Arabic Handwritten Word Spotting Using Language Models

Muna Khayyat, Louisa Lam and Ching Y. Suen
Centre for Pattern Recognition and Machine Intelligence
Concordia University
Montreal, Quebec H3G 1M8, Canada
m_khay, llam, suen@encs.concordia.ca

Abstract

With the ever-increasing amounts of published materials being made available, developing efficient means of locating target items has become a subject of significant interest. Among the approaches adopted for this purpose is word spotting, which enables the identification of documents through the use of pertinent keywords. This paper reports on an effective method of word spotting for Arabic handwritten documents that takes into consideration the nature of Arabic handwriting. Parts of Arabic Words (PAWs) form the basic components of this search process, and a hierarchical classifier (consisting of a set of classifiers each trained on a different part of the input pattern) is implemented. For the first time in Arabic word spotting, language models are incorporated into the process of reconstructing words from PAWs. Details of the method and promising experimental results are also presented.

1. Introduction

Large numbers of documents have been and continue to be digitized, resulting in an increasing need for effective methods to search and index these documents to make their contents more accessible. Many word spotting techniques have been proposed for this purpose, especially for Latin-based and Chinese languages. However, not much work has been done on Arabic word spotting.

Arabic script is always cursive even when printed, and it is written horizontally from right to left. Words consist of connected components or sub-words, and these are often called Pieces of Arabic Words (PAWs) in the literature. In Arabic script, there is no difference in the within word space (i.e. the space between the PAWs) and the between words space. This is illustrated in figure 1. This lack of clear boundaries between words, together with the fact that Arabic writing is naturally cursive and more unconstrained than in other languages, make word spotting in the Arabic language a challenging task in need of further research.

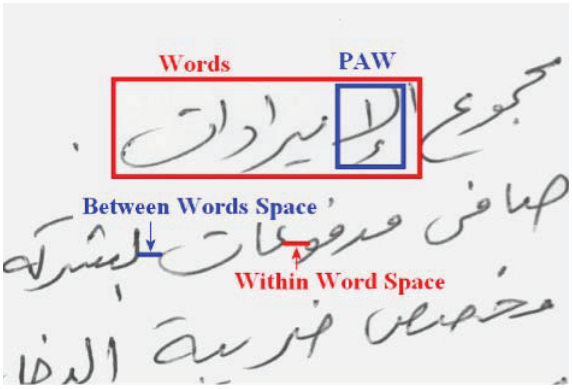

Figure 1. Arabic Script.

Arabic word spotting approaches tend to segment documents into PAWs rather than words, and then find ways to reconstruct the words from the PAWs [10] [9] [7]. Some approaches only spot PAWs (sub-words) and others try to spot words by reconstructing the words from their PAWs. We propose a learning-based word spotting system that partially segments the selected lexicon words and the documents into PAWs. For the first time in the literature of Arabic word spotting, language models will be integrated with the partial segmentation of the words, to represent contextual information and reconstruct words.

This paper is organized as follows: Section 2 contains a literature review, Section 3 describes the details of our word spotting method, Section 4 presents the experimental results, and we conclude this work in Section 5.

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.183

2. Related Work

Many Arabic word spotting methods avoid segmenting documents into words due to the problem of not having clear boundaries for words. Sarii and Kefali [10] preferred to segment the document into major connected components (PAWs), to circumvent the problem of word segmentation in Arabic documents. Thus, they decided to favor Arabic sub-word processing instead of words. They converted the PAWs into Word Shape Tokens (WST) in which they represented each PAW by global structural features. Similarly, input queries were coded and then a string matching technique was applied to reconstruct words from the PAWs. They validated their word spotting system using both printed and handwritten Arabic manuscripts and historical documents. This approach uses open lexicons and avoids pre-clustering.

Saabni and El-Sana [9] segmented the documents into PAWs to avoid pre-clustering; they used Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs) for matching in two different systems, and then additional strokes were used by means of a rule-based system to determine the final match. Similarly, Moghaddam and Cheriet [7] proposed an Arabic word spotting system that is based on shape matching. They extracted the connected components from the documents and then created their library of PAWs (basic connected components) using an Euclidean distance technique and DTW. Then PAWs were clustered into meta-classes to improve the accuracy and reduce the computational complexity. Both approaches [9] and [7] searched for PAWs rather than words, and they were tested on historical Arabic documents.

3. Proposed Method

We aim to search for lexicon words within Arabic handwritten documents. Our method is based on the partial segmentation of the lexicon and the documents into PAWs, to overcome the lack of boundaries problem. The segmented PAWs are passed to a hierarchical classifier to perform the final classification or arrive at a rejection decision. Figure 2 shows the block diagram of this method.

Our system is a learning-based word spotting system for which there are training data consisting of samples of the lexicon words (Words Database) and a separate set of Testing Documents. Each lexicon word in the Words Database is partially segmented into its constituent components or PAWs by first segmenting the word into its connected components. Large connected components and those crossing the base line are consid-

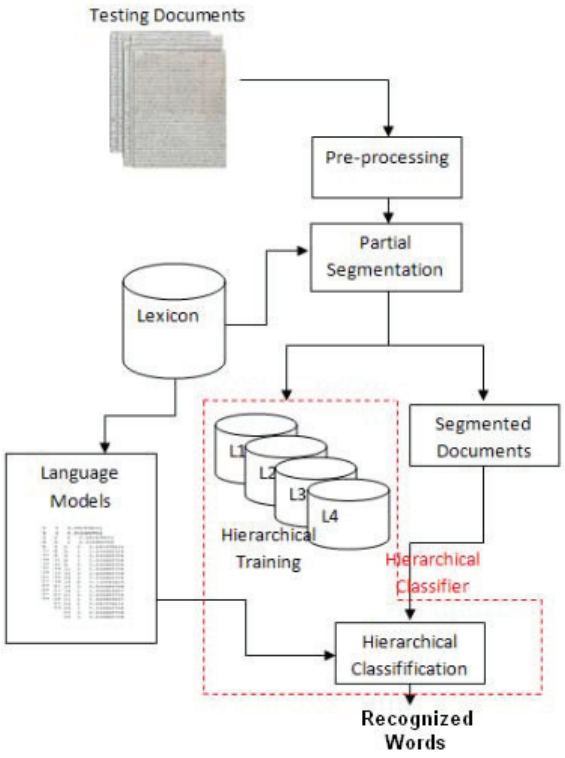

Figure 2. Block diagram of the proposed method.

ered major components, while the rest are considered minor components, each of which is considered a part of its closest major component. Figure 3 shows two major connected components with their related minor components. The resulting major components form the PAWs of the words database. These PAWs are assigned to groups according to their locations within words, and then each group is trained by a classifier. This will result in a sequence of different classifiers that form the hierarchical classifier.

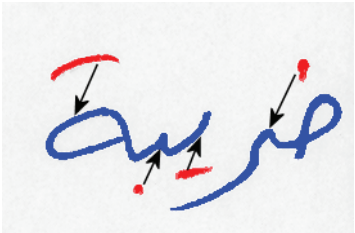

Figure 3. Major and minor connected components.

For the testing documents, the text lines are partially segmented into PAWs as described above, then each PAW is passed to the hierarchical classifier. A graph

is created for the PAWs with confidence values above a predefined threshold. Then the paths of the graphs are evaluated to decide whether to spot or reject the word within the path. The following sections describe our method in detail.

3.1. Preprocessing and Feature Extraction

When a document image is passed into the system, a simple Gaussian noise removal algorithm is applied, the document is binarized using the Otsu algorithm [8], and then the document is segmented into text lines using morphological dilation. Details of the process are presented in [5]. This process was designed to obtain homogeneous results from documents originally processed with different grey scales levels, and it also creates the possibility of working with binary document images.

Text lines and word images of the lexicon are then partially segmented into PAWs, which are normalized to 50×70 pixels. A mean filter is applied to the binary images to obtain gray scale images for feature extraction. Finally, gradient features [12] are extracted from the images.

3.2. Partial Segmentation

Partial segmentation refers to the process of segmenting the word images in $\mathcal{W}$ into PAWs in $\mathcal{S}$, where
$\mathcal{W}$ is the set of all images from the lexicon L,
W_i is the subset of all words in class i, ($\mathcal{W} = \bigcup_{i=1}^{n} W_i$ and n is the number of word classes).
$\mathcal{S}$ is the set of all PAW images obtained from segmenting the images in $\mathcal{W}$ (S represents sub-words),
S_j is the subset of all PAWs in class j, ($\mathcal{S} = \bigcup_{j=1}^{m} S_j$ and m is the number of PAW classes).
Thus, each word image $w \in \mathcal{W}$ is represented as a sequence of PAWs in $\mathcal{S}$.

Each PAW $s \in S$ is assigned to a unique class S_j. Each word can consist of one to four PAWs. This partial segmentation results in a new database LW of PAWs instead of words. LW is segmented into LW_i ($1 \leq i \leq 4$), where i is the location of the PAW according to the writing sequence ($LW = \cup_i LW_i$).

Similarly, each text line of a testing document is partially segmented into PAWs.

3.3. Language Models

Language models are integrated into our system to determine the probability of a sequence of PAWs within the lexicon words [4]. In our work we use unigram, bigram, trigram and 4-gram word-based language models to evaluate the paths in the graph. We define first the unigram, and then the n-gram for $n \geq 2$.

For a PAW $s_i \in S$ (i.e., the PAW belongs to class i), the strength α_i of s_i is defined as follows:
Suppose s_i occurs in the n lexicon words $\{W_i^1, W_i^2, ..., W_i^n\}$,
l_i^j denotes the location of s_i in word W_i^j ($1 \leq j \leq n$),
f_i^j is a factor determined by the number of PAWs following s_i in word W_i^j, then

$$\alpha_i = \sum_{j=1}^{n} l_i^j f_i^j \tag{1}$$

is the unigram probability the PAW s_i in the database.

For n-grams, where $n = 2, 3$, and 4 the probabilities are defined as follows:

$$P(s_i|s_{i-n+1}...s_{i-1}) = \frac{C(s_{i-n+1}...s_i)}{C(s_{i-n+1}...s_{i-1})} \tag{2}$$

where i is the position of the PAW ($i \leq n$), $P(s_i|s_{i-n+1}...s_{i-1})$ are the n-gram probabilities that define the language model, and $C(s_i...s_n)$ is the number of occurrences of the sequence of PAWs $s_i...s_n$ in the isolated words lexicon.

3.4. Hierarchical Classifier

The hierarchical classifier consists of a set of classifiers each of which is trained on a different part of the patterns. The longest Arabic word in the words database (described in 4.2) has four PAWs. Thus, four databases are constructed from the PAWs database, with each database containing the set of PAWs in the corresponding position within the words. Four Support Vector Machines (SVMs) [3] are trained separately on these four sets of PAWs.

The word spotting system segments the document into PAW images which are passed to the first classifier. For any PAW image s, if the highest confidence value produced by this classifier exceeds a predefined threshold $t1$,then all PAW candidate classes S_i with confidence values higher than another threshold $t2$ (where $t2 < t1$) would be added with their confidence values as nodes to the graph. This adds more flexibility to the system, in which not only the first candidate class S_i is considered, but also some (up to three) PAW classes with lower confidence values. The thresholds t1 and t2 are determined from the posterior probabilities (confidence values) of the words in the isolated words database.

If the graph has at least one non-leaf node, then the following PAW image s is passed to the second classifier and similarly new nodes are added to the graph. The same strategy is repeated until all the paths in the graph end with leaf nodes.

3.5. Path Evaluation

Each node in the graph represents a candidate PAW. The score of the node is calculated using the confidence values of the PAW classifier, while the links between PAWs are evaluated using the n-grams (language models) probabilities.

The language models are utilized to create links between pairs of nodes, to indicate wether the two PAWs are contiguous PAWs in the lexicon or not. Depending on the results of the language models, links between the nodes in the first and second levels are added. The case is similar for the other levels. The node is considered a leaf node if the candidate PAW in the node cannot be extended as decided from the language models.

A path is a set of links between nodes. The path may produce one of the lexicon words depending on its value R defined below. If the path ends with a leaf node, and R satisfies the following inequality, then a lexicon word is found (spotted). Otherwise, the path is rejected:

$$R = \frac{1}{n}\sum_{i=1}^{n} \alpha_{S_j} |\ln P(s|S_j)| + \frac{1}{q}\sum_{l=1}^{q} P(S_j) \leq t \quad (3)$$

S_j is the PAW class, s is the PAW tested by the classifier, $P(S_j)$ is the probability from the language models, q and n are the numbers of paths and nodes within the path respectively, and t is the threshold for accepting or rejecting the path.

3.6. Word Reconstruction

The lexicon words are represented in a look up table, where each word is assigned a unique key of 8 digits as follows:

$$AABBCCDD$$

AA represents the first PAW class ID; if this ID has only one digit, then the left digit is zero. Similarly, BB, CC, and DD are the two-digit representations of second, third and fourth PAW class ID's.

Accepted paths are assigned keys as described above. Then the key is passed to a search algorithm to look for the word within the table representing the lexicon. If the key is found, then the word is reconstructed. Otherwise, the PAW sequence is not a valid one.

Hence, the language models described in section 3.3 have a significant influence on the number of incorrect paths. Integrating only bigram models to create the paths will result in more incorrect paths, while integrating both bigram and trigram models will definitely reduce the number of incorrect paths. Integrating language models with higher n-gram models should reduce the number of incorrect paths in general. However, in the Arabic language there are few words with more than 4 PAWs, therefore integrating higher than trigram models will produce many less incorrect paths for this language.

4. Experimental Results

Our word spotting system was evaluated using two CENPARMI databases (Section 4.2). The Arabic documents database contains documents written by different writers with various styles, and the isolated Arabic handwritten words to represent the lexicon L.

The proposed system can spot words consisting of one to four PAWs in Arabic handwritten documents. The precision PR, Recall RR, and $f1_{score}$ rates of the system are $65\%, 53\%$, and 58.2% respectively when the threshold $t = 0.1$ in equation (3).

4.1 Prior Results on Arabic Word Spotting

Ball et al. [2] presented three Arabic handwritten word spotting systems based on different approaches to segmentation: segmenting a document into characters, into words, and a manual segmentation of the document into words. The systems were evaluated using the CEDARABIC documents written by 10 writers. With an RR of 50%, the systems reported PR's of 28%, 34%, and 65% respectively. At the same time, Leydier et al. [6] and Shahab et al. [11] presented Arabic word spotting systems that were evaluated on documents provided by a single writer, and only one query of one PAW was tested in the former. These approaches resulted in PR 's of 72.5% and 80% respectively.

In comparison, our system was designed to spot complete words in documents, and it was tested on 43 documents freely written by 24 writers in their own styles (described in Section 4.2). These comparisons would support the validity of our approach.

4.2. Databases

The CENPARMI database of Arabic off-line handwritten words [1] is used as the lexicon L for the proposed method. The database contains 69 words that include some commercial terms, together with words used in weights, measurements, and currencies of Saudi Arabia. These words are segmented into PAWs, resulting in 23, 23, 20, and 3 words of one, two, three, and four PAWs respectively. The total number of PAW classes within our lexicon is 92.

The performance of our method was tested using the CENPARMI Arabic handwritten documents database.

We used 43 documents written by 24 writers with a total of 850 words.

4.3. Results and Analysis

Testing documents were processed by three different systems. The first one uses only bigrams to create the links between the nodes in the graph. The second uses bigrams and trigrams to create the links and the last one also includes 4-grams. The evaluation program calculates the follwing: True Positives (TP), False Negatives (FP), and False Negatives (FN) for all the words, as well as for the words of 1, 2, 3, and 4 PAWs respectively. The numbers of incorrect paths that were sent to the search algorithm were also compiled. The three systems produced identical results except for the different numbers of incorrect paths.

Table 1 shows the PR, RR, $f1_{score}$, and the number of incorrect paths when the bigram models and also when both bigram and trigram models are integrated.

				No. of incorrect paths	
t	PR	RR	$f1_{score}$	**Bigrams**	**Bigrams + Trigrams**
0.05	0.76	0.42	0.54	33	10
0.10	0.65	0.53	0.58	108	13
0.15	0.58	0.56	0.57	135	13
0.20	0.53	0.59	0.56	157	18
0.25	0.50	0.62	0.55	159	19

Table 1. Experimental Results with different thresholds.

Figure 4 shows the Recall - Precision curves of our system. Different combinations of language models produce identical Recall - Precision curves, and they only affect the speed of the system. Changing the values of parameters l and f will affect the system results of α_i and therefore R as shown in equations (1) and (3), with the consequence that some previously accepted paths may be rejected and vice versa.

Moreover, changing the thresholds $t1$ and $t2$ (explained in section 3.4) may change the number of candidate PAWs in the graph. This may increase the recall rate RR, while lowering the precision rate PR. The reason is that nodes with low confidence values will be added to the graph, which means different writing styles can be tolerated. In addition, the Arabic alphabet contains very similar characters that are difficult to differentiate separately, and would rely on the context for identification. In our system, reducing $t1$ and $t2$ adds all likely candidate PAWs to the system, and the language models would resolve the ambiguities.

Some words were not detected by our system, since they include touching PAWs. Other words are ambiguous and not well written which make them difficult even for an Arabic native reader to decipher. Some words were scratched and then the writer tried to write them again above or nearby. These errors all added to the FNs. Moreover, some words with the same roots as the lexicon words were incorrectly detected, which added to the FPs.

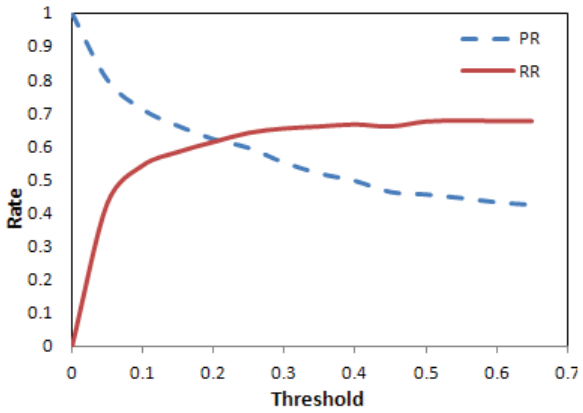

Figure 4. Precision and Recall Rates vs. Threshold.

Figure 5 shows the numbers of incorrect paths when bigram models are used, and also when trigrams are integrated to the system with bigrams. The results show that integrating only bigram models significantly increases the number of incorrect paths, while integrating trigram models can filter out incorrect paths using the langauge model. Integrating 4-gram language models will not result in any incorrect path.

The lexicon that was used to evaluate our system is relatively small. However, evaluating our system with large lexicons of thousands of words will significantly slow down the system, since the system has to validate many additional paths. For large lexicons, efficient hash tables can be used to speed up the search and optimize the code.

Figure 6 shows the $f1_{score}$ curves for words of 1, 2, 3, and 4 PAWs. These curves appear to have similar behavior for words of 1, 2, and 3 PAWs respectively, while it is different for words with 4 PAWs. In general, the highest $f1_{score}$ is attained when the threshold $t = 0.1$, and the $f1_{score}$ decreases when t increases. But this is not the case for 4 PAWs, and it is due to the fact that there are few words with 4-PAWs in the testing documents. This means missing one such word will significantly decrease the recall rate and increase the precision

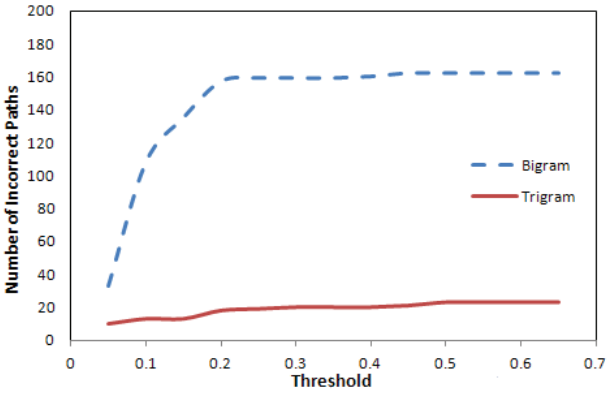

Figure 5. Number of incorrect paths vs. threshold.

rate. Our system is not biased to the number of PAWs that constitute the words, since the language models are capable of connecting these PAWs to reconstruct words.

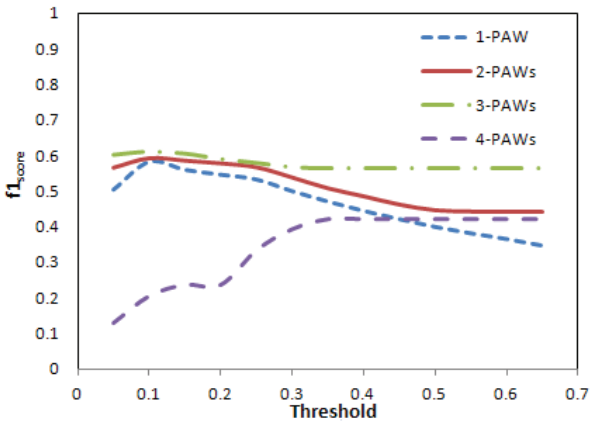

Figure 6. $f1_{score}$ curves of words with 1, 2, 3 and 4 PAWs.

5. Conclusion

We propose a word spotting system for Arabic handwritten documents that makes use of the partial segmentation of a lexicon (words database) and the testing documents into PAWs. The system includes a hierarchical classifier and it integrates partial segmentation with language models to spot Arabic handwritten words.

In the literature on Arabic word spotting, documents were either manually segmented into words, or PAWs were spotted rather than words. Our word spotting system can spot words consisting of different numbers of constituent PAWs.

We efficiently integrated language models into our system, to represent the contextual information of the words. This results in an effective method to reconstruct words from their connected components (PAWs).

Finally, our method was tested on documents written by different writers with promising results for Arabic word spotting.

References

[1] H. Alamri, J. Sadri, C. Y. Suen, and N. Nobile. A novel comprehensive database for Arabic off-line handwriting recognition. In *Proc. 11th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR)*, pages 664–669, 2008.

[2] G. Ball, S. N. Srihari, and H. Srinivasan. Segmentation-based and segmentation-free approaches to Arabic word spotting. In *Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 53–58, 2006.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001.

[4] J. T. Goodman. A bit of progress in language modeling: Extended version technical report. *Computer Speech and Language*, 15(4):403–434, 2001.

[5] M. Khayyat, L. Lam, C. Y. Suen, F. Yin, and C.-L. Liu. Arabic handwritten text line extraction by applying an adaptive mask to morphological dilation. In *10th IAPR Int. Workshop on Document Analysis Systems (DAS)*, March 2012.

[6] Y. Leydier, F. L. Bourgeois, and H. Emptoz. Text search for medieval manuscript images. *Pattern Recognition*, 40(12):3552–3567, 2007.

[7] R. Moghaddam and M. Cheriet. Application of multi-level classifier and clustering for automatic word spotting in historical document images. In *Proc. 10th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pages 511–515, 2009.

[8] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man, Cyber.*, 9(1):62–66, 1979.

[9] R. Saabni and J. El-Sana. Keyword searching for Arabic handwritten documents. In *Proc. 11th Int. Conf. on Frontiers in Handwriting Recognition, (ICFHR)*, pages 271–277, 2008.

[10] T. Sari and A. Kefali. A search engine for Arabic documents. In *Actes du dixiéme Colloque Int. Francophone sur l´Écrit et le Document*, pages 97–102, 2008.

[11] S. Shahab, W. G. Al-Khatib, and S. A. Mahmoud. Computer aided indexing of historicl manuscripts. In *Proc. Int. Conf. of Computer Graphics, Imaging and Vision (CGIV)*, pages 151–159, 2006.

[12] M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura. Handwritten numeral recognition using gradient and curvature of gray scale image. *Pattern Recognition*, 35(10):2051–2059, 2002.