British Journal of Mathematics & Computer Science
12(4): 1-9, 2016, Article no.BJMCS.20376
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Recognition of Typewritten Characters Using Hidden Markov Models

I. A. Adeyanju[1*], O. S. Ojo[2] and E. O. Omidiora[2]

[1]Department of Computer Engineering, Federal University Oye-Ekiti, PMB 373, Oye-Ekiti, Ekiti State, Nigeria.
[2]Department of Computer Science and Engineering, Ladoke Akintola University of Technology, P.M.B 4000, Ogbomoso, Nigeria.

Article Information

DOI: 10.9734/BJMCS/2016/20376
Editor(s):
(1) Kai-Long Hsiao, Taiwan Shoufu University, Taiwan.
(2) Tian-Xiao He, Department of Mathematics and Computer Science, Illinois Wesleyan University, USA.
Reviewers:
(1) Ayman Shehata Mohammed, Assiut University, Egypt.
(2) S. K. Srivatsa, Anna University, Chennai, India.
(3) Diana Bílková, University of Economics, Prague, Czech Republic.
(4) Dominik Strzałka, Rzeszów University of Technology, Poland.
Complete Peer review History: http://sciencedomain.org/review-history/11984

Original Research Article

Received: 24 July 2015
Accepted: 19 September 2015
Published: 26 October 2015

Abstract

This paper presents a typewritten characters recognition system using Hidden Markov Model (HMM). Character recognition systems convert images of printed, typewritten or handwritten documents into computer readable texts that can be easily edited or searched. Character recognition for typewritten documents is however difficult due to broken edges, touching characters, shape variance, skewing, and heavy printing resulting from the typewriter impact. Three documents (old memo, old war letter and newly typewritten essay) were used to create three datasets of typewritten characters each consisting of 1995, 702 and 2049 characters respectively. The research result showed that, recognition accuracy values are 94.88%, 91.45% and 97.24% for old memo, old war letter and newly typewritten essay datasets respectively. Hence, HMM is an efficient method that can be employed to recognise typewritten documents.

Keywords: Character recognition; Hidden Markov model; Otsu algorithm; accuracy; precision and false positive rate.

**Corresponding author: E-mail: ibrahim.adeyanju@fuoye.edu.ng;*

1 Introduction

Character recognition is commonly defined as the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text. According to Abby [1], character recognition is a technology that enables conversion from different types of documents such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data. Almohri [2] stated that character recognition is one of the most important fields of pattern recognition and has been the centre of attention for researchers in the last forty decades. The goal is to process data that are normally been processed only by humans with computers. One of the apparent advantages of computer processing is dealing with huge amounts of information at high speed. Some other advantages of character recognition are: reading postal address off envelopes, reading customer filled forms, archiving and retrieving text, digitizing libraries etc. Using character recognition, the handwritten and typewritten text could be stored into computers to generate databases of existing texts without using the keyboard [3].

However, typewritten documents are unique among machine-printed documents in the way they are created. Each character is produced independently of the others by pressing a key on the typewriter and ink is mechanically transferred on the paper proportionally to the force of the keystroke. This results in non-uniformity of the intensity of the printed areas. Even within a single word, there can be characters that are faint (lightly pressed) while others are strongly pressed resulting in much darker, blurred and filled-in characters. These problems are worse in carbon copies (of which many exist as primary sources). Another peculiarity lies in the historical nature of these typewritten documents. The majority of office documents and official correspondence of the 20^{th} century are typewritten; a fact that also introduces certain unique challenges such as disintegration of document parts, stains and punch holes, tears and rusts that can considerably reduce their recognition accuracies [4].

Character recognition consists majorly of four phases which are; data acquisition, pre-processing, feature extraction and classifications phases [5]. The output of one step is the input of the next step.

2 Related Work

Omidiora et al. [6] compared machine learning classifiers for recognition of online and offline handwritten digits. The paper compared four machine learning classifiers namely Naive Bayes, Instance Based Learner, Decision Tree and Neural Network for single digit recognition. The experiments were conducted using the WEKA machine learning tool on two datasets; the MNIST offline handwritten digits and a collection of online ISGL handwritten digits acquired with a pen digitiser. Experiments were designed to allow for comparison within the datasets in a cross validation and across them where the online dataset is used for training and the offline dataset for testing and vice versa. Results indicate that the Instance Based Learner classifier performed slightly best with a maximal accuracy of 97.86% followed by the neural network classifier. The decision tree gave the worst performance of the four classifiers. The research investigated the performance of these classifiers in recognition of other characters (alphabets, punctuation and other symbols) and as well as extend the recognition task to other levels of text granularity such as words, sentences and paragraphs. Antonacopoulos [4] proposed a new framework for recognition of heavily degraded characters in historical typewritten documents based on semi-supervised clustering. This paper presents a new semi-supervised clustering framework to the recognition of heavily degraded characters in historical typewritten documents, where off-the-shelf OCR typically fails.

However; Märgner in 2006 did a research on an On-line Handwritten Arabic Word Recognition Using HMM. This is a character based approach without explicit segmentation and achieved recognition accuracy value of 89.77% [7]. Abdul Rahim [8] presented a paper; online handwriting recognition using support vector machine. This research aimed to investigate the usage of support vector machines (SVM) in place of Neural Network in a hybrid SVM/HMM recognition system. The main objective is to further improve the recognition rate by using support vector machine (SVM) at the segment classification level. The main objective is to further improve the recognition rate by using support vector machine (SVM) at the segment

classification level. This was motivated by successful earlier work by Ganapathiraju [9] in a hybrid SVM/HMM speech recognition (SR) system. Rashid in 2011 [10] presented a paper that evaluates Hidden Markov Model (HMM) techniques for OCR of low resolution text–both on screen rendered isolated characters and screen rendered text-lines–and compares it with the performance of other commercial and open source OCR systems. Results show that HMM-based methods reach the performance of other methods on screen rendered text and yield above 98% character level accuracies on both screen rendered text-lines and characters. HMMs combined with statistical grammar rules are very attractive for online cursive handwriting recognition because training is easy and avoids segmentation issues [11]. HMMs have been commonly used for off-line cursive handwriting [12,13] and combined with dictionaries of limited size and cooperative writers it has achieved up to 98% accuracy [14].

3 Research Approach

The research methodology employed in this work is discussed in Sections 3.1 to 3.3.

3.1 Dataset Creation

The HMM based Character Recognition system was experimented on a total number of three (3) sets of typewritten dataset. They are old memo dataset, old war letter dataset and newly typewritten essay dataset and in each dataset 2/3 of the total number of the characters were for training and 1/3 for testing. Table 1 below shows dataset breakdown. The database was divided into two datasets; the training dataset and the testing dataset.

Table 1. Analysis of experimental dataset

Dataset	Number of characters	Number of training dataset	Number of testing dataset	Number of lower case characters	Number of upper case characters	Number of digits
Old memo	1995	1330	665	570	78	17
Old war letter	702	468	234	155	64	15
Newly typewritten essay	2049	1366	683	588	75	20

3.2 Pre-processing Stage and Feature Extraction

Here a global thresholding method; Otsu Algorithm was used to discriminate the bright part of the whole image from the foreground pixels which have lower values.

In Otsu's method, a threshold is needed that minimizes the intra-class variance (the variance within the class), defined as a weighted sum of variances of the two classes [15,16]:

$$\sigma_{\omega}^2(t) = \omega_1(t)\,\sigma_1^2(t) + \omega_2(t)\,\sigma_2^2(t) \quad (1)$$

Where ω_i denotes the probabilities of the two classes separated by a threshold t, and σ_i denotes the variances of these classes.

$$\sigma_b^2(t) = \sigma^2 - \sigma_{\omega}^2(t) = \omega_1(t)\,\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (2)$$

Which is expressed in terms of class probabilities ω_i and class means μi, which in turn can be updated iteratively.

The Otsu Algorithm is as used in this research work is as follows

Step 1: Set up initial ω_i (0) and μ_i (0)

Step 2: Step through all possible thresholds t=1..maximum intensity

i. Update ω_i and μ_i

ii. Compute σ_b^2(t)

Step 3: Desired threshold corresponds to the maximum σ_b^2(t)

Step 4: Compute the two maxima (and two corresponding thresholds). σ_{b1}^2(t) is the greater maximum and σ_{b2}^2(t) is the greater or equal maximum

$$\text{Desired threshold} = \frac{Threshold1+Threshold2}{2} \quad (3)$$

Where

σ_{b2}^2(t)-- The variance of the pixels in the foreground (above threshold)

σ_{b1}^2(t)- The variance of the pixels in the background (below threshold)

3.3 HMM Classification Stage

The typewritten characters were classified using Left-to-Right Hidden Markov Model (HMM).

Before HMMs can be used in actual applications, the development of a Hidden Markov Model approach entails addressing three problems.

a. First, given an observation sequence O and a model λ, how do we compute P(O|λ)? This is known as the evaluation problem.
b. Second, given an observation sequence O and a model λ, what is the optimal state sequence in λ accounting for O (decoding problem).
c. Third, given a set of observation sequences, how do we estimate the parameters of the model (learning problem).

3.3.1 The evaluation problem

Given an observation sequence O =$0_1 \cdots 0_t$, the computation of P(O|λ) is straightforward: $P(O|\lambda) = \sum_S P(O|S,\lambda)P(S|\lambda)$

Where the sum runs over all state sequences S.

As there are N^T possible state sequences, this computation is clearly intractable.

Nonetheless, because of the properties mentioned above, it is easy to see that the paths shared across state sequences S need to be computed only once. Hence, P(O|λ) can be inductively obtained by introducing a forward probability α_t (i): α_t (i) =P(o_t,...,o_t ,s_t = i|λ), which leads to the Forward algorithm below.

Algorithm: Forward procedure

1. Initialization
   $\alpha_1(1) = 1.$ (4)
2. Recursion
   $\alpha_t(j) = [\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}]\, b_j(o_t)$ (5)
3. Termination
   $P(O|\lambda) = \alpha T(N).$

3.3.2 The decoding problem

Sometimes, we are interested, for many reasons, in finding the optimal state sequence accounting for an observation sequence O. In the maximum likelihood sense, this amounts to searching for a state sequence.

$$S^* = \arg\max P(O, S|\lambda). \quad (6)$$

Using the properties of HMMs discussed above, this turns out to be a straightforward task, if we define the partial Viterbi probability in this way:

$$\delta 1(1) = \max_{S1 \ldots St-1} P(O1 \ldots Ot, S1 \ldots St = i|\lambda) \quad (7)$$

This is the probability of the best partial state sequence, generating the t first observations and leading to state i at time t. This leads to the following Viterbi algorithm.

Algorithm: Viterbi decoding

1. Initialization
   $\delta_1(1) = 1,$ (8)
   $B_1(1) = 1$ (9)
2. Recursion
   $\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}] b_j(O_t),$ (10)
   $B_t(j) = \operatorname{argmax}_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}].$ (11)
3. Termination
   $P(S*) = \delta_T(N),$ (12)
   $BT(N) = \operatorname{argmax}_{1 \le i \le N}[\delta_{t-1}(i) a_{iN}],$ (13)
4. Backtracking
   $S_t^* = B_{t+1}(S^*_{t+1}).$ (14)

3.3.3 The learning problem

This problem differs from the two above mentioned problems in the way that only the elemental structure of the HMM is given. Given one or more output sequences, this problem asks for the model parameters M and δ. In other words: The parameters of the HMM have to be trained.

begin
init estimated versions of a_{ij} and b_{jk}, V^T, convergence criterion C, Z:=0
do z:=z+1
compute ˆ a(z) from a(z-1) and b(z-1) by a_{ij}
compute ˆ b(z) from a(z-1) and b(z-1) by b_{jk}, V^T
$a_{ij}(z) :=$ ˆ $a_{ij}(z-1)$
$b_{jk}(z) :=$ ˆ $a_{jk}(z-1)$
until convergence criterium achieved
return $a_{ij} =: a_{ij}(z)$ and $b_{jk} =: b_{jk}(z)$
end

where; $a_{ij} = \frac{\sum_{t=1}^{T} \gamma ij(t)}{\sum_{t=1}^{T} \sum_k \gamma ik(t)}$ $b_{jk} = \frac{\sum_{t=1,\ v(t)=vk}^{T} \sum_t \gamma jl(t)}{\sum_{t=1}^{T} \sum_t \gamma jl(t)}$

4 Experimental Design and Results

4.1 Experimental Setup

The development tool used is Microsoft Visual C# version 4.0 on Windows 8 Ultimate 64-bit operating system, Intel®Core™ i5-3210M CPU @2.50 GHz Central Processing Unit, 4GB Random Access Memory and 500GB hard disk drive. The decision taken to recognize or classify the images as true positive, false positive, false negative and true negative was determined by threshold. Threshold is decided heuristically. Generally, there is no formula for calculating the threshold value. Its value was taken as some factor of maximum value of minimum Euclidean distances of each image from other images. Threshold is the acceptance or rejection of a character template match which is dependent on the match score falling above or below the threshold. The threshold is adjustable within the character recognition system. Results generated based on True Positive (TP), False Negative (FN), False Positive (FP), True Negative (TN), that is, the recognition accuracy was determined by the threshold value set. It was discovered that, high threshold value generates high recognition accuracy and low threshold value generates low recognition accuracy.

4.2 Evaluation Results

4.2.1 Results on old memo dataset

Table 2 shows results generated with Old Memo dataset highlighting False Positive Rate (FPR), recognition precision and recognition accuracy at each threshold values of 0.20, 0.40, 0.60 and 0.80. At 0.2 threshold, recognition accuracy and precision values are 93.68% and 94.39% respectively with False Positive Rate (FPR) value of 5.78% while the accuracy and precision values increased to 93.98% and 94.69% at threshold of 0.4, the FPR decreased to 5.45%. At 0.6 threshold, the accuracy, precision and FPR values are 94.69%, 95.00% and 4.99% respectively however at 0.8 threshold, the system recorded its highest accuracy and precision values of 94.88% and 95.31% respectively while the FPR reduced to 4.68%. Thus the system recorded its best result at 0.8 threshold.

Table 2. Evaluation results generated with old memo dataset

Threshold	FPR (%)	Precision (%)	Accuracy (%)
0.20	5.75	94.39	93.68
0.40	5.45	94.69	93.98
0.60	4.99	95.00	94.69
0.80	4.68	95.31	94.88

4.2.2 Results on new typewritten dataset

Table 3 shows results generated with old war letter dataset highlighting False Positive Rate (FPR), recognition precision and recognition accuracy at each threshold values of 0.20, 0.40, 0.60 and 0.80. At 0.2 threshold, recognition accuracy and precision values are 88.90% and 90.43% respectively with False Positive Rate (FPR) value of 9.57% while the accuracy and precision values increased to 89.32% and 90.87% at threshold of 0.4, the FPR decreased to 9.13%. At 0.6 threshold, the accuracy, precision and FPR values are 90.60%, 91.77% and 8.23% respectively however at 0.8 threshold, the system recorded its highest accuracy and precision values of 91.45% and 92.64% respectively while the FPR reduced to 7.36%. Thus the system recorded its best result at 0.8 threshold.

Table 3. Evaluation results generated with old war letter dataset

Threshold	FPR (%)	Precision (%)	Accuracy (%)
0.20	9.57	90.43	88.90
0.40	9.13	90.87	89.32
0.60	8.23	91.77	90.60
0.80	7.36	92.64	91.45

4.2.3 Results on new typewritten essay dataset

Table 4 shows results generated with old war letter dataset highlighting False Positive Rate (FPR), recognition precision and recognition accuracy at each threshold values of 0.20, 0.40, 0.60 and 0.80. At 0.2 threshold, recognition accuracy and precision values are 96.49% and 97.34% respectively with False Positive Rate (FPR) value of 2.66% while the accuracy and precision values increased to 96.63% and 97.49% at threshold of 0.4, the FPR decreased to 2.51%. At 0.6 threshold, the accuracy, precision and FPR values are 97.07%, 97.79% and 2.21% respectively however at 0.8 threshold, the system recorded its highest accuracy and precision values of 97.24% and 97.94% respectively while the FPR reduced to 2.06%. Thus the system recorded its best result at 0.8 threshold.

Table 4. Evaluation results generated with newly typewritten essay dataset

Threshold	FPR (%)	Precision (%)	Accuracy (%)
0.20	2.66	97.34	96.49
0.40	2.51	97.49	96.63
0.60	2.21	97.79	97.07
0.80	2.06	97.94	97.24

Summarily, recognition accuracy, precision, false positive rate were determined at different threshold values of 0.20, 0.40, 0.60, 0.80 on each of the set of the datasets (old memo dataset, old war letter dataset and new typewritten dataset), as threshold values increases, the accuracy and precision values increases while the false positive rate decreases on the three sets of dataset. However at threshold value of 0.80, result shows that recognition accuracy and precision for old memo dataset are 94.88% and 0.95, for old war letter are 91.45% and 0.93, for newly acquired dataset are 97.24% and 0.98 respectively while FPR for old memo dataset is 0.0468 compared with 0.0736 for old war letter and 0.0206 for new typewritten essay dataset.

Accuracy and Precision results at threshold value of 0.8 produced the best result and it's as summarized in Table 5 below:

Table 5. Evaluation results across the three datasets

Dataset	FPR	Precision	Accuracy (%)
Old Memo	0.0468	0.95	94.88
Old War Letter	0.0736	0.93	91.45
Newly acquired	0.0206	0.98	97.24

Fig. 1 below showed recognition accuracy bar chat with threshold value 0.8

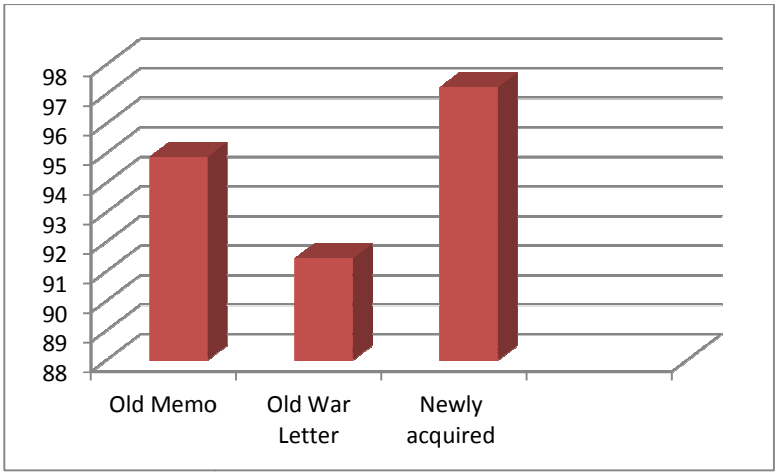

Fig. 1. Accuracy Bar-chat at 0.8 threshold value

5 Conclusion and Future Work

In conclusion, HMM has been used successfully to classify degraded historical typewritten documents with broken edges, touching characters, broken characters, shape variance, skewing and heavy printing. The results also show effects of degradation on accuracy levels of character recognition system with higher recognition accuracy and precision values recorded with new typewritten essay datasets than in old memo and old war letter datasets, thus, this paper has successfully compared recognition in old and new typewritten documents. The results also show that there is better accuracy as threshold levels increases. The developed character recognition system can also be useful for digitally archiving old and historical typewritten documents.

Thus this research work has helped to realise an efficient character recognition system using hidden markov model as classifier.

As a result of the findings during the course of study, it is recommended that there could be further research on word and sentence recognition systems with HMM as classifier and also a further research on HMM classifier for handwritten and computer format characters.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Available:http://www.abbyy.com/finereader/about-ocr/what-is-ocr/

[2] Almohri H, Gray J, Alnajjar H. A real-time DSP-based optical character recognition system for isolated Arabic characters using the TI. Proceedings of the IAJC-IJME International Conference. 2008;201-228.

[3] Vinciarelli A. Offline cursive handwriting: From word to text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2003;26(6):605-622.

[4] Antonacopoulos A, Pletschacher S, Hu J. A new framework for recognition of heavily degraded characters in historical typewritten documents based on semi- supervised clustering. New York, USA, IEEE Computer Society Press. 2009;48-53.

[5] Fenwa OD, Omidiora EO, Fakolujo OA. Development of a feature extraction technique for online character recognition system. Journal of Innovative Systems Design and Engineering. 2012;3(3):10-23.

[6] Omidiora EO, Adeyanju IA, Fenwa OD. Comparison of machine learning classifiers for recognition of online and offline handwritten digits. Computer Engineering and Intelligent Systems. 2013;4(13):39-48. ISSN 2222-1719.

[7] Märgner V, El Abed H, Pechwitz M. On-line handwritten Arabic word recognition using HMM - a character based approach without explicit segmentation. Laurence Likforman-Sulem. SDN06. 2006;259-264.

[8] Abdul Rahim A, Khalia M, Viard-Gaudin C, Yusof R. On-line handwriting recognition using support vector machines and hidden markov models approaches. 2008 IEEE Region 10 Conference TENCON; 2008. ISBN: 0-7803-8560-8.

[9] Ganapathiraju A. Support vector machines for speech recognition. PhD Thesis, Mississippi State University; 2002.

[10] Rashid SF, Shafait F, Breuel TM. An evaluation of HMM-based techniques for the recognition of screen rendered text. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. 2011;1260-1264. ISBN: 978-076-954-520-2.

[11] Makhoul J, Starner T, Schwartz R, Chou G. On-line cursive handwriting recognition using hidden markov models and statistical grammars. Proceedings of Human Language Technology Conference. 2011;56-70.

[12] Arica N, Yarman-Vural F. An overview of character recognition focused on off-line handwriting. IEEE Transactions on Systems, Man and Cybernetics - part C: Applications and Reviews. 2001;47-59.

[13] Casey RG, Lecolinet E. A survey of methods and strategies in character segmentation. In IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006;213-248.

[14] Bunke H, Roth M, Schukat-Talamazzini EG. Off-line cursive handwriting recognition using hidden markov models. Technical Report IAM-94-008, University of Bern. 2005;82-91.

[15] Kumar S, Sharma P. Offline handwritten & typewritten character recognition using template matching. International Journal of Computer Science & Engineering Technology (IJCSET). 2013; 4(06):818-825. ISBN: 2229-3345.

[16] Yasser A. Pre-processing techniques in character recognition, character recognition. Minoru Mori (Ed.); 2010. ISBN: 978-953-307-105-3.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/11984