

Score Normalization for HMM-based Word Spotting Using a Universal Background Model

José A. Rodríguez^{*+}, Florent Perronnin⁺

⁺ Xerox Research Centre Europe

6, Chemin de Maupertuis, 38240 Meylan (France)

^{*} Computer Vision Center (Universitat Autònoma de Barcelona)

Edifici O, Campus Bellaterra, 08193 Bellaterra (Spain)

jrodriguez@cvc.uab.es, Florent.Perronnin@xrce.xerox.com

Abstract

Handwritten word spotting (HWS) is traditionally performed as an image matching task between one or multiple query images and a set of word images in a document. In this article, we address the word spotting problem as a hidden Markov model (HMM) word verification problem and demonstrate the importance of score normalization for improving detection performance. Our main contribution is the introduction of a novel score normalization technique in which the conventional HMM filler model is simplified by using a Gaussian mixture model (GMM). The accuracy of the proposed score normalization is on par with the traditional HMM-based score normalization approaches but it has a lower computational cost. We also identify an interesting special case, the semi-continuous HMM, where the proposed score normalization formalism fits very elegantly and comes at a negligible cost.

Keywords: word spotting, hidden Markov models, score normalization, Gaussian mixture models, handwriting recognition

1. Introduction

Word spotting is the pattern classification task which consists in detecting keywords in document images [10]. This can be formulated as a two-class decision problem: given a word image and a keyword hypothesis, a match is declared if the score of the word image on the keyword model exceeds an application-dependent threshold.

Handwritten word spotting (HWS) has been traditionally approached from a “query-by-example” perspective. A query image is provided to the system and for each candidate image in the document a similarity score between them is computed [14]. Two main classes of approaches have been proposed: holistic techniques such as template matching [19] or local approaches such as DTW[14]. The main challenge in both cases is the definition of a score, i.e. a suitable measure of similarity between word images.

While query-based approaches can achieve acceptable performance in single-writer scenarios, the combination of multiple examples into a statistical model is expected

to increase the retrieval accuracy.

There has been much previous research on modelling handwritten words using statistical models, especially hidden Markov models (HMM), leading to superior performances. Furthermore, this is a common choice for spotting other types of information, such as spoken words [15] or printed text [3]. But, surprisingly, only few recent works in HWS (like [5, 4]) consider this option.

In this work, we adopt this statistical approach to perform word spotting. Each keyword to spot is represented by a HMM, which is used to determine how likely is any word image to correspond to this class. However, using the raw likelihood value $p(X|w)$ outputted by the HMM, while efficient in closed-world hypotheses (e.g. systems using lexicons), is insufficient for a verification task [2]. Instead, a more correct confidence measure is the posterior probability [9]:

$$p(w|X) = \frac{p(X|w)p(w)}{p(X)}. \quad (1)$$

Considering that $p(w)$ can be integrated in the decision threshold and therefore ignored, the posterior probability can be interpreted as a correction of the likelihood $p(X|w)$ with the term $p(X)$ and is thus called score normalized.

However, how to model $p(X)$ is not trivial. One traditional approach for estimating $p(X)$ is the use of filler models. The filler model approach consists in using a model identical to the keyword model but trained with all available samples instead of keyword-specific ones. In a HMM framework, filler models are therefore HMMs.

The main contribution of this article is the introduction of a novel score normalization method that avoids the use of a HMM for modelling $p(X)$ and employs a Gaussian mixture model (GMM) instead. A GMM is a simple particular case of HMM with only 1 state, which in practice means that the ordering of the frames is not taken into consideration. Our experiments demonstrate that the increase in performance obtained by the GMM score normalization is on par with that of the filler model or, in other words, that considering the order has little impact. But thanks to this simplification, the computational cost is significantly reduced both at training and test time. Similar to other

fields, the GMM trained on all samples will be called *universal background model* (UBM).

We also explore the influence of this novel score normalization on two different types of HMMs, namely the continuous density HMM (C-HMM) and the semi-continuous HMM (SC-HMM), and show that the proposed UBM score normalization appears as an elegant and efficient choice to the latter one.

The rest of the article is structured as follows. In section 2 we present an overview of our word spotting system. In section 3 we provide details on the HMM modeling part. In section 4 we review the traditional approaches to score normalization and introduce our proposed approach. Experimental results are provided in section 5. Finally, conclusions are drawn in Section 6.

2. HWS system overview

This study is part of a more general research whose goal is to detect keywords in an incoming flow of scanned letters. One potential application of this is the routing of mails based on the presence of certain keywords. Of course, the outcome of this work can be applied to other problems of similar nature, such as indexing historical documents, document categorization and, more generally, metadata extraction from document images.

Although a detailed description of the complete system pipeline is out of the scope of this work, this is a brief overview of the process:

(1) A segmentation algorithm extracts sub-images that potentially represent words, employing state-of-the-art techniques based on projection profiles and clustering of gap distances.

(2) A simple classifier using holistic features is employed for performing a first rejection pass (*fast rejection*), which prunes out about 90% of the segmented words while falsely rejecting only 5% of the keywords.

(3) The non-pruned word images are normalized with respect to slant, skew and text height, using standard techniques.

(4) For each normalized image, an analytic word descriptor is computed: a window moves from left to right over the image and feature vectors are extracted at each position to build a feature vector sequence.

(5) Using a set of HMMs, a score is assigned to each feature vector sequence. The corresponding image will be flagged as the keyword if the score exceeds a predefined threshold.

This paper focuses on issues involving step (5). Please note that the system is independent of the feature extraction method at (4). Therefore, the validity of our approach is tested on two different state-of-the-art feature sets in handwriting recognition.

3. HMM modelling

In small vocabulary handwritten recognition or in spoken word spotting a common approach is to model each word with a different HMM [7]. This is to be contrasted

with large vocabulary systems where each word is decomposed into sub-word units, typically letters [11] or letters in their right and left context [6], and each sub-word unit is modeled with an HMM. As the number of keywords to spot for a given application is generally small (e.g. a few tens) we followed the former approach. The topology of our HMMs is the classical left-to-right without skip-state jumps.

At training time, we use the Baum-Welch algorithm to estimate the model parameters. At test time, we use the forward-backward (or an approximation using Viterbi) algorithm to compute the probability that the model parameters generated the given observation sequence. A more in-depth description of HMMs, including the mentioned algorithms, can be found in [13].

In the following, we consider two types of HMMs: the continuous HMM (C-HMM) and the semi-continuous HMM (SC-HMM).

3.1 Continuous HMM

In a C-HMM, the probability $p(x|\lambda_{n,s})$ of emitting observation x in the state s of keyword n is modelled as a mixture of $N_{n,s}$ Gaussians:

$$p(x|\lambda_{n,s}) = \sum_{i=1}^{N_{n,s}} w_{n,s,i} p(x|\mu_{n,s,i}, \Sigma_{n,s,i}), \quad (2)$$

where $w_{n,s,i}$ are the mixture weights and $p(\cdot|\mu_{n,s,i}, \Sigma_{n,s,i})$ is a Gaussian with mean vector $\mu_{n,s,i}$ and covariance matrix $\Sigma_{n,s,i}$. In the following, we assume that covariance matrices are diagonal as (i) any distribution can be approximated with an arbitrary precision by a weighted sum of Gaussians with diagonal covariances and (ii) the computational cost of diagonal covariances is much lower than the cost involved by full covariances.

3.2 Semi-continuous HMM

In a SC-HMM, the emission probability of each state is modelled as a mixture of N Gaussians that are shared by all the states of all the keywords [8, 1]:

$$p(x|\lambda_{n,s}) = \sum_{i=1}^N w_{n,s,i} p(x|\mu_i, \Sigma_i). \quad (3)$$

In this case, the only keyword- and state-specific parameters are the mixture weights $w_{n,s,i}$. In the SC case, we also make the diagonal covariance assumption. Usually, the pool of Gaussians must be large enough (several hundreds) to achieve high precision.

The SC-HMM is very attractive from a computational standpoint. Indeed, the cost of training/testing HMMs is generally dominated by Gaussian computations. Having the same shared set of Gaussians across keywords and states can lead to a substantial saving in computation. How much is saved will depend on several factors, the most important one being the number of keywords

that need to be spotted. On the other hand, the accuracy of the SC-HMM is generally lower than the accuracy of the C-HMM. This is not surprising as in the SC-HMM case, emission probabilities are constrained by the pool of shared Gaussians. For our problem, we found out experimentally that the performance of the SC-HMM was very significantly degraded compared to the C-HMM. However, this difference is dramatically reduced when applying the proposed UBM score normalization.

4. Score normalization

As mentioned in the introduction, for a verification problem such as HWS, the raw likelihood $p(X|w)$ is not a robust confidence measure itself, and a correction term $p(X)$ is necessary to estimate the more correct posterior $p(w|X)$. This strategy is called score normalization and is a well studied topic in speech/speaker recognition applications [9, 15]. More recently, several works in the field of handwriting recognition have used score normalization for writer verification [17] or rejection strategies [2].

4.1 Score normalization approaches

In the speech and handwriting recognition literature, two main score normalization approaches can be identified. The first score normalization approach consists in splitting the evidence $p(X)$ into:

$$p(X) = p(w)p(X|w) + p(\bar{w})p(X|\bar{w}) \quad (4)$$

where $p(\bar{w}) = 1 - p(w)$ and $p(X|\bar{w})$ is an anti-model of word w . The posterior then becomes

$$\begin{aligned} p(w|X) &= \frac{p(w)p(X|w)}{p(w)p(X|w) + p(\bar{w})p(X|\bar{w})} \quad (5) \\ &= \frac{1}{1 + \frac{p(\bar{w})p(X|\bar{w})}{p(w)p(X|w)}}. \quad (6) \end{aligned}$$

and the following likelihood ratio can be used for scoring:

$$\frac{p(X|w)}{p(X|\bar{w})}. \quad (7)$$

The most widely used approach to computing $p(X|\bar{w})$ is the so-called cohort models [16]. X is scored against a set of alternative models, the cohorts, and the scores are then combined by taking the average or the maximum, for instance. However, in a small vocabulary word spotting application there is generally no available set of cohorts and this approach cannot be applied.

The second approach, which is more suited to the word spotting problem, consists in modeling directly the distribution $p(X)$ of all the words that might be encountered by the system [15]. This model is generally referred to as *filler model*, garbage model, world model or background model. It is trained on a sufficiently large set of representative sample images. The traditional approach is to use for the filler model the same topology as for the word model. This means that if the word model is an HMM with e.g. 100 states, then the filler model will be an HMM with 100 states.

4.2 Proposed score normalization

In this work, we propose a novel score normalization approach by choosing the background model to have a different structure from the keyword model. More precisely, we suggest the use of a GMM instead of a filler model. Words are scored using the following likelihood ratio:

$$\frac{p(X|\lambda^{HMM})}{p(X|\lambda^{GMM})}, \quad (8)$$

where λ^{GMM} are the set of parameters that define the GMM. Despite the fact that a GMM does not take into account the ordering of the frames, contrarily to an HMM, experimental results show that the proposed approach is on par in terms of detection accuracy with respect to the filler model based normalization. But more importantly, thanks to its simpler structure, it has a significantly lower computational cost at training and test time. Also, there is only one GMM model for score normalization for all keywords. In contrast, in the case of a filler model, if the different keywords have different topologies (e.g. different number of states), also different filler models must be used, which increases even more the difference in computational cost.

In the following, we distinguish the C-HMM and SC-HMM cases. In the C-HMM case, the GMM background model is trained on a large set of samples independently from the keyword HMMs. In the SC-HMM case, we take advantage of the already existing pool of Gaussians on which the SC-HMMs are built. Indeed, the same pool of Gaussians is used for score normalization. This choice is attractive from a computational standpoint since the additional cost of score normalization is negligible. Moreover, this choice ensures a maximum reduction of a mismatch between the training and test conditions. Indeed, as the computation of $p(X|w)$ and $p(X)$ is based on the same pool of Gaussians, a mismatch should have a similar effect on both terms.

4.3 Interpretation of score normalization

The need for score normalization can be understood using a comprehensive example. Let us imagine the image of a word written with a writing style that does not appear in the training set. Even if the image corresponds to a keyword, the log-likelihood $p(X|w)$ might be low because of the “unseen” frames. In contrast, if score normalization is used, the frames will also score low on the filler or background model $p(X)$, thus keeping the ratio $p(X|w)/p(X)$ approximately constant. The same effect is found in the opposite case: when a word contains frames that are very frequent in the “universe”, the log-likelihood $p(X|w)$ will be high, but in this case $p(X)$ is also large and score normalization compensates. In general, score normalization compensates for mismatches between training and test conditions and words with frequent patterns (i.e. frequent letters).

In Section 5.4 we provide further discussion on these issues based on the obtained results.

5. Experiments

In this section, the experiments carried out to evaluate the proposed score normalization are reported. First, we proceed to describe the experimental conditions and then results for C-HMM and SC-HMM will follow.

5.1. Experimental setup

The experiments are carried out on a dataset containing 630 real scanned letters (written in French) submitted to the customer department of a company. As mentioned in the introductory section, these data are unconstrained and therefore the letters may contain different writing styles, artifacts, spelling mistakes and other kinds of noise. Word hypotheses are segmented from the page images (see Section 2) and the sub-images corresponding to certain keywords are manually labelled. Only the 10 most frequent word classes (e.g. “Monsieur”, “Madame”, “résilier”, “contrat”, etc.) are used in the experiments. From 208 to 750 positive examples are available for each keyword.

On average, after the fast rejection stage about 25 hypotheses per keyword remain in each document. For this more difficult set of hypotheses we use the explained HMMs.

As mentioned above, for assessing the validity of the proposed approach in a general way, we use two well-known state-of-the-art features, namely the ones proposed by

- Marti & Bunke [11]: this feature set consists in computing 9 geometrical values in each image column, such as position of upper and lower pixel, mean position, second order moment, etc.).
- Vinciarelli et al. [18]: a window slides on the image and, at each position, the window is split into regular cells and the number of pixels in each cell is counted.

A left-to-right HMM is built for each keyword using 10 states per letter. Training and testing is carried out using 5-fold cross-validation, ensuring that the different writers are not mixed among the folds. The GMM is trained on a random subset of all images containing approximately 10^6 feature vectors.

The performance of each keyword detector is evaluated by inspecting the DET curves [12]. These are trade-off curves plotting false acceptance (FA) versus false rejection (FR) rates. Let θ_n be the threshold for keyword n , meaning that all samples with scores $S > \theta_n$ are retrieved. Then, $FR(\theta_n)$ is defined as the percentage of samples with label w_n with $S < \theta_n$. Similarly, $FA(\theta_n)$ is defined as the number of samples not labelled with w_n with $S > \theta_n$. For summarizing a curve with a single value, we employ the common average precision (AP) figure. Some results show the mean of the AP across all keywords, which will be called mean AP or, shortly, mAP.

5.2. Results for continuous HMM

In the first round of experiments we compare the performance of three settings: 1) without score normalization 2) with a filler model, 3) with the proposed GMM-based score normalization.

For the system without score normalization, experiments show that training the HMMs with more than 1 Gaussian per mixture does not give a significant performance improvement. This is surprising when comparing with the results found in the literature, which use a higher number of Gaussians, and also given the high number of training samples. Therefore, we keep 1 Gaussian per mixture for the sake of computational efficiency. When using a filler model, the best performance is obtained when both the keyword HMM and the filler HMM have 8 Gaussians per mixture. Finally, for the proposed system, the best performance is obtained for an HMM with 8 Gaussians per mixture and the normalizing GMM of 64 Gaussians per mixture.

Using these configurations, the following results are obtained. Table 1 compares the mAP values (i) for the proposed score normalization approach (GMM), (ii) for the HMM without score normalization and (iii) for the filler model. Fig. 1 shows the corresponding DET curves for the exemplary keyword *résiliation* (translated as “cancellation”) for the Marti & Bunke features.

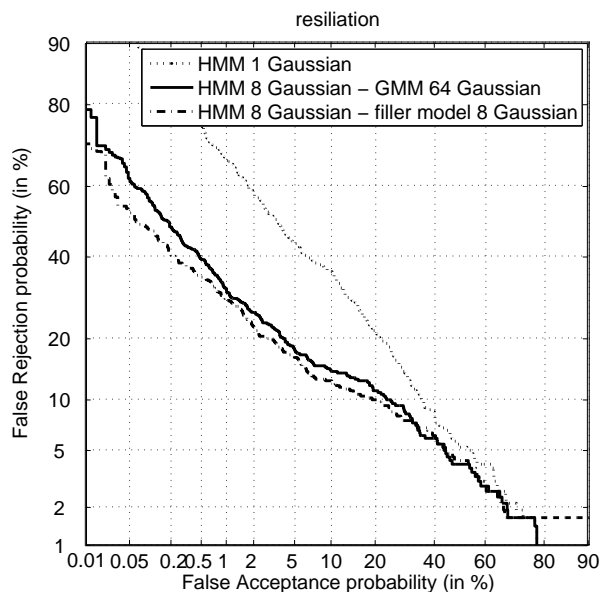


Figure 1. DET curves comparing the proposed GMM score normalization to the filler model based normalization and the system without score normalization, for Marti & Bunke features, for the C-HMM case.

In Fig. 1 and Table 1 we can observe that there is a significant reduction in terms of FA and FR when score normalization is considered. Also, it can be observed that the filler model approach and the proposed GMM approach

Table 1. Mean average precision (mAP) values of the proposed GMM score normalization, of the filler model based normalization and of the system without score normalization, for the different feature types using C-HMMs.

Score norm.	Marti & Bunke	Vinciarelli et. al
None	0.329	0.336
GMM	0.696	0.782
Filler	0.705	0.776

Table 2. Mean average precision (mAP) values of the proposed GMM score normalization, of the filler model based normalization and of the system without score normalization, for the different feature types using SC-HMM.

Score norm	Marti & Bunke	Vinciarelli et. al
None	0.146	0.123
GMM	0.754	0.762

are on par in terms of detection accuracy.

However, the training process of the GMM with 64 Gaussians takes 2.5 less time than training a filler model with 8 Gaussians. And at test time, the scoring on the GMM is 4.6 times faster than on the filler model on average. On top of that, if keyword dependent filler models are employed -like in our case-, their computational time must still be multiplied by the number of different filler models in the system. Thus it is clear that the proposed GMM-based normalization provides an advantage in terms of computational cost with respect to the traditional filler model approach.

5.3. Results for semi-continuous HMM

The second round of experiments deals with the SC-HMM. For a system without score normalization, the best results were obtained with a pool of 1024 Gaussians. However, since the performance is not significantly better than in the case of 512 Gaussians, and the computational time is approximately twice as high, we keep a value of 512 Gaussians.

In our experiments, we found that the SC-HMM by itself performed very poorly for our particular problem (probably due to the high variability of the data and the high number of parameters to tune, 512 times the number of states). However, when applying the score normalization based on the same pool of Gaussians underlying the SC-HMM, the performance dramatically increases. This is appreciated from the corresponding mAP values reported in Table 2. In the SC-HMM case, the filler model comparison was discarded because the huge computational cost makes it an impractical alternative.

Again, the DET curves for the keyword *résiliation* are provided as an example in Fig. 2.

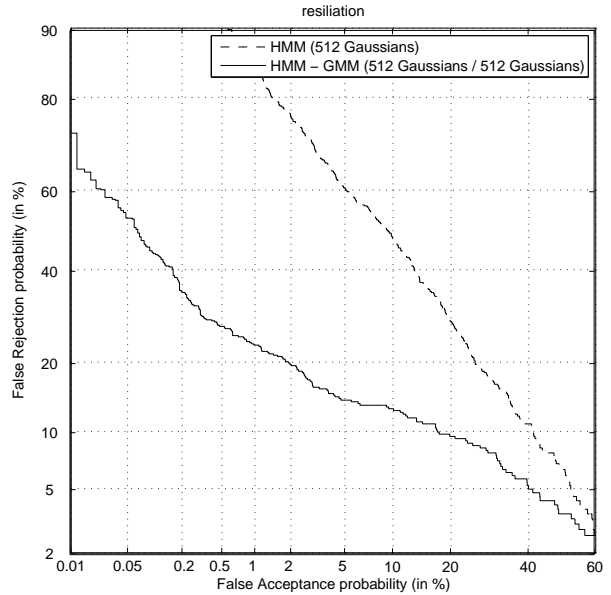


Figure 2. DET curves comparing the SC-HMM with and without score normalization, for the Marti & Bunke features.

Note that there is no practical reason that prevents us from using a different GMM for the SC-HMM and the background model. But interestingly, experiments show that the best results are obtained always when both are identical, as Tables 3 and 4 and show. This seems to validate our intuition that the mismatches of the HMM are compensated by the mismatches of the GMM when exactly the same Gaussian pool is used in both cases, while with different pools these mismatches are not as efficiently compensated.

5.4. Evidence for confusable words

Results have shown that score normalization works well in practice. We have also hypothesized that score normalization can help in the case of words containing frames that are frequently found in the “universe”.

Without a better approximation, we assume that the words with more frequent letters will be the words with more frequent patterns, and therefore the “most confusable” ones. Therefore we define the confusability of a word as the average frequency of its constituting letters in the considered language. Although we do not have enough words to carry out an in-depth analysis, it is interesting to observe that some of the word which enjoyed the largest increase in terms of mAP present also high confusability values (the average frequency of its letters is high).

6. Conclusions

In the framework of word spotting using hidden Markov models, we have shown that score normalization is necessary, and we have presented a novel approach for score normalization that uses a GMM for modelling the

Table 3. mAP values for the different combinations of HMM and GMM, for the Marti & Bunke features using SC-HMM.

HMM / GMM	128 G	256 G	512 G	1024 G
128 G	0.724	0.717	0.702	0.691
256 G	0.679	0.743	0.733	0.726
512 G	0.666	0.728	0.754	0.749
1024 G	0.670	0.723	0.748	0.759

Table 4. mAP values for the different combinations of HMM and GMM, for the Vinciarelli et al. features

HMM / GMM	128 G	256 G	512 G	1024 G
128 G	0.702	0.611	0.534	0.470
256 G	0.656	0.736	0.690	0.645
512 G	0.636	0.715	0.762	0.739
1024 G	0.633	0.703	0.747	0.775

background. This simpler choice performs comparably to the usually chosen filler models while it runs much faster both at training and test time. Additionally, we have shown that this score normalization is a natural and elegant choice for the SC-HMM that can drastically improve its performance at a very low additional cost.

Several reasons why this score normalization works in practice can be identified in terms of mismatches between training and test conditions and of frequent patterns. We provided results that validate the intuition that score normalization compensates for these two problems. Since the results were qualitative for the second case, a future task is to carry out a more in-depth study using more words to establish the relationship between confusability and performance increase.

Acknowledgements

The work of José A. Rodríguez is partially supported by the Spanish projects TIN2006-15694-C02-02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018). We would like to thank J. Lladós and G. Sánchez for their support.

References

- [1] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:2033–2045, 1990.
- [2] A. Brakensiek, J. Rottland and G. Rigoll, "Confidence measures for an address reading system", *7th Int. Conf. on Document Analysis and Recognition*, 2003, pp 294.
- [3] F. R. Chen, L. D. Wilcox and D. S. Bloomberg, "Word spotting in scanned images using hidden Markov models", *IEEE Conf. on Audio, Speech and Signal Processing*, 1993, volume 5, pp 1–4.
- [4] C. Choisy, "Dynamic handwritten keyword spotting based on the NSHP-HMM", *9th Int. Conf. on Document Analysis and Recognition*, 2007, pp 242–246.
- [5] J. Edwards, Y. W. Teh, D. A. Forsyth, R. Bock, M. Maire and G. Vesom, "Making Latin Manuscripts Searchable using gHMMs", *NIPS*, 2004.
- [6] G. A. Fink and T. Plötz, "On the Use of Context-Dependent Modelling Units for HMM-Based Offline Handwriting Recognition", *9th Int. Conf. on Document Analysis and Recognition*, 2007, pp 729–733.
- [7] D. Guillevic and C. Y. Suen, "HMM word recognition engine", *4th Int. Conf. on Document Analysis and Recognition*, 1997, pp 544–547.
- [8] X. D. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals", *Readings in speech recognition*, 1990, pp 340–346, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [9] H. Jiang, "Confidence measures for speech recognition: A survey.", *Speech Communication*, 45(4):455–470, 2005.
- [10] R. Manmatha, C. Han and E. M. Riseman, "Word Spotting: A New Approach to Indexing Handwriting", *IEEE Conf. on Computer Vision and Pattern Recognition*, 1996, pp 631.
- [11] U.-V. Marti and H. Bunke, "Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system", *Int. J. of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.
- [12] A. Martin, G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, "The DET curve in assessment of detection task performance", *Proc. of EuroSpeech'97*, 1997, pp 1895–1898.
- [13] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. of the IEEE*, 77:257–286, 1989.
- [14] T. M. Rath and R. Manmatha, "Word Image Matching Using Dynamic Time Warping.", *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, pp 521–527.
- [15] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system", *Int. Conf. on Acoustics, Speech, and Signal Processing*, 1990, pp 129–132.
- [16] A. Rosenberg, J. DeLong, C. Lee, B. Juang and F. Soong, "The use of cohort normalized scores for speaker verification", *Proc. Int. Conf. on Spoken Language Processing*, 1992, volume 1, pp 599–602.
- [17] A. Schlapbach and H. Bunke, "Off-Line Writer Verification: A Comparison of a Hidden Markov Model (HMM) and a Gaussian Mixture Model (GMM) Based System", *Proc. of the 10th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR'06)*, 2006.
- [18] A. Vinciarelli, S. Bengio and H. Bunke, "Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):709–720, 2004.
- [19] B. Zhang, S. N. Srihari and C. Huang, "Word image retrieval using binary features", *Proc. of the SPIE Conf. on Document Recognition and Retrieval XI*, 2004.