Probabilistic Interpretation and Improvements to the HMM-Filler for Handwritten Keyword Spotting

Joan Puigcerver
PRHLT Research Center
Universitat Politècnica de València
València - Spain
Email: joapuipe@prhlt.upv.es

Alejandro H. Toselli
PRHLT Research Center
Universitat Politècnica de València
València - Spain
Email: ahector@prhlt.upv.es

Enrique Vidal
PRHLT Research Center
Universitat Politècnica de València
València - Spain
Email: evidal@prhlt.upv.es

Abstract—**Traditionally, the HMM-Filler approach has been widely used in the fields of speech recognition and handwritten text recognition to tackle lexicon-free, query-by-string keyword spotting (KWS). It computes a score to determine whether a given keyword is written in a certain image region. It is conjectured, that this score is related to the confidence of the system, respect to the previous question. However, it is still not clear what this relationship is.**

In this paper, the HMM-Filler score is derived from a probabilistic formulation of KWS, which gives a better understanding of its behavior and limits. Additionally, the same probabilistic framework is used to present a new algorithm to compute the KWS scores, which results in better average precision (AP), for a keyword spotting task in the widely used IAM database. We show that the new algorithm can improve the HMM-filler results up to 10.4% relative (5.3% absolute) points in AP, in the considered task.

I. Introduction

In the recent years, incredibly large quantities of handwritten documents have been, and are being, scanned into digital images and being stored by digital libraries. Despite this enormous effort for preservation and aid accessibility, most of the information contained in these documents is largely inaccessible, since the transcription of these documents is not usually available, and even more, it would be extremely expensive to accurately transcribe all of them. Thus, automated methods are needed to search accurately and efficiently for information on such collections.

Keyword Spotting (KWS) was introduced in the Speech Processing field many years ago for similar purposes, and has been recently adopted for processing handwritten documents. Traditionally, two scenarios have been proposed for KWS, depending on the type of the user query: query-by-example, where the query is an exemplar image, on query-by-string, where it is a simple string. The fundamental question that both approaches try to answer is: given a keyword and a image region, is the keyword written in the image?

One of the most popular methods for query-by-string KWS on handwritten documents, is the one known as "HMM-Filler" [1], [2]. This approach rests on the idea of using a *filler* (or "garbage") model and, a *word-specific* model for each query keyword, all of them built using character *hidden Markov models* (HMMs). Typically, HMM-Filler is used for *line-oriented* KWS (the image regions are lines of document pages). The HMM-Filler gives an score to help to determine, without any kind of segmentation into words or characters, whether the given keyword appears in the line image.

Other methods have been presented in the literature for similar purposes [3], but one of the attractive features of the HMM-Filler approach is that it is "lexicon-agnostic"; that is, it does not require any predefined word lexicon. The downsides of this method is that it requires the search to be done on-the-fly, which entails much higher search times, compared to index-based methods [3]. Additionally, the lack of context information of the original HMM-Filler model typically brings a lower performance than lexicon-based models. However, these two problems have been addressed in the literature to improve the performance of the original HMM-Filler model [4], [5], [6].

Surprisingly, and despite the huge popularity and works related to the HMM-Filler model, the theory and probabilistic interpretation of the scores computed by this, have not been studied yet in the literature. And thus, it is not clear if superior performance can be achieved under the same constraints under which the HMM-Filler model is applied.

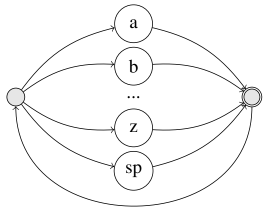
In the present work, we give a probabilistic interpretation of KWS, from which the scores of the HMM-Filler model can be interpreted probabilistically. Furthermore, we show that the performance achieved by the traditional HMM-Filler can be substantially improved, thanks to a more advanced algorithm, which essentially entails the same algorithmic complexity.

The rest of the paper is organized as follows: Sect. II outlines the traditional HMM-Filler approach to KWS. Sect. III introduces the probabilistic framework used to tackle the problem presented by KWS. The traditional HMM-Filler is derived from this framework, and a new algorithm is proposed, in this section. Sect. IV describes the experiments performed to compare both algorithms and analyzes the results obtained. Finally, the conclusions of this work are drawn in Sect. V.

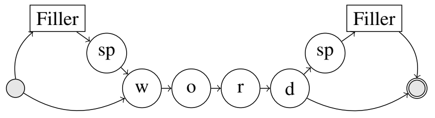
II. HMM-Filler approach to KWS

Traditionally, the HMM-Filler approach [1] has been widely used in the KWS community. This approach typically uses character HMMs with diagonal Gaussian Mixture Models as the emitting probability density function in the states.

From these character HMMs, two composite models are built: a generic "filler HMM" or garbage model, and a specific "keyword HMM" (both depicted in Fig. 1). The filler model is able to process any arbitrary sequence of characters, while the keyword model is constrained to sequences that contain the keyword $\mathbf{v}$.



(a) Character Filler HMM : f



(b) Keyword HMM : k_v

Fig. 1: (a) "Filler HMM", f, and (b) "keyword HMM", k_v, built for the keyword $\mathbf{v}$ = "word".

Then, a spotting confidence score $S(\mathbf{v},\mathbf{x})$ is computed for each keyword $\mathbf{v}$ and image region $\mathbf{x}$. Informally, the higher the score is, the more certain is the system that the given keyword is written in the image region.

$$S(\mathbf{v},\mathbf{x}) = \frac{\max\limits_{\mathbf{w}\in\Sigma^*} \log p_{k_v}(\mathbf{w},\mathbf{x}) - \max\limits_{\mathbf{w}'\in\Sigma^*} \log p_f(\mathbf{w}',\mathbf{x})}{|\,\mathbf{v}\,|} \tag{1}$$

The denominator of the score is introduced by convenience, in order to normalize the scores of long keywords. Observe that the likelihood of a sequence decays exponentially with its length. Typically $|\,\mathbf{v}\,|$ is the length of the keyword in number of frames, given by the implicit HMM-state segmentation. However, in this work we use the average length in frames of the keyword $\mathbf{v}$, given by multiplying the number of characters in $\mathbf{v}$ by the average number of frames consumed by each character, when recognizing the validation set.

In the following section, a general probabilistic framework for KWS is introduced, which is used to derive a probabilistic interpretation of the score $S(\mathbf{v},\mathbf{x})$.

III. Probabilistic framework for KWS

In order to tackle the question addressed in the introduction section from a probabilistic point of view, we define a binary random variable $\mathcal{R}$, which denotes whether the image $\mathbf{x}$ is relevant for the query keyword $\mathbf{v}$. We consider an image to be "relevant" if $\mathbf{v}$ is written in it.

Then, the probability distribution $P(\mathcal{R} \mid \mathcal{Q},\mathcal{X})$ can be used to answer the question presented by KWS, where $\mathcal{Q}$ is a random variable over all possible query keywords and $\mathcal{X}$ is defined over all possible images. Observe that, the probability that we seek to compute is $P(\mathcal{R}=1 \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x})$.

As we indicated before, we consider an image to be relevant if the given keyword is written in it. Thus, the probability $P(\mathcal{R}=1 \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x})$ is intuitively related to the distribution of the different possible transcriptions of the image region, i.e. $P(\mathcal{W} \mid \mathcal{X})$, where $\mathcal{W}$ is a random variable over all possible sets of transcriptions.

Following this idea, the distribution $P(\mathcal{R} \mid \mathcal{Q},\mathcal{X})$ is marginalized over all possible transcriptions $\mathcal{W}$ of $\mathcal{X}$. Then, taking into account that all relevant transcriptions must include the keyword $\mathbf{v}$, we obtain:

$$\begin{aligned} P(\mathcal{R}=1 \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x}) &= \\ \sum_{\mathbf{w}'\in\Sigma^*} P(\mathcal{R}=1,\mathcal{W}=\mathbf{w}' \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x}) &= \\ \sum_{\mathbf{w}\in\Sigma^*\,\mathbf{v}\,\Sigma^*} P(\mathcal{W}=\mathbf{w} \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x}) & \end{aligned} \tag{2}$$

Since $\mathbf{w}$ contains $\mathbf{v}$, it is reasonable to assume that $P(\mathcal{W}=\mathbf{w} \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x})$ is independent of $\mathbf{v}$, resulting in:

$$P(\mathcal{R}=1 \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x}) \approx \sum_{\mathbf{w}\in\Sigma^*\,\mathbf{v}\,\Sigma^*} P(\mathcal{W}=\mathbf{w} \mid \mathcal{X}=\mathbf{x}) \tag{3}$$

Applying Bayes rule to $P(\mathcal{W}=\mathbf{w} \mid \mathcal{X}=\mathbf{x})$, in the previous equation, gives:

$$P(\mathcal{R}=1 \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x}) = \sum_{\mathbf{w}\in\Sigma^*\,\mathbf{v}\,\Sigma^*} \frac{p(\mathcal{W}=\mathbf{w},\mathcal{X}=\mathbf{x})}{p(\mathcal{X}=\mathbf{x})} \tag{4}$$

Observe that $p(\mathcal{X}=\mathbf{x})$ does not depend upon the keyword, thus it can be extracted from the sum. Moreover, it can be marginalized over all possible transcriptions $\mathbf{w}'$ of the image region, leaving:

$$P(\mathcal{R}=1 \mid \mathcal{Q}=\mathbf{v},\mathcal{X}=\mathbf{x}) = \frac{\sum\limits_{\mathbf{w}\in\Sigma^*\,\mathbf{v}\,\Sigma^*} p(\mathcal{W}=\mathbf{w},\mathcal{X}=\mathbf{x})}{\sum\limits_{\mathbf{w}'\in\Sigma^*} p(\mathcal{W}=\mathbf{w}',\mathcal{X}=\mathbf{x})} \tag{5}$$

From now on, for sake of clarity, we will omit the random variable names, except for $\mathcal{R}=1$, which we will write simply as $\mathcal{R}$. Thus, the previous equation is simply rewritten as:

$$P(\mathcal{R} \mid \mathbf{v},\mathbf{x}) = \frac{\sum\limits_{\mathbf{w}\in\Sigma^*\,\mathbf{v}\,\Sigma^*} p(\mathbf{w},\mathbf{x})}{\sum\limits_{\mathbf{w}'\in\Sigma^*} p(\mathbf{w}',\mathbf{x})} \tag{6}$$

In summary, the probability of an image $\mathbf{x}$ being relevant for a keyword $\mathbf{v}$, $P(\mathcal{R} \mid \mathbf{v},\mathbf{x})$, is equal to the sum of the posteriors of all transcriptions $\mathbf{w}$ of line image $\mathbf{x}$, which contain the keyword $\mathbf{v}$ (see Eq. 3). And this can be decomposed into a fraction where the numerator is the sum of the likelihoods of all transcriptions containing the keyword $\mathbf{v}$, and the denominator is the sum of the likelihood of all possible transcriptions, with no restrictions (see Eq. 6).

As a final note, observe that the problem of the exponential decay of the likelihood $p(\mathbf{x},\mathbf{w})$ with the length of $\mathbf{x}$ and $\mathbf{w}$ will also affect Eq. 6 as it affected the HMM-Filler scores, since it is intrinsic in the modeling of $p(\mathbf{x},\mathbf{w})$. Thus, the heuristic used in Eq. 1 is also used here during the experimental phase, giving:

$$\log P'(\mathcal{R} \mid \mathbf{v},\mathbf{x}) = \frac{\log P(\mathcal{R} \mid \mathbf{v},\mathbf{x})}{|\,\mathbf{v}\,|} \tag{7}$$

A. Probabilistic interpretation of HMM-Filler

The probabilistic interpretation of the HMM-Filler approach, introduced in section II, comes after applying a Viterbi approximation to both the numerator and the denominator of Eq. 6. That is, the sum is substituted by the maximum:

$$P(\mathcal{R} \mid \mathbf{v}, \mathbf{x}) \approx \frac{\max\limits_{\mathbf{w} \in \Sigma^* \, \mathbf{v} \, \Sigma^*} p(\mathbf{w}, \mathbf{x})}{\max\limits_{\mathbf{w}' \in \Sigma^*} p(\mathbf{w}', \mathbf{x})} \tag{8}$$

By simply taking the logarithm of the previous expression and applying the same heuristic than in Sect. II, a very similar expression to Eq. 1 is obtained.

$$\log P(\mathcal{R} \mid \mathbf{v}, \mathbf{x}) \approx \frac{\max\limits_{\mathbf{w} \in \Sigma^* \, \mathbf{v} \, \Sigma^*} \log p(\mathbf{w}, \mathbf{x}) - \max\limits_{\mathbf{w}' \in \Sigma^*} \log p(\mathbf{w}', \mathbf{x})}{|\,\mathbf{v}\,|} \tag{9}$$

There is a subtle difference between Eq. 1 and Eq. 9: observe that the former used two probability distributions p_f and p_{k_v}, while the latter only uses p. However, observe that all paths in the model k_v are found in f. Plus, in the classical filler no weights are used in the f and k_v automatons. Thus, $\max_{\mathbf{w} \in \Sigma^* \, \mathbf{v} \, \Sigma^*} \log p_f(\mathbf{w}, \mathbf{x}) = \max_{\mathbf{w} \in \Sigma^*} \log p_{k_v}(\mathbf{w}, \mathbf{x})$, which used together with Eq. 9 and defining $p = p_f$ results in the score defined in Eq. 1.

$$\begin{aligned} &\log P(\mathcal{R} \mid \mathbf{v}, \mathbf{x}) \approx \\ &\frac{\max\limits_{\mathbf{w} \in \Sigma^* \, \mathbf{v} \, \Sigma^*} \log p_f(\mathbf{w}, \mathbf{x}) - \max\limits_{\mathbf{w}' \in \Sigma^*} \log p_f(\mathbf{w}', \mathbf{x})}{|\,\mathbf{v}\,|} = \\ &\frac{\max\limits_{\mathbf{w} \in \Sigma^*} \log p_{k_v}(\mathbf{w}, \mathbf{x}) - \max\limits_{\mathbf{w}' \in \Sigma^*} \log p_f(\mathbf{w}', \mathbf{x})}{|\,\mathbf{v}\,|} = \\ &S(\mathbf{v}, \mathbf{x}) \end{aligned} \tag{10}$$

B. Exact solution

The denominator of the fraction in Eq. 6 is the likelihood of the image region. Thus, can be obtained by both the *forward* and *backward* algorithms, which have been widely studied in the literature. Furthermore, this factor does not depend on the query keyword $\mathbf{v}$, and thus can be precomputed to save computational efforts during query time.

With respect to the numerator, observe that the "keyword HMM" model introduced in Sect. II can be used to compute this value. More precisely, the *forward* algorithm can be used again to obtain the total likelihood of all possible transcriptions which include the query keyword $\mathbf{v}$. We implemented this *forward* algorithm to work directly over lattices, in order to speed-up the computations, as explained in the following subsection.

C. Improving computational cost

As explained in [7], the computation time of the likelihood given by the specific "keyword HMM" is too expensive to be done on-the-fly for each query keyword, in realistic scenarios. A much faster alternative is to approximate that likelihood by means of a character-lattice (CL), generated using the generic "filler HMM".

Following the same motivation, in our experiments, we approximated the solution to Eq. 6 and Eq. 8 by means of the CL of the image $\mathbf{x}$.

Observe that the denominator of Eq. 6 can be easily computed as the *forward* likelihood in the final node of the CL. In order to compute the numerator from the CL, one must sum only the likelihood of those paths containing the keyword $\mathbf{v}$.

First, an algorithm based on the well-known Knuth–Morris–Pratt algorithm [8] for string matching, is used to build a DFA that accepts all the sequence of characters in Σ^* that contain the keyword $\mathbf{v}$, that is the regular language $\Sigma^* \, \mathbf{v} \, \Sigma^*$. Fig. 2 shows this automaton for the exemplar keyword "word".

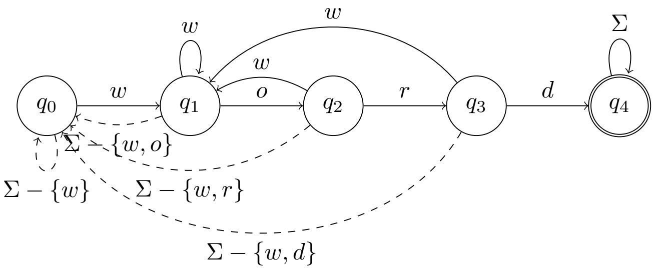


Fig. 2: Complete DFA accepting all strings containing the sequence of characters "word".

Then, one can simply intersect the previous DFA with the image CL, obtaining a weighted DAG which encodes all the possible paths from the CL, containing the keyword $\mathbf{v}$. The *forward* likelihood in the final state of the intersection graph is equal to the sum in the numerator of Eq. 6.

Moreover, the step of intersection and the *forward* computation can be directly addressed using an embedded dynamic-programming algorithm which avoids creating the keyword DFA and intersecting the graphs explicitly, and has the same asymptotic running time, $O((|V| + |E|) \times |\,\mathbf{v}\,|)$, where V and E are the set of nodes and edges of the graph, respectively.

D. Context modeling improvement

Observe that Eq. 6 and Eq. 8 do not depend on any concrete model for the distribution $p(\mathbf{w}, \mathbf{x})$. Thus, more advanced models rather than the HMM-Filler model can be employed to model it, achieving further improvements in the performance of the KWS system.

More precisely, Fischer et al. [9] suggested to use character LM to improve the performance of HMM-based KWS. The main idea is that a prior distribution over the character sequence can be used (i.e. a n-gram LM), instead of using the automaton "Filler HMM". Following that work, Toselli et al. [5] used high order character n-grams to boost the performance of CL-based KWS systems.

In our experiments, we follow the ideas of [9], [5], in order to show how the two approximations to $P(\mathcal{R} \mid \mathbf{v}, \mathbf{x})$, described earlier, are affected by the increasing size of the character n-gram models.

IV. Experiments

A. Corpora

The IAM dataset [10] is used during experimentation to validate the theoretical improvements introduced in Sect. III. The IAM dataset consists of English handwritten texts from many writers. The same training, validation and test partitions have been used in previous KWS works [9], [4]. The line segmentation provided by the original corpus is used in this work. Additionally, the LOB text corpus [11] was used to build a 20K-word lexicon and train the language model (test lines were excluded from LOB). Tab. I shows the basic statistics of the corpus.

TABLE I: Basic statistics of the IAM database and its partitions.

	Training	Validation	Test	Total
Running chars	269 270	39 318	39 130	347 718
Running words	47 615	7 291	7 197	62 103
Lines	6 161	920	929	8 010
Char set size	72	69	65	81
Lexicon size	7 778	2 442	2 488	9 809

B. Set of Keywords to be Spotted

This work uses the same set of queries as in [4], [12], [3]. It contains all words of the training partition, excluding stop words and punctuation symbols. Among the 3 291 keywords to be spotted (M), only 1 098 are *relevant* (written) in some of the 929 test lines (N). Clearly, non-relevant keywords are also interesting, since the system may erroneously find other similar words, thereby leading to important precision degradations. The main information about the query set is summarized in Tab. II.

TABLE II: Basic statistics of the selected keyword set in the test set.

	Total	Relevant
Line images (N)	929	855
Keywords (M)	3 421	1 081
Line-query events ($N \cdot M$)	3 178 109	1 916

C. Evaluation measures

The *interpolated average precision* (AP) [13] is used in this work to compare the different KWS methods. This summarizes the recall and interpolated precision measures into a single scalar value, corresponding to the area under the Recall-Precision (RP) curve.

Observe that *mean average precision* (mAP), chosen by many authors to report their performance, is ill-defined when some keywords are non-relevant. Thus, it cannot be used for the scenario considered in this work.

D. Experimental setup

The required likelihoods are computed using a left-to-right character-HMM, trained using the Baum-Welch algorithm, using the training line images of the IAM dataset and the provided transcription. Specific details of the image processing, feature extraction, and HMM training are described in [1]. All parameters were adjusted using the validation data, optimizing the character error rate (CER).

In order to generate the CL, we used different size n-gram language models, using the Kneser-Ney back-off technique. The n-grams were trained on the well-known Lancaster-Oslo/Bergen text corpus (LOB) [11]. Since the IAM corpus was build from the LOB, we excluded from the latter all sentences used in the test partition of the IAM database.

The CLs were generated using the HTK software [14] or the iATROS [15], depending on the order of the n-gram models. In order to limit the size of the generated CLs, a maximum node input degree (NID) of 30 edges, and a *beam* search was used during decoding. Additionally, the *grammar scale factor* and *character insertion penalty* parameters were adjusted using the validation data, as we did before, optimizing CER.

All the experimental setup is essentially the same as the one described in [5]. Moreover, we used exactly the same CLs. Thus, further details can be found in the previous paper, if needed.

E. Results

During the analysis of the results, we refer to Eq. 8 as "Viterbi", since it is the Viterbi approximation to Eq. 7, which is computed using a "Forward"-like algorithm. Tab. III shows their average precision results.

TABLE III: Average Precision results in IAMDB, using different n-gram sizes and approximations to $P(\mathcal{R} \mid \mathbf{v}, \mathbf{x})$. "Viterbi" refers to Eq. 8 (same as HMM-Filler), and "Forward" to Eq. 7.

n-gram	0	1	2	3	4	5	6
Viterbi	0.345	0.375	0.412	0.454	0.488	0.499	0.505
Forward	0.346	0.381	0.418	0.471	0.528	0.544	0.558
Abs. improv.	0.001	0.006	0.006	0.017	0.039	0.045	0.053
Rel. improv.	0.3%	1.8%	1.5%	3.7%	8.0%	9.1%	10.4%

The exact computation proves to be better than the Viterbi case, as expected by the theoretical development from Sect. III. Although for low-order n-gram LMs, the two methods perform similarly, when the size of the LM increases, the exact computation of $P(\mathcal{R} \mid \mathbf{v}, \mathbf{x})$ clearly shows to be superior to its Viterbi approximation.

It is specially remarkable the fact that the absolute and relative improvements, respect the Viterbi approximation, are almost monotonically increasing with the size of the n-gram (the only exception is for the bi-gram case). We believe that this is because of with low-order n-grams, the distributions learned by the LM are too noisy, and the benefits of the proposed algorithm are diminished by this noise.

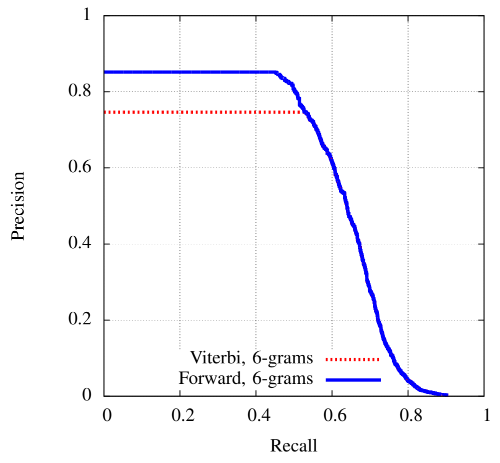


Fig. 3: Recall-Precision curve of the Viterbi and Forward approximations to $P(\mathcal{R} \mid \mathbf{v}, \mathbf{x})$, both using a 6-gram language model.

Fig. 3 shows the Recall-Precision curves of the KWS system using the Viterbi approximation and the exact computation of $P(\mathcal{R} \mid \mathbf{v}, \mathbf{x})$, using a 6-gram LM, and gives more insights about the behavior of two algorithms. The flat regions in both curves tell that there are many events with the same (high) score. The *forward* algorithm seems to discriminate better between the true relevant events and the false positive cases, which explains the increase in the maximum precision with only a slight drop in the minimum recall. Since the flat region is present in both cases, we hypothesize that it is due to the underlying model for $p(\mathbf{x}, \mathbf{w})$, and not the algorithms.

Finally, once the confidence of the underlying probabilistic model decays, both algorithms behave very similar as shown by the RP curve.

V. Conclusion

First, this paper introduced a probabilistic framework for Keyword Spotting (KWS). The core of the framework is to tackle the KWS task using a probability distribution over a binary variable $\mathcal{R}$, conditioned on the query keyword $\mathbf{v}$ and the image region $\mathbf{x}$, $P(\mathcal{R} \mid \mathbf{v}, \mathbf{x})$.

Later, from this probabilistic model, the traditional HMM-Filler model was derived and interpreted as an approximation to the previous probability. Furthermore, we presented an exact algorithm to compute $P(\mathcal{R} \mid \mathbf{v}, \mathbf{x})$, based on the well-known *forward* algorithm for HMMs.

We have experimentally shown that the proposed algorithm is always superior to the regular HMM-Filler, and yields to significant improvements in the accuracy of the KWS system, especially when used together with high order n-gram LMs. The improvements in AP are thanks to a achieving a better precision than the HMM-Filler model. Additionally, our results suggest that higher differences may be achieved with bigger n-gram models, which would also increase the AP. Notice that high-order n-grams only increase the time of building the CLs, which is carried only once, and do not affect the query-time, the most important from the user's perspective.

More research needs to be carried out in order to find where the benefits of the *forward* algorithm for computing $P(\mathcal{R} \mid \mathbf{v}, \mathbf{x})$, presented in this work, reach their maximum. Anyhow, the presented work shows significant improvements over the widely used HMM-Filler approach, on a widely used handwritten text database.

Acknowledgment

This work was partially supported by the Spanish MEC under FPU grant FPU13/06281 and under the STraDA research project (TIN2012-37475-C02-01), by the Generalitat Valenciana under the grant Prometeo/2009/014, and through the EU 7th Framework Programme grant tranScriptorium (Ref:600707).

References

[1] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Lexicon-free handwritten word spotting using character HMMs," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 934 – 942, 2012, special Issue on Awards from ICPR 2010.

[2] S. Wshah, G. Kumar, and V. Govindaraju, "Script independent word spotting in offline handwritten documents based on hidden markov models," in *International Conference on Frontiers in Handwriting Recognition (ICFHR),*, Sept., pp. 14–19.

[3] A. H. Toselli, E. Vidal, V. Romero, and V. Frinken, "Word-graph based keyword spotting and indexing of handwritten document images," Universidad Politcnica de Valencia, Tech. Rep., 2013.

[4] ——, "Word-Graph Based Keyword Spotting in Handwritten Document Images," 2013.

[5] A. H. Toselli, J. Puigcerver, and E. Vidal, "Context-Aware Lattice Based Filler approach for Key Word Spotting in Handwritten Documents," in *Proc. of the 13th Intl. Conf. on Document Analysis and Recognition (ICDAR'15)*, 2015, to be published.

[6] J. Puigcerver, A. H. Toselli, and E. Vidal, "Word-graph and character-lattice combination for kws in handwritten documents," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 181–186.

[7] A. H. Toselli and E. Vidal, "Fast HMM-Filler approach for Key Word Spotting in Handwritten Documents," in *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR'13)*, Washington, DC, USA, Aug. 2013.

[8] D. Knuth, J. Morris, Jr., and V. Pratt, "Fast pattern matching in strings," *SIAM Journal on Computing*, vol. 6, no. 2, pp. 323–350, 1977. [Online]. Available: http://dx.doi.org/10.1137/0206024

[9] A. Fischer, V. Frinken, H. Bunke, and C. Suen, "Improving HMM-Based Keyword Spotting with Character Language Models," in *Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR'13)*, Aug 2013, pp. 506–510.

[10] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *Intl. Journal on Document Analysis and Recognition*, vol. 5, pp. 39–46, 2002.

[11] S. Johansson, G. N. Leech, and H. Goodluck, *Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computers*, Department of English, University of Oslo, 1978.

[12] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A Novel Word Spotting Method Based on Recurrent Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 211 –224, Feb. 2012.

[13] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[14] S. Young, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book: Hidden Markov Models Toolkit V2.1*, Cambridge Research Laboratory Ltd, Mar. 1997.

[15] M. Luján-Mares, V. Tamarit, V. Alabau, C. D. Martınez-Hinarejos, M. P. i Gadea, A. Sanchis, and A. H. Toselli, "iATROS: A speech and handwritting recognition system," *V Jornadas en Tecnologıas del Habla (VJTH?2008)*, pp. 75–78, 2008.