Keyword Spotting Framework using Dynamic Background Model

Gaurav Kumar, Zhixin Shi, Srirangaraj Setlur, Venu Govindaraju
University at Buffalo
gauravku,zshi,setlur,govind@buffalo.edu

Sitaram Ramachandrula
HP Labs, Bangalore
sitaram@hp.com

Abstract

An important task in Keyword Spotting in handwritten documents is to separate Keywords from Non Keywords. Very often this is achieved by learning a filler or background model. A common method of building a background model is to allow all possible sequences or transitions of characters. However, due to large variation in handwriting styles, allowing all possible sequences of characters as background might result in an increased false reject. A weak background model could result in high false accept. We propose a novel way of learning the background model dynamically. The approach first used in word spotting in speech uses a feature vector of top K local scores per character and top N global scores of matching hypotheses. A two class classifier is learned on these features to classify between Keyword and Non Keyword.

1. Introduction

Keyword Spotting in handwritten documents is a task of locating and recognizing a given set of keywords in a document image or a set of document images. A number of approaches have been proposed in this area but it still remains an unsolved problem. The approaches applied to this problem have been broadly classified into two categories: template based matching [10][11][15] and recognition based matching [6][7][16][21]. In template based matching approach, similarity between a set of features extracted from the input image with standard templates of the keywords is calculated based on some distance metric. The recognition based approaches rely on the confidence scores returned by the recognizers. Such methods are further classified into segmentation based [9] and segmentation free approaches [7][16][21]. We focus on recognition based matching.

An important task in Spotting is to apply a best rejection criteria in order to reject non-keyword images. Very often a background or filler model [6][16] is used and a word image is accepted as a genuine keyword if the likelihood of the keyword model is more than that of the background. Designing a background model is a difficult task. Due to large variations in handwriting styles it is often the case that most of the genuine keywords have a better match for the background than the keywords themselves because the background model allows all possible sequence of characters. A relaxed background model could result in a higher false accept rate. Due to similarity between characters and due to lack of training data, the training might not be proper which also hampers the spotting accuracy. Xue [21] showed that the performance of a recognizer highly depends on its nature and the quality of the input image. He proposed a statistical method to learn the dependency of the recognizers with the lexicon words and demonstrated his findings on five different recognizers belonging to both segmentation free [20] and segmentation based categories[9]. Bouchaffra [1] proposed means to convert the confidence score returned by the recognizer to its true probabilistic measure.

The existing approaches in word spotting using statistical techniques such as HMM try to learn a background model by considering all training samples as single entity. This requires additional training for the background model and a large number of training samples. Also, they rely on certain threshold for rejection criteria. This work proposes an adaptive way of learning the background model without much prior training and which can be easily integrated with any recognizer. Since the model is learned dynamically, the dependencies of the recognizer with the lexicon is automatically taken into consideration. This idea was first applied in speech [2][4][8] where such technique was applied on spotting in continuous speech of natural numbers. One of the most significant advantage of this approach is that its simple to learn and because its dynamic and integrated with the recognizer, it nullifies lack of training. Secondly, its robust to any set of keywords and no training is required if set of keywords is changed. The

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.223

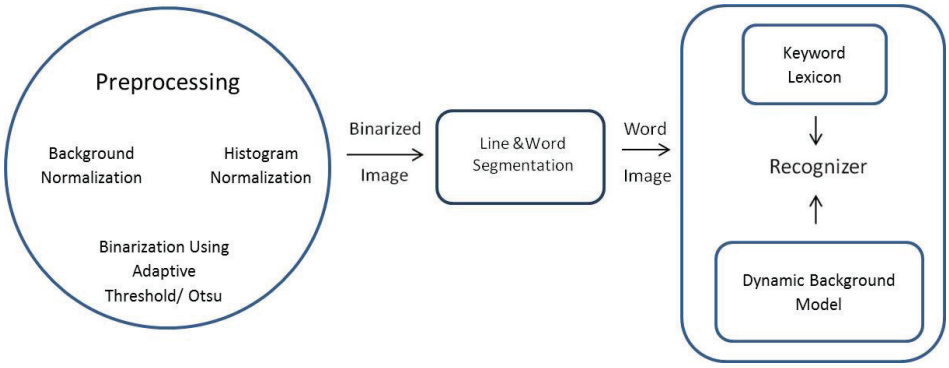


Figure 1: Spotting Framework

only limitation of this approach is that it relies on proper word segmentation. We show results on scanned images and camera images that suffer from problems of varied illumination, intensity and bleeding issues.

The remainder of the paper is organized as follows. We introduce the dynamic background model in Section 2. Section 3 covers a brief overview of the recognizer used in our system. Section 4 covers some of the preprocessing steps for camera images. Finally the experimental results and conclusion are in Section 5 and 6 respectively.

2. Dynamic Background Model

We propose a generic and robust keyword spotting framework that leverages the character level scores given a keyword and the global word level matching scores. In a recognizer based Keyword Spotting framework, given a keyword and an input image, the goal is to determine how confident is the recognizer in determining the closeness of the input image with the keyword. Most common approaches applied in recognition particularly in decoding stage are the common Viterbi algorithm applied in case of HMMs [14] and Dynamic Programming in case of other segmentation based approaches [9]. Both these approaches implicitly find out the best sequences of the observation that are best fit for the characters in the keyword. In other words, they find the best segmentation points or a combination of segmentation points that represent individual characters of the keyword. Let W = $[c_1, c_2, c_3, ..c_n]$ be a lexicon word where c_i represent each character and let X=$[x_1, x_2, x_3..x_m]$ be the best fit segmentation points that either individually or in combination represent each character. Let S=$[s_1, s_2, s_3..s_n]$ represent the final segments representing each character, where each s_i is a combination of one or more x_i. Given a genuine keyword image, each s_i has a very high confidence of it matching the character c_i. Conversely, for an input image not belonging to the keyword, the confidence would be low. The proposed dynamic background model learns these behavior.

The concept of dynamic background model was first proposed in speech [2] where local hypotheses at phoneme level and global hypotheses at word level were used to learn such models. Similar to this idea, the proposed dynamic background model uses a combination of local character matching scores and global word hypotheses scores as features. The first set of features, consisting of local character matching scores, are obtained using a two pass algorithm. In the first pass we find the best segmentation hypothesis and estimate the best segmentation points for the current image. In the second pass the *K* best matching scores per segment are obtained. The second set of features include the top *N* hypotheses scores at the word level. The length of resulting feature vector F is thus denoted by

$$|F(x)| = K * word\_length + N \tag{1}$$

These features are then normalized using the min-max normalization given by $Score(f_i) = \frac{f_i - min_i}{max_i}$ where f_i is the feature in ith dimension and min_i and max_i is the minimum and maximum of ith dimension.
The estimated segmentation points and corresponding score level features for a word image are shown in figure 2. At max four segments are combined per character and best character hypotheses matching the combination is considered. In the first pass, the recognizer finds the suitable start and end segments for each character of each word in the lexicon containing the set of keywords. In the second pass top *K* character matching scores for these combination is found. As elaborated earlier, the underlying assumption is that for a given keyword image the segments combination would be more appropriate resulting in a better local character matching scores. On the other hand if word image is not a keyword,

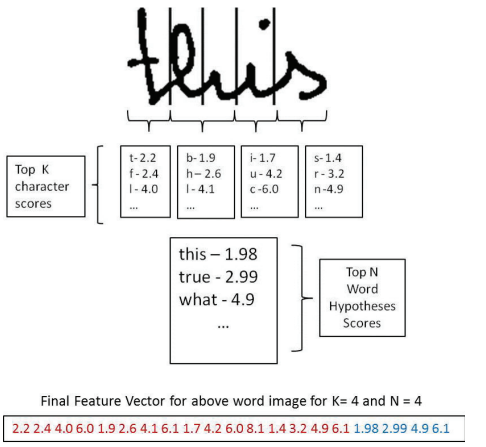


Figure 2: Segmentation Output and Feature Extraction At max four segments are combined to get the final hypotheses at character level.

the combination would be inappropriate resulting in a poorer local matching score. Similarly, the word hypotheses scores for Keywords would be better than the non-keywords. It was also observed that the difference between the top two word level hypotheses scores was higher in case of a keyword than a non-keyword. The addition of local character matching scores to our feature vector provides a smoothing factor to overall results and helps in disregarding non-keywords over keywords. Thus this approach easily learns the dependencies between a lexicon word and the recognizer and is simple to integrate.

2.1 Classification

The final stage of the framework is to learn a two class classifier to separate keywords from Non-keywords. Let $X \epsilon \mathbb{R}^m$ be the features extracted from R labeled samples of keywords and Non-keywords and let $Y \epsilon [0, 1]$ be the corresponding labels. A two class Logistic Regression (LR) and Support Vector Machines (SVM) [3] is trained on above labeled samples. Since a lexicon has keywords of varying length and the dimension of the feature vector depends on the length of the keyword as denoted by 1, we are dealing here with different length feature vectors belonging to same class. Logistic Regression (LR) tends to overcome this issue. We fix the size of the feature vector to be $K * max\_word\_length + N$ and set all unknown values to zero. We define the error function by taking the negative logarithm of the likelihood denoted by

$$E(w) = -ln\ p(y|w) \tag{2}$$

$$-ln\ p(y|w) = -\sum_{r=1}^{R} y_r ln f_r + (1 - y_r) ln(1 - f_r) \tag{3}$$

where y_r is the target label and f_r is denoted by the sigmoid function.

$$f_r = \sigma(w^T * \overrightarrow{x_r}) \tag{4}$$

Given enough labeled samples of variable length, weights w can be updated to allow varying length feature vector.

The second classifier used here is Support Vector Machines (SVM). The underlying assumption of using SVM is that the scores at the character and word level for a keyword and non-keyword are well separated in the higher dimension feature space and there exists a decision boundary to separate such hypotheses. The training involves minimizing of the error function

$$\frac{1}{2} w^T w + C \sum_{i=1}^{R} \xi_i \tag{5}$$

subject to the constraint

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,\ for\ \xi_i \geq 0, i = 1, ..R \tag{6}$$

Here C, is the Capacity constant used to penalize the error in classification during training, w is the vector coefficients and ϕ is the basis function applied on the training data x_i. We used a radial basis kernel for training.

3 Recognizer

The recognizer [9] used in our system is a segmentation based recognizer but the model can be integrated with HMM based, segmentation free recognizers as well. Given a set of lexicon words and an input image, it returns the lexicon words sorted in the increasing order of corresponding matching scores. The input image is first segmented into segments. A maximum of four segments is then combined and a 74 dimensional gradient based feature vectors is extracted. The distance between features extracted from each combination with trained character clusters is then calculated as below:

$$d(i) = min_j D(\overrightarrow{fv}, c(c_i, j)) \tag{7}$$

where d(i) is the minimum distance of feature vector of the combined segment with character c_i and (c_i, j) is the jth cluster center for c_i. $D(*, *)$ is the Euclidean distance between the feature vectors. Given a lexicon word the matching score is calculated using Dynamic

Programming. The score for a given Lexicon word L = $(c_1, c_2, c_3, .., c_k)$ is given by

$$S = min \frac{d_1^2 + d_2^2 + d_i^3 + .. + d_k^2}{k} \quad (8)$$

where d_i is the euclidean distance for each combination and a minimum score is taken over all possible groupings of segments to represent character. Reader is referred to [9] for further reading.

4. System Overview

The proposed approach was evaluated on document images taken from camera as well as from scanner. The overview of the framework is shown in figure 1. Images taken from Camera suffer from problems such as varying illumination, intensity, bleeding, perspective distortion and uneven background. We evaluate our approach on images without any perspective distortions or cluttered background. These will be considered in future work. The camera images require extra preprocessing before binarization as compared to the scanned Images. Some of the additional preprocessing steps are listed below.

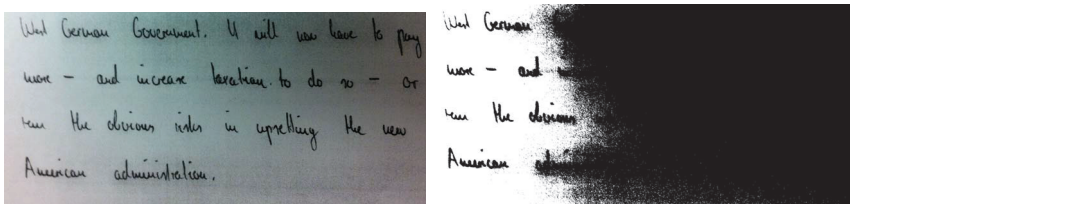
4.1 Preprocessing And Word Segmentation

The colored camera image is first converted into grayscale and background light intensity is normalized using a adaptive linear or non linear function [17] that best fits the background. The background normalized image is further enhanced by Histogram Normalization[18]. The algorithm computes the distribution over grey levels in the image. A percentage of both high and low intensity grey level values is ignored and remaining values are rescaled to range of 0-255. Zhixin applied these normalization techniques on camera images of Historical Documents with good results. Finally, the normalized image is binarized using an adaptive thresholding algorithm [5]. Five n*n windows are considered with one in the middle centered at pixel under consideration and 4 other blocks adjacent to the corner of the center block. A weighted difference between the average pixel intensity in the middle block and that in the other 4 blocks provides the center pixel's binary value. The binarization result is shown in figure 3e. The effect of Normalization on these camera images can be seen in figure 3d. On certain poor quality blurred images normalization improves the binarization results immensely. The scanned images were binarized using the standard otsu algorithm[13]. The binarized output is then line segmented using algorithm proposed by Zhixin [19] which uses steerable filter to convert a down sampled version of the input document image into Adaptive Local Connectivity Map (ALCM). Connected Component based grouping is done to extract each text line. Finally, Word Segmentation is done by finding convex hulls for each connected component and learning distribution over the distances between the centroids of the convex hulls for within and between word gaps. The word segmented output is shown in figure 3f.

5. Experimental Results

We evaluate our system on two datasets consisting of Camera Images and Scanned Images respectively. The Camera images dataset consisted of images of 100 handwritten documents from IAM dataset [12] taken from 5MP resolution camera. Each document contained on an average 8 lines and 60 words. Therefore total number of possible candidates were approximately 6000 and number of Keyword occurrences for lexicon size 10 were approximately 2.4%. A portion of sample image is shown in 3a. The images were background and histogram normalized and binarized using the adaptive thresholding algorithm followed by line and word segmentation. A two class SVM and Logistic Regression classifiers were trained on a total of 2000 Keyword and Non-keyword images not taken from the 100 documents used for evaluation. The values K and N were found empirically and top K local scores and top N word hypotheses scores were extracted and passed to Logistic Regression and SVM for this one time training. Best results were obtained for K = 2 and N = 4. The groundtruth of each document was used as lexicon to get the best matching scores for keyword for training. An arbitrary lexicon was used to get all non-keyword scores. Two separate experiments with Top 10 and Top 100 most frequent occurring terms in the IAM dataset, considered as keywords, were carried out . The average length of the keywords for lexicon of size 10 and 100 was approximately 7 characters. In both cases the system was evaluated in terms of average Precision defined by $\frac{TruePositive}{TruePositive+FalsePositive}$ and average Recall defined as $\frac{TruePositive}{TruePositive+FalseNegative}$ for both SVM and LR. The results are shown in Table 1. As evident both LR and SVM perform equally good. In case of LR, different threshold can be applied on sigmoid output. A high threshold would result in lower false accept and a low threshold would result in low false reject. With increase in the size of the lexicon the precision and recall decreases which is expected .Even, in case of SVM, there exist ways to measure the confidence where sigmoid function can be applied on distance from margin and a threshold can be applied on the sigmoid output.

The approach was also evaluated on 900 line images



(a) Original Image (b) Binarized Image Using Otsu

(c) Background Normalized Image (d) Histogram Normalized Image

(e) Final Binarized Image after c) d) and Adaptive Binarization (f) Word Segmentation Results (g) Spotting

Figure 3: Spotting Result for Camera Images. Binarization using global algorithms like otsu fail drastically for Camera Images due to varying light intensity and illumination. The normalization followed by an adaptive binarization gives far better binarization results. Spotting result show two True Positive and one False Positive.

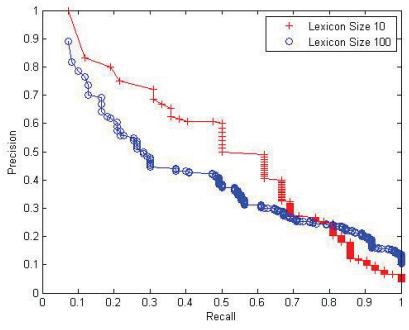


Figure 4: ROC for Scanned Images from IAM Dataset Using Logistic Regression

of IAM Dataset [12] on same keyword list of size 10 and 100. The line images were binarized using the otsu algorithm and same word segmentation algorithm was applied. The ROC curve for Logistic Regression is plotted considering different threshold from the output of sigmoid function. The plot is shown in figure 4. The best f-measure, which is weighted harmonic mean of precision and recall is shown in Table 2.

Table 2: Best F-measure for Scanned Images with LR

Lexicon Size	Precision	Recall	F-measure
10	0.4906	0.6190	0.5474
100	0.4173	0.4818	0.4473

6. Conclusion & Future Work

We proposed an adaptive way of learning the background model often required for Keyword Spotting in handwritten documents. The idea was first applied in speech and the results prove that they are applicable in handwriting as well. The only limitation of this approach is that it relies on proper word segmentation. However, it is adaptable and can be easily integrated with any recognizer. Although the method was applied on a segmentation based recognizer, a similar two pass algorithm can be applied on segmentation free recognizers as well. The normalization techniques used for camera images help in enhancing the images and segment out the background from foreground text. We focused on images with varying illumination and intensity

Table 1: Spotting Accuracy for Camera Images

	Lexicon Size 10		Lexicon Size 100	
Classifier	Avg. Precision	Avg. Recall	Avg. Precision	Avg. Recall
Logistic Regression	0.44	0.60	0.42	0.45
SVM	0.40	0.62	0.49	0.47

and some bleeding as well. Other issues such as perspective distortion will be considered in future works. Certain feature selection and active learning strategies can also be incorporated to improve the classifiers accuracy. Such methods would allow the selection of most distinguishable keywords and non-keywords features for learning a better classifier.

References

[1] D. Bouchaffra and V. Govindaraju. A methodology for mapping scores to probabilities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):923 –927, sep 1999.

[2] H. Bourlard, B. D'hoore, and J.-M. Boite. Optimizing recognition and rejection performance in wordspotting systems. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume i, pages I/373 –I/376 vol.1, apr 1994.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] C. De La Torre, L. Hernandez-Gomez, F. Caminero-Gil, and C. Del Alamo. On-line garbage modeling for word and utterance verification in natural numbers recognition. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 845 –848 vol. 2, may 1996.

[5] G. I. Ercole Giuliao, L. S. I. Orazio Paita, and A.-G. I. Luigi Stringa. Electronic character-reading system, 09 1977. US 4047152.

[6] A. Fischer, A. Keller, V. Frinken, and H. Bunke. Lexicon-free handwritten word spotting using character hmms. *Pattern Recogn. Lett.*, 33(7):934–942, May 2012.

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

[8] E. L. J. Caminero and L. Hernandez. Two-pass utterance verification algorithm for long natural numbers recognition. In *Proceedings of the International Conference on Spoken Language Processing*, pages 779–782, 1998.

[9] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:366–379, April 1997.

[10] V. Lavrenko, T. M. Rath, and et al. Holistic word recognition for handwritten historical documents, 2004.

[11] S. Madhvanath, E. Kleinberg, and V. Govindaraju. Holistic verification of handwritten phrases. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(12):1344 –1356, dec 1999.

[12] U.-V. Marti and H. Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39–46, 2002. 10.1007/s100320200071.

[13] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, Mar. 1979. minimize inter class variance.

[14] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[15] T. M. Rath and R. Manmatha. Word spotting for historical documents. *INTERNATIONAL JOURNAL ON DOCUMENT ANALYSIS AND RECOGNITION*, pages 139–152, 2007.

[16] J. A. Rodríguez-Serrano and F. Perronnin. Handwritten word-spotting using hidden markov models and universal vocabularies. *Pattern Recogn.*, 42:2106–2116, September 2009.

[17] Z. Shi and V. Govindaraju. Historical document image enhancement using background light intensity normalization. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 473 – 476 Vol.1, aug. 2004.

[18] Z. Shi, S. Setlur, and V. Govindaraju. Digital image enhancement using normalization techniques and their application to palm leaf manuscripts, 2005.

[19] Z. Shi, S. Setlur, and V. Govindaraju. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 176 –180, july 2009.

[20] H. Xue and V. Govindaraju. On the dependence of handwritten word recognizers on lexicons. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1553 – 1564, dec 2002.

[21] H. Xue and V. Govindaraju. Hidden markov models combining discrete symbols and continuous attributes in handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3):458 – 462, march 2006.