common Subsequence similarity or the bad of letters method.

Method	mAP
Complete pipeline	93.6%
Complete pipeline w/o query jittering	87.3%
Complete pipeline w/o re-ranking	89.7%
Baseline [18]	79.8%

Table 5.2: Word Spotting accuracy evaluated independently of OCR improvement. We present results for the complete pipeline and the pipeline without the two improvements introduced in this paper

We studied the effect of two parameters on the systems performance: the size of the set of retrieved words and the replacement threshold (see Figure 5.4). The system is robust with respect to both. A value of m between 4 and 20 would give a good result, and a value of $m = 5$ would give an overall OCR result of 65.2%. For the threshold, which is in log scale, once is below 0.20 all candidate replacements are made, above 0.30 none are made. Within the range $[0.21 \dots 0.28]$ the dependency between the threshold and the accuracy follows a smooth bell curve with a relatively large plateau between 0.24 and 0.26. We also tried to apply the OCR improvement procedure iteratively, each time using the improved results from the previous round. The improvements were

miniature: The second round contributed 6 more correct OCR results, the third round contributed one more correct word, and the process converged.



(a) (b)

Figure 5.4: The sensitivity of the proposed system to its parameters. (a) OCR accuracy vs. the size of the retrieved set D. (b) OCR accuracy vs. the score threshold used to decide whether to switch the OCR result to the new candidate.

Chapter 6

Finding Related Articles

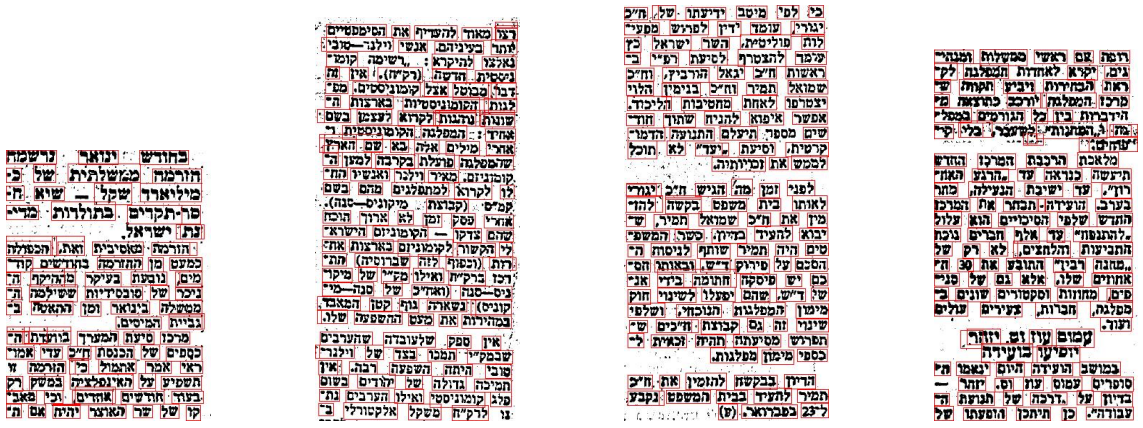
In this chapter, we will present our image based solution to the event detection problem. Given an article in a historical newspaper, we try to retrieve the articles related to the same event. The proposed method represents each article as a vector and computes the distance between each query article and all the articles in the dataset. Whereas when assuming having access to the text it is quite straightforward to represent the articles as vectors, we assume having only images as input to our algorithm. Since we cannot use the text to build a corpus, we chose a ready-to-use Hebrew corpus and generate a synthetic image for each word in it. We use the word spotting engine with these synthetic images as queries and get surprisingly good results.

6.1 Method description

6.1.1 Word spotting with synthetic queries

First, given the documents, we implement the word spotting engine. We followed the stages described in the word spotting chapter. For the pre-processing stage, we chose for each connected component minimal and maximal sizes of 35 and 2000 pixels respectively. The allowed height range is $[4, 120]$ pixels and the allowed width range is $[3, 280]$ pixels. For candidate extraction, we followed the five criteria. The bounding boxes' minimal height and width are set to 20 and 13 pixels respectively. The maximal horizontal gap is set to 32 pixels and the maximal vertical gap is set to 12 pixels (criteria 2,3). The sizes of the candidate targets are bounded (criterion 1) by a rectangle of 240×55 pixels. See Figure 6.1 for candidate targets examples.

For patch representation, the patch is resized to a fixed size of 160×56 pixels and divided into a grid of 20×7 cells each 8×8 pixels. The length of the vector v, denoted d, is therefore $20 \times 7 \times (31 + 58) = 12,460$. We apply also the jittering process as described in previous chapters. The matrix M is consisted of the vector representations for random candidates. n, the number of exemplars used, is set to 3750. We use a random partition $\{I_i\}$ that contains 250 subsets of the indices $[1 \dots 3750]$ of size 15, resulting in patch representation vector x in $\mathbb{R}^{250}$.



(a) (b) (c) (d)

Figure 6.1: (a,b,c,d) The binarized pages from the *Davar* newspaper with the candidate targets overlayed.

In order to represent an article as a vector we needed to define a dictionary. We used a Hebrew corpus as our dictionary and for every word in it we generated a synthetic image. We consider all the words in the corpus as queries. Each query image q is considered separately and treated and represented as explained above, resulting a vector in $\mathbb{R}^{250}$. This vector is then compared, by means of $L2$ distance, to the similar vectors of all candidate targets extracted from the documents' dataset. All candidates are ranked in accordance with the computed $L2$ distance.

Then, for each query, we apply a threshold and consider only the set of candidates for which $S(p;q) > \tau$. Where $S(p;q)$ is the visual similarity between the bounding box of candidate p and the bounding box associated with the query q. The default value of τ in our experiments is 0.99.

(a) (b)

Figure 6.2: Top 100 retrievals for the queries *hakneset* (a) and *Israel* (b).

To eliminate overlapping bounding boxes we implement the same method as explained in the previous chapters. For better accuracy, we also disallow the same candidate to appear in more than one set of results. We eliminate multiple occurrences of the same bounding box across different sets of results associated with different queries. Out of all the occurrences of the same bounding box only the instance with the highest rank (lowest distance) is considered. The rest of the instances are eliminated from the retrieved lists.

6.1.2 Article representation

Bag of words

Applying a threshold enables us to treat this problem as a NLP task. We can now create a histogram representing the number of times a word appears in the article and compute a Bag of words vector. This is the simplest baseline. Every document is represented by an unordered collection of the distinct words in it, and so each document is represented by a vector of size m . Where m is the number of words in the dictionary. In this scheme, all terms have the same weight disregarding how many times they appear in the corpus and are considered equally important when it comes to assessing relevancy on a query. Therefore, certain terms have little or no discriminating power in determining relevance.

Basic tf-idf

To overcome the problem mentioned above we use the term frequency-inverse document frequency (tf-idf) method [40]. This well-known representation is often used in Information Retrieval, and while it is simple, it gives good and competitive results. tf-idf representation is intended to reflect how important a word is to a document in a corpus. The tf-idf value increases proportionally to the number of times a word appears in the document, but decreases proportionally to the frequency of the word in the corpus. For the term frequency $tf(t, d)$, the simplest choice is to use the raw frequency of a term in a document.

$$tf(t, d) = f(t, d)$$

Where $f(t, d)$ is the number of times the term t occurs in document d.

The inverse document frequency is the logarithmic scaled inverse fraction of the document that contain the term.

$$idf(t, D) = log\left(\frac{N}{df_t + 1}\right)$$

Where D is the corpus and N is the number of documents in the corpus $N = |D|$. df_t is the document frequency, defined to be the number of documents in D that contain the term t.

Finally, we get:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, d)$$

tf-idf variants

In our experiments, we also tested variants of the tf weight:

- Logarithmic scaled frequency:

$$tf(t,d) = \begin{cases} log(f(t,d)) + 1, & f(t,d) \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

- Root of frequency:

$$tf(t,d) = \begin{cases} \sqrt{(f(t,d))}, & f(t,d) \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

Querying an article

Given an article to query it is treated as explained above and is represented as vector using the tf-idf method resulting a vector in $\mathbb{R}^m$, where m as before is the size of the dictionary. Nearest neighbor search is then performed. There are several similarity measures or distance measures we can use to find the nearest neighbors. among them are cosine similarity, Jaccard similarity, KL divergence and Euclidean distance. We chose to work with the cosine similarity. This metric is often used in the literature to measure documents similarity in text analysis [13], [15] and is independent of document length, which is an important property. Moreover, it is simple and fast to calculate since the nearest neighbor search can be performed by means of matrix multiplication.

Given two documents d_1, d_2 their cosine similarity is

$$sim(d_1, d_2) = \frac{d_1 \cdot d_2}{||d_1||||d_2||}$$

Where d_1, d_2 are N-dimentional vectors over the term set $T = \{t_1 \dots t_m\}$. Each dimension represents a term with its weight in the document

Finally, the results are ranked in a descending order in accordance with the computed similarity and k-top results are considered.

6.2 Evaluation

This method is evaluated on a dataset given to us by the JPRESS project. The dataset is comprised of all the issues of *Davar* newspaper from February 1981 (see Figure 6.3 for sample text). We were given 10881 images which were pre-segmented into 3233 articles. The candidate extraction stage evaluated on those articles resulted in 2,289,263 word candidates.

We worked with a corpus downloaded from *MILA* [17], the knowledge center for processing Hebrew. We chose the *HaKnesset* Corpus, which contains all the sessions protocols of the Knesset from January 2004 through November 2005, and used the list of all tokens appearing 10+ times by frequency. For better accuracy, we pre-processed the corpus so it will contain only Hebrew words.

Out of this cleaned and sorted list we utilized the first 100,000 words as our dictionary. When generating the synthetic images, we used 'Times New Roman' font since it is the printing font in most of the dataset's documents. We tried different font sizes, and empirically found that large font sizes produce better spotting results, and therefore chose to use a font size of 100.

Figure 6.3: The front page of *Davar* newpaper 01.02.1981

We chose 111 query articles from the dataset, some reporting a major continuous event while others reporting an anecdotal event that has only a few or no related articles. For each query article, we considered the top 8 nearest neighbors and labeled the returning results as one of the following categories: same event (match), same topic, or not related. The average number of related events per query is 9.4 and the median is 7. The average number of articles with the same topic as the query is 25 and the median is 12.

The two most important evaluation measures in information retrieval systems are precision and recall. Precision is always reported in formal information retrieval experiments, while recall has always been a more difficult measure to calculate, because it requires the knowledge of the total number of relevant items in the collection. It becomes increasingly difficult as collection size grows. For many prominent applications, particularly web search, measuring precision at all recall levels may not be relevant to the user. What does matter is the number of good results in the first page. This leads to measuring precision at fixed low levels of retrieved results, such as 10. This is referred to as "precision at k". It has the advantage of not requiring any estimate of the total number of relevant documents.

Since labeling the entire dataset is a time-consuming task, and since our goal is mainly to compare different approaches, we chose to report precision at k. To evaluate the results, we report the precision of our retrieval system alongside with the results of the work of Daniel Labenski. Daniel's work is in the field of data mining. It solves the event detection problem based on pre-computed noisy OCR using the same dataset and queries and using the root normalized tf-idf for vector representation. The labeling process was a joint work of us. Since no Ground truth is available for this dataset, we combined all the labels from all the experiments and estimated a gold standard for a better understanding of the results. We used this estimated gold standard to calculate an approximate recall at 8.

Table 6.1 shows the results achieved by our detection system and variants of it. We present results for our pipeline using the tf-idf and its variants. As it can be seen the normalization gives a sizable improvement compared to the baseline tf-idf. Log normalization variant seems to perform slightly better than the root normalization alternative and gives a reasonable accuracy comparing to the text based approach, retrieving 87.5% of the possible related articles in the gold standard at rank 1. We also present the results for the topic detection task in Table 6.2. As it can be seen in topic retrieval the gap between our method and the text-based method is much reduced.

Method	Precision@1	Precision@3	Precision@5	Precision@8	recall@8
tf-idf	0.78	0.67	0.58	0.52	0.53
tf-idf root normalization	0.83	0.70	0.62	0.54	0.55
tf-idf log normalization	0.84	0.72	0.64	0.56	0.57
Text based	0.87	0.76	0.70	0.62	0.64
Gold standard	0.96	0.90	0.85	0.75	0.79

Table 6.1: Results achieved by our event detection system and variants of it. We present results for our pipeline using the tf-idf and its variants. As it can be seen the normalization gives a sizable improvement compared to the baseline tf-idf . Log normalization variant seems to perform slightly better than the root normalization alternative and gives a reasonable accuracy comparing to the text based approach, retrieving 87.5% of the possible related articles in the gold standard at rank 1.

Method	Precision@1	Precision@3	Precision@5	Precision@8	recall@8
tf-idf	0.95	0.90	0.86	0.84	0.31
tf-idf root normalization	0.97	0.92	0.90	0.87	0.33
tf-idf log normalization	0.98	0.94	0.91	0.89	0.33
Text based	0.99	0.95	0.93	0.89	0.34
Gold standard	1.00	0.98	0.94	0.90	0.38

Table 6.2: Results achieved by our Topic detection system. As it can be seen in topic retrieval the gap between our method and the text-based method is much reduced.


בצל איומי הפרישה של ח"כ דרוקמן

מחריף המאבק במפד"ל על הרכב מטה הבחירות

מאת א. פנרתי סופר „דבר"

שלושה הלכו – ארבעה נשארו

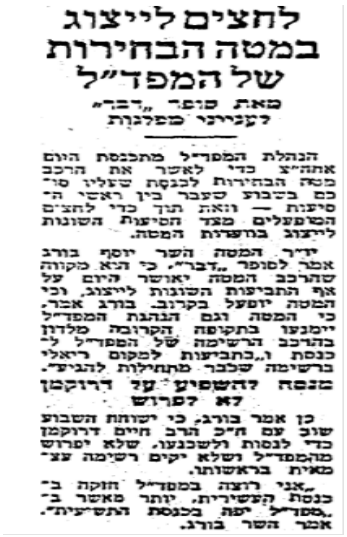
מאת כותבי המכתבי של „דבר" בירושלים

אין סיכוי למנוע הקדמת הבחירות

מאת: דניאל בלוך.

חוגים קואליציוניים:

(a)


לחצים לייצוג במטה הבחירות של המפד"ל

מאת סופר „דבר"
לענייני מפלגות

הנהלת המפד"ל מתכנסת היום אתה"צ כדי לאשר את הרכב מטה הבחירות לכנסת שלפני סוכם בשבוע שעבר בין ראשי הסיעות — ואת תנך כדי לחצים המתפעלים מצד חסידי השונות לייצוג במשרות המטה.

ידיר המטה השר יוסף בורג אמר לסופר „דבר", כי הוא מקווה שהרכב המטה יאושר היום על אף התביעות השונות לייצוג. וכי המטה יוסיף בקרוב. בורג אמר, כי המטה וגם הנהגת המפד"ל יימנעו מלקרוא הקרובה כלדרך בהרכב הרשימה של המפד"ל לכנסת ו„בהיעות למקום ריאלי ברשימה שכבר מתחילות להגיע".

מנסה להשפיע עד דרוקמן לא לפרוש

כן אמר בורג, כי ישוחח השבוע שוב עם ח"כ הרב חיים דרוקמן כדי לנסות ולשכנעו, שלא יפרוש מהמפד"ל ושלא יקים רשימה עצמאית בראשותו.

„אני רוצה במפד"ל חזקה בי כנסת קשירית, יותר מאשר בי המפד"ל יפות בכנסת התשיעית", אמר השר בורג.

(b)


נזלת או דלקת ריאות

החפלגות הרב דרוקמן מהמפד"ל והקסנת השפעתם של „העצירים" תאפשר קואליציה עם המערך

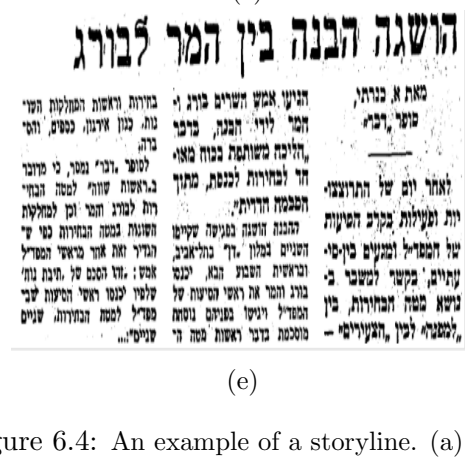
מאת חגי אשד

חיים דרוקמן — יוסף בורג — זבולון המר

(c)


בורג והמר נועדו למציאת פשרה במשבר במפד"ל

(d)


הושגה הבנה בין המר לבורג

מאת א. פנרתי,
סופר „דבר"

(e)


מתגבשת פשרה במפד"ל בנושא מטה הבחירות

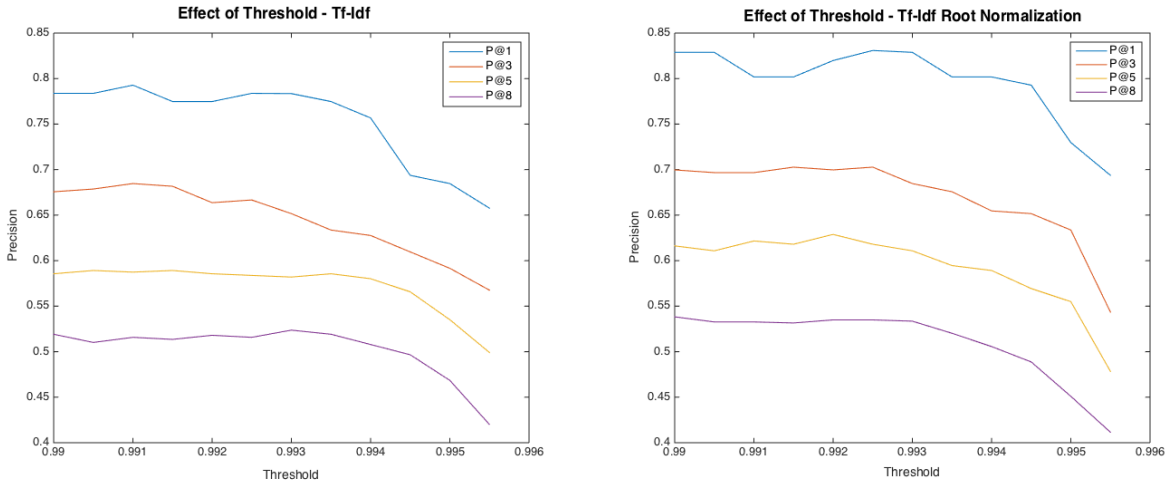
מאת סופר „דבר"
לענייני מפלגות

(f)

Figure 6.4: An example of a storyline. (a) The query article and the top 5 retrievals (b - f).

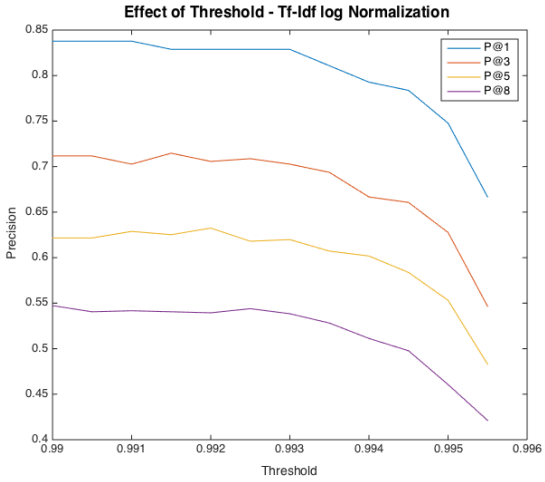
We studied the effect of the visual similarity threshold τ, which we integrated in the word spotting engine, on the event detection system (Figure 6.5) and on the topic detection system (Figure 6.6). The systems are partially robust to it. Within the range $[0.99 \ldots 0.9935]$ the dependency between the threshold and the precision is quite constant, and although there are many false positive results the precision is high. Once τ is above 0.9935 not enough words are considered as positive results so the number of false negative is large and the performance starts to decrease.

We also tried to improve the results by using a Hebrew morphology engine developed by Hspell [14] - a free Hebrew linguistic project. We used the projects Hebrew analyzer to expand each term to all recognized lemmas found in the dictionary, but there was no visible improvement.



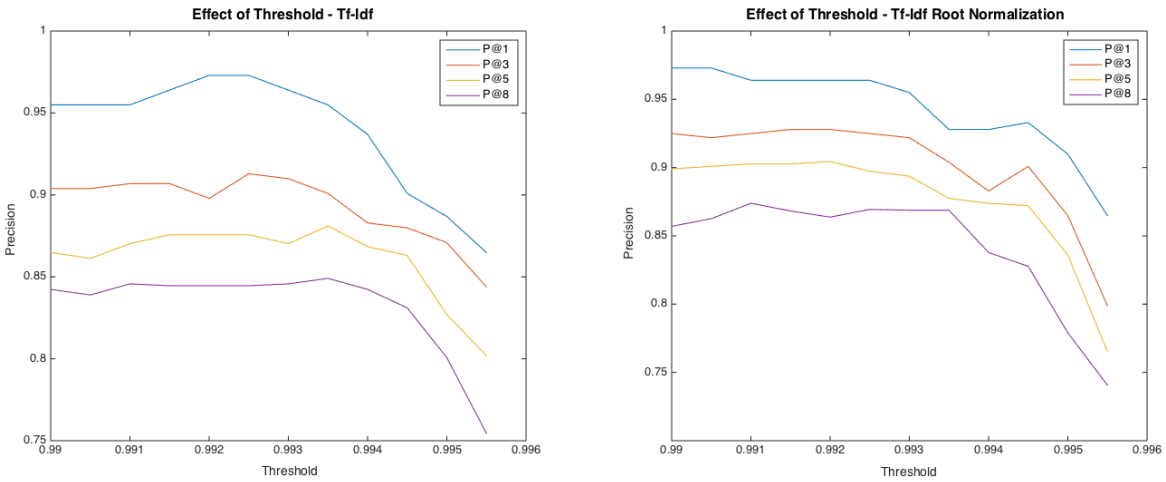
(a)

(b)



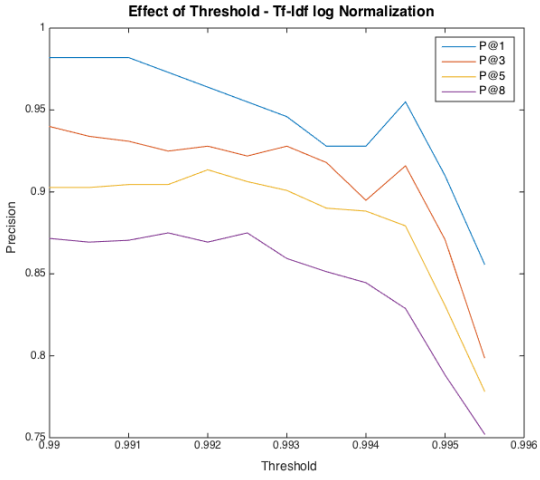
(c)

Figure 6.5: The sensitivity of the event detection system to the threshold τ. The graphs show the precision vs. the threshold τ, used to decide whether a spotting candidate is considered as a positive result . Graphs (a), (b) and (c) are for tf-idf variants: baseline, root normalized and log normalized respectively.



(a)

(b)



(c)

Figure 6.6: The sensitivity of the topic detection system to the threshold τ. As in Figure 6.5 the graphs show the precision vs. the threshold τ, ans as before graphs (a), (b) and (c) are for tf-idf variants: baseline, root normalized and log normalized respectively.

Chapter 7

Operational Word-Spotting module

The operational word spotting module was built in collaboration with the Friedberg Genizah Project, and is based again on the work of [18]. Our goal was the traditional goal to seek all sub-images that contain occurrences of a word within the dataset of documents. The image of the query word is selected from the documents dataset and manually marked by the user. We will present in this chapter the necessary adjustments which were made for the Cairo Genizah documents, and show some results from the live word spotting module.

The Genizah project contains more than 200,000 documents. The documents were written in multiple languages by multiple people, and are poorly preserved (see Figure 7.1 for sample text). We were provided with approximately 4600 documents for developing and testing the word spotting engine. For the following stages of the module we chose reasonable values for each parameter separately based on observing the documents.

7.1 Pre-processing dataset images

Connected components are computed and discarded if necessary according to the predetermined parameters. The minimal and maximal size for each connected component is 20 and 2000 pixels respectively. The allowed height range is [5,130] pixels and the allowed width range is [2,400] pixels.

7.2 Extracting candidate targets

Connected components are grouped to try and create a candidate. Each bounding box must meet the criteria described in the word spotting chapter. The sizes of the candidate targets are bounded (criterion 1) by a rectangle of 150×100 pixels. The bounding boxes' minimal height and width are 17 and 20 pixels respectively. The maximal horizontal gap and the maximal vertical gap are both set to 15 pixels (criteria 2,3).

Since in many of the Genizah documents there are no spaces between the words, criteria no. 5 could not be met. This condition, in some cases, is the result of the poor quality of the documents

and in others it is caused by the old handwriting style and the ancient writing tools. As mentioned before, the resulting candidate targets are typically dense and often overlapping. When omitting criterion no. 5, the number of word candidates and especially the number of false candidates is increased significantly and we get 2620 candidates per image on average. This fact caused two problems which needed to be addressed. The first problem is memory limitations. This problem is mostly affected by the extremely large size of the candidates' representation matrix, which needs to be kept in the memory. Even when working only on the documents we were given, the number of candidate targets is approx. $2620 \times 4600 = 12,052,000$. This problem is still an issue and the current solution is to run the engine only on a small portion of the project's documents. The second problem occur since there are more false candidates than real word candidates. The original algorithm does not perform very well under those circumstances. We will elaborate more on the issue and its solution in the next stage.

7.3 Representing binary image patches

The margin used in order to embed the patch's bounding box in a larger patch is 8 pixels in all four directions. This larger patch is resized to a fixed size: 160×64 pixels and divided into a grid of 20×8 cells each 8×8 pixels. The length of the vector v, denoted d above, is therefore $20 \times 8 \times (31 + 58) = 14,240$. Since each query word is selected from the dataset and manually marked by the user, the marking is slightly inaccurate and we apply the jittering process from the previous chapter in this module as well.

The approach of representing each normalized descriptor vector by its similarities to a predetermined set of exemplars does not perform well in this configuration and gives poor results. We believe it is due to the inaccurate previous stage of segmenting the words, and the fact that most of the extracted bounding boxes are false candidates. In this case when randomly choosing target candidates, the matrix M will consist mostly vector representations for false candidates, which are bounding boxes containing parts of words, more than one word or even noise. We decided to skip this step followed by max-pooling and to use instead the Principal Component Analysis (PCA) procedure for dimensionality reduction [16]. We use the same conventional image descriptors and randomly choose 50,000 vector representations (same as v) to evaluate the PCA transformation matrix $W \in \mathbb{R}^{d \times d}$. The transformation maps the data vector v from an original space of d variables to a new space of d variables which are uncorrelated over the dataset. However, not all the principal components need to be kept, and we chose to keep only the first p principal components. The vector v is transformed into a vector $t \in \mathbb{R}^p$ by means of a linear projection: $t = W_p v$, Where W_p is the truncated transformation matrix and $p = 500$.

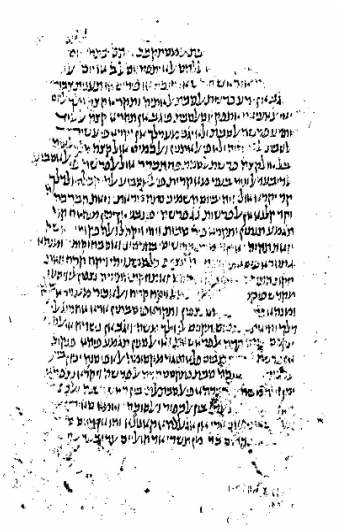
7.4 Query

Given a query image, it is treated as explained above, and is represented by a vector in $t \in \mathbb{R}^{500}$. This vector is then compared, by means of $L2$ distance, to the similar vectors-computed in the same way. All candidate targets are ranked in accordance with the computed distance. To eliminate multiple occurrences of the same target word we use the heuristic step described in the word spotting chapter.

7.5 Using the Genizah engine

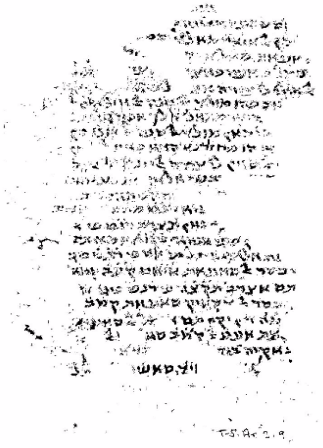
The user chooses a word to be queried by manually marking a patch from a certain document. Since the system requires large computational resources the word can be chosen from any document in the Genizah but will only be searched in a small collection of documents. For now, the results are given from the Arabic-Jewish collection of the 'Taylor-Schechter Genizah' from Cambridge University Library. The results are sent by email a few minutes after the user marks the word. See Figures 7.2 and 7.3 for some examples.

(a)



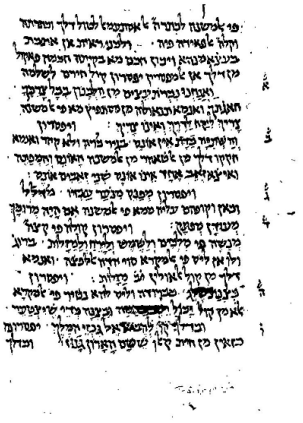
(b)

(c)



(d)

(e)



(f)

Figure 7.1: Samples of Genizah fragments. Images (a,c,e) on the left column are the original scans, images (b,d,f) are after binarization process. It is easy to notice the poor quality of the manuscripts.

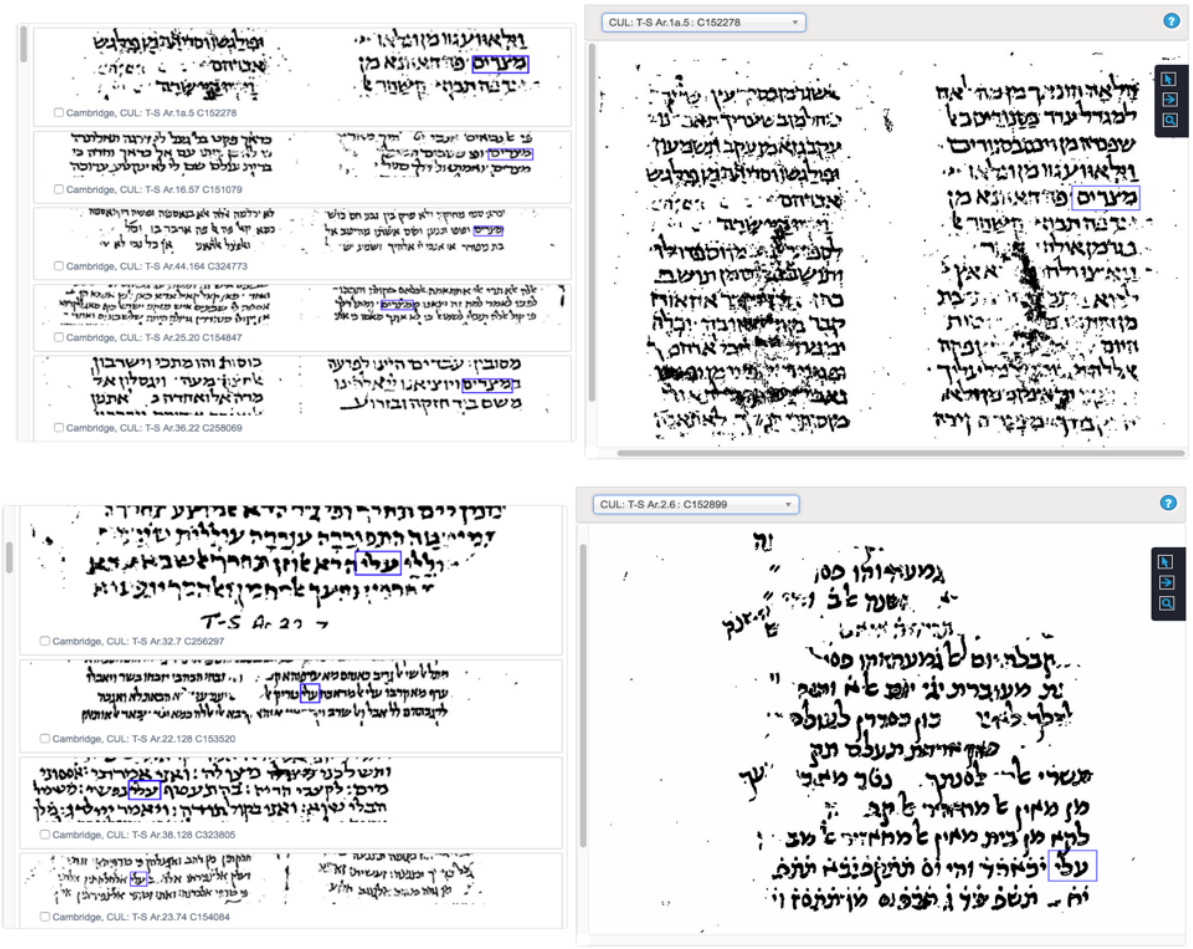


Figure 7.2: Top retrievals from the Genizah word-spotting engine as it is presented on the website for two sample queries. On the right is the manual selection, and on the left the retrieved results. In these two examples top results are correct.

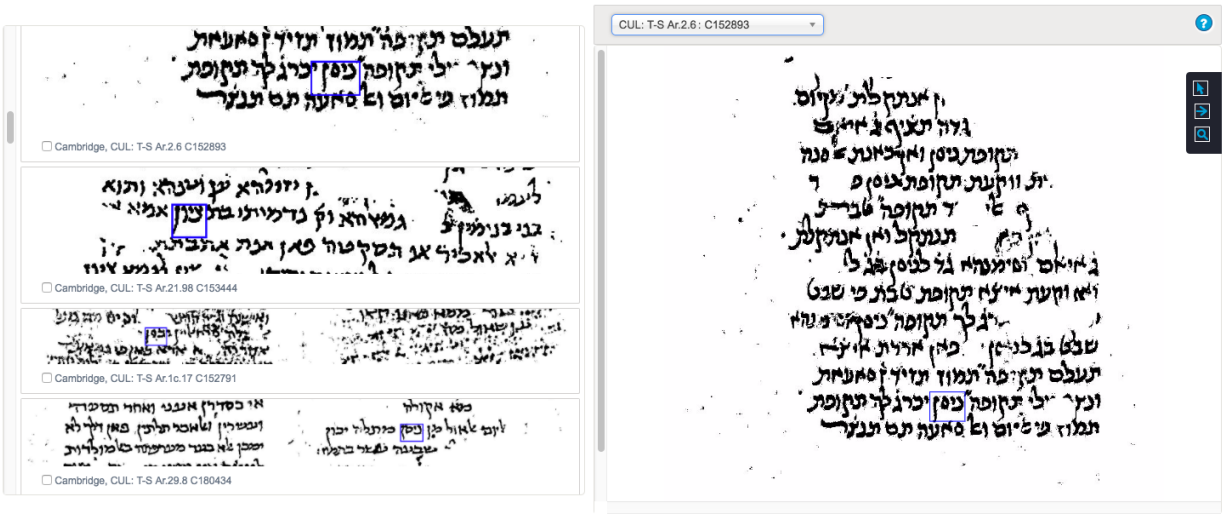


Figure 7.3: Top retrievals from the Genizah word-spotting. In this example second and third results are mistakes.

Chapter 8

Discussion

Currently, as quality OCR technologies are still lacking when dealing with handwritten documents, and especially with historical manuscripts, word-spotting technologies provide a useful substitute. The underlying word-spotting engine is extremely efficient. Indexing the words of each manuscript page is done in a few seconds, depending on the page complexity, and retrieval requires a fraction of a second. Though this may not be the most accurate spotting technology in existence, it is certainly the simplest method providing a level of accuracy within reach of state-of-the-art-ones. Specifically, no optimization is performed during preprocessing or during querying, representation is easily stored in conventional data structures, and the whole pipeline is completely unsupervised. This contrasts with methods that employ techniques such as neural networks, SVM or DTW. This flexibility implies that it can be very useful in practice - either as is, or as a plug-in method that can provide a list of candidates to another system.

Our main contribution in this work is presenting various applications to the underlying word spotting engine in historical documents.

1. In this work, we develop an unsupervised method for the improvement of OCR results that utilizes word-spotting. The method has the advantage that no ground truth OCR is employed during its application. Indeed, we use ground truth only for the purpose of experimental evaluation. While it is possible to use ground truth combined with word spotting, in a similar manner, in order to obtain more accurate OCR results. This is of much less interest to the current effort, since it would lead to a fully supervised OCR method. The main advantage of the proposed method is that it can effectively utilize new collections and adapt to them, in order to improve OCR results, without any additional labeling effort. We are not aware of any other similarly unsupervised method. The scalability and automatic nature of our method imply that it has the potential of becoming very useful in practice. It remains to be seen whether OCR betterment can also be achieved on a larger scale or on scripts with better developed OCR engines.

2. We have shown in this work how historical newspaper articles can be tied together into a

continuous story with high precision. We have developed a system which enables scholars to mine huge mass of historical information that is now available on-line in a digital form. It is an unsupervised image-based system for the retrieval of related articles within a scope of one month. Our system utilizes the underlying word spotting engine. It does not need any labeling nor ground truth, and it has the advantage that no OCR is needed and that the document images are its sole input. We have shown the feasibility of an image-based approach, and showed it performs well even comparing to a text-based system. The comparison was done mainly for experimental evaluation purposes, and such a comparison is less interesting to us, since this system can work even when no text is available. Since no optimization is necessary and the whole process is unsupervised, this method can be useful and can easily be extended to larger periods of time and can be adopted to different types of datasets. For further work, one can try to improve the accuracy of the retrieval by combining our system with the text based system in cases where noisy OCR is available.

3. Lastly we presented a real-time word spotting engine incorporated with the Cairo Genizah dataset, which includes more than 200,000 fragments. The documents are poorly preserved, were written in multiple hands and in multiple languages and styles, and thus for many queries the retrieved results are often incorrect. Improvement could be done by post-processing the results by filtering candidates with incompatible aspect ratio for instance. More accurate word segmentation could significantly improve the systems performance and could also facilitate the large memory usage as well. Using this engine is extremely easy. A query can be manually chosen online and within a few minutes the results are sent by mail to the user. To the best of our knowledge this is the first on-line word spotting engine implemented in this scale. This operational engine provides new ways for scholars to conduct their research of the Cairo Genizah's documents and enables them to search within an unlabeled massive collection. We believe that due to its flexible nature it can be incorporated with different datasets with only few minor adjustments.

Bibliography

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, Dec 2006. 11

[2] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, Lansdowne, VA, USA, 1998. 8

[3] J. Allan, S. Harding, D. Fisher, A. Bolivar, S. Guzman-Lara, and P. Amstutz. Taking topic detection from evaluation to practice. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 101a–101a, Jan 2005. 8

[4] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Efficient exemplar word spotting. In *BMVC*, 2012. 6, 7

[5] J. Almazn, A. Gordo, A. Forns, and E. Valveny. Word spotting and recognition with embedded attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2552–2566, Dec 2014. 6

[6] B.B Chaudhuri and U Pal. A complete printed bangla OCR system. *Pattern Recognition*, 31(5):531 – 549, 1998. 15

[7] Bidyut B. Chaudhuri. *On OCR of a Printed Indian Script*, pages 99–119. Springer London, London, 2007. 15

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005. 11

[9] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez. Improving offline handwritten text recognition with hybrid HMM/ANN models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):767–779, April 2011. 6

[10] A. Fischer, A. Keller, V. Frinken, and H. Bunke. HMM-based Word Spotting in Handwritten Documents Using Subword Models. In *2010 20th International Conference on Pattern Recognition*, pages 3416–3419, Aug 2010. 6

[11] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke. A novel word spotting method based on recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):211–224, Feb 2012. 6

[12] Basilis Gatos and Ioannis Pratikakis. Segmentation-free word spotting in historical printed documents. In *Proceedings of the 10th International Conference on Document Analysis and Recognition*, ICDAR, pages 271–275, Washington, DC, USA, 2009. IEEE Computer Society. 6

[13] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. 22

[14] Nadav Har'el and Dan Kenigsberg. HSpell - the Free Hebrew Spell Checker and morphological analyzer, 2004. 26

[15] Anna Huang. Similarity measures for text document clustering, 2008. 22

[16] I. Jolliffe. Principal component analysis, 2005. 30

[17] MILA The knowledge center for processing Hebrew. *http://www.mila.cs.technion.ac.il*. 22

[18] A. Kovalchuk, L. Wolf, and N. Dershowitz. A simple and fast word spotting method. In *ICFHR*, 2014. 1, 6, 7, 10, 14, 17, 29

[19] Qianli Liao, Joel Z. Leibo, Youssef Mroueh, and Tomaso A. Poggio. Can a biologically-plausible hierarchy effectively replace face detection, alignment, and recognition pipelines? *CoRR*, abs/1311.4082, 2013. 6, 7

[20] R. Manmatha and Nitin Srimal. *Scale Space Technique for Word Segmentation in Handwritten Documents*, pages 22–33. Springer Berlin Heidelberg, 1999. 6

[21] Sean Moran, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Enhancing first story detection using word embeddings. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, pages 821–824, New York, NY, USA, 2016. ACM. 8

[22] A. J. Newell and L. D. Griffin. Multiscale histogram of oriented gradient descriptors for robust character recognition. In *2011 International Conference on Document Analysis and Recognition*, pages 1085–1089, Sept 2011. 6

[23] Miles Osborne and Victor Lavrenko. Using paraphrases for improving first story detection in news and twitter. In *In HLT-NAACL*, pages 338–346, 2012. 8

[24] K. Pramod Sankar, R. Manmatha, and C. V. Jawahar. Large scale document image retrieval by automatic word annotation. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(1):1–17, 2014. 6

[25] The Friedberg Genizah Project. *http://www.genizah.org/TheCairoGenizah.aspx*. 5

[26] The Jewish Historical Press Project. *http://jpress.nli.org.il*. 4

[27] T. M. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 218–222 vol.1, Aug 2003. 6

[28] J. A. Rodriguez and F. Perronnim. Local gradient histogram features for word spotting in unconstrained handwritten documents. In *ICFHR*, 2008. 6

[29] L. Rothacker, M. Rusiol, and G. A. Fink. Bag-of-features HMMs for segmentation-free word spotting in handwritten documents. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1305–1309, Aug 2013. 6

[30] M. Rusinol, D. Aldavert, R. Toledo, and J. Llados. Browsing heterogeneous document collections by a segmentation-free word spotting method. In *2011 International Conference on Document Analysis and Recognition*, pages 63–67, Sept 2011. 6

[31] Maral Rusiol, David Aldavert, Ricardo Toledo, and Josep Llads. Efficient segmentation-free keyword spotting in historical document collections. *Pattern Recognition*, 48(2):545 – 555, 2015. 6

[32] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1605–1614, 2006. 6

[33] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. In *INFORMATION PROCESSING AND MANAGEMENT*, pages 513–523, 1988. 8

[34] Pramod Sankar K., C. V. Jawahar, and R. Manmatha. Nearest neighbor based collection OCR. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 207–214, New York, NY, USA, 2010. ACM. 7

[35] Martijn Spitters and Wessel Kraaij. Tno at tdt2001: Language model-based topic detection. 2001. 8

[36] Martijn Spitters and Wessel Kraaij. Using language models for tracking events of interest over time. In *In Proceedings of LMIR 2001*, pages 60–65, 2001. 8

[37] S. Sudholt and G. A. Fink. PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 277–282, Oct 2016. 6

[38] K E A van de Sande, J.R.R. Uijlings, T Gevers, and A.W.M. Smeulders. Segmentation as Selective Search for Object Recognition. In *ICCV*, 2011. 6

[39] Kai Wang and Serge Belongie. *Word Spotting in the Wild*, pages 591–604. Springer Berlin Heidelberg, 2010. 6

[40] C. Wieprecht, L. Rothacker, and G. A. Fink. Word spotting in historical document collections with online-handwritten queries. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 162–167, April 2016. 7

[41] Tomas Wilkinson and Anders Brun. *Visualizing Document Image Collections Using Image-Based Word Clouds*, pages 297–306. Springer International Publishing, Cham, 2015. 9

[42] J. P. Yamron, S. Knecht, and P. Van Mulbregt. Dragon's tracking and detection systems for the TDT2000 evaluation. In *In Proceedings of the Broadcast News Transcription and Understanding Workshop*, pages 75–79, 2000. 8

תקציר

מסמכים היסטוריים רבים עוברים תהליך של דיגיטציה בשנים האחרונות, עובדה שתורמת לכך שישנם מאגרי מידע רבים המכילים מסמכים מקוונים. דבר זה נותן לחוקרים גישה נוחה ומהירה למסמכים ועדויות על תרבויות, ומאפשר לחקור את המקורות הללו. איכות ה - OCR (אלגוריתם להמרה של תמונה לטקסט) עדיין לוקה בחסר עבור כתבי יד היסטוריים, עבור מסמכים בדפוס מיושן ואף עבור שפות לא נפוצות. בתור אלטרנטיבה, או בנוסף, ניתן להריץ חיפוש מבוסס תמונה. בהינתן תמונה של מילה, נחפש מופעים של אותה מילה במקומות שונים במסמכים. מנוע פשוט ומהיר לזיהוי מילים במסמכים היסטוריים משמש בעבודה זו עבור שלוש אפליקציות שונות;

אנחנו מציעים אלגוריתם יעיל ולא מונחה (unsupervised) לשיפור OCR. האלגוריתם עושה שימוש במנוע לחישוב OCR כקופסא שחורה ומקבל כקלט תמונות של מסמכים לא מתויגים. בהינתן מסמך חדש לניתוח, הוא נכנס לקופסא השחורה, ועבור כל תוצאה של ה – OCR נריץ חיפוש מבוסס תמונה לזיהוי המופעים של אותה מילה באוסף המסמכים. מבין התוצאות שנקבל נציע מועמד להחלפת ה- OCR על סמך שילוב מדד דמיון ויזואלי ודמיון טקסטואלי (edit distance). בנוסף, אנו מציגים בעבודה זו גישה מבוססת תמונות למציאת מאמרים בעיתון העוסקים באותו אירוע. בשלב מקדים, ובהינתן מילון, נייצר עבור המילים בו תמונות סינטטיות. בהינתן סט תמונות של מאמרים לא מתויגים, נכניס אותם למנוע לזיהוי מילים עבור כל המילים במילון, ובהתבסס על התוצאות נייצר וקטור לייצוג כל מאמר בהתבסס על אלגוריתם Tf-Idf. גם שיטה זו היא unsupervised, ואת חיפוש המאמרים נבצע באמצעות nearest neighbor.

השימוש האחרון שאנו מציעים הוא מנוע מקוון לזיהוי מילים בזמן אמיתי. פיתחנו בשיתוף פעולה עם פרויקט פרידברג לחקר הגניזה מנוע המשולב במאגר תמונות בסדר גודל ענק – גניזת קהיר.

אוניברסיטת תל-אביב TEL AVIV UNIVERSITY

הפקולטה להנדסה ע"ש איבי ואלדר פליישמן
בית הספר לתארים מתקדמים ע"ש זנדמן-סליינר

שימוש בזיהוי מילים על בסיס תמונות לאפליקציות של עיבוד מסמכים היסטוריים

חיבור זה הוגש כחלק מהדרישות לקבלת תואר

"מוסמך אוניברסיטה" – M.Sc.

באוניברסיטת תל-אביב
ביה"ס להנדסת חשמל

על ידי

עדי זילברפניג

העבודה הוכנה בהנחיית

פרופ' ליאור וולף

פברואר 2017