Word-Graph and Character-Lattice Combination for KWS in Handwritten Documents

Joan Puigcerver
PRHLT Research Center
Universitat Politècnica de València
Camí de Vera s/n
46022 - València - Spain
Email: jpuigcerver@dsic.upv.es

Alejandro Héctor Toselli
PRHLT Research Center
Universitat Politècnica de València
Camí de Vera s/n
46022 - València - Spain
Email: ahector@dsic.upv.es

Enrique Vidal
PRHLT Research Center
Universitat Politècnica de València
Camí de Vera s/n
46022 - València - Spain
Email: evidal@dsic.upv.es

***Abstract*—We present a handwritten text Keyword Spotting (KWS) approach based on the combination of KWS methods using word-graphs (WGs) and character-lattices (CLs). It aims to solve the problem that WG-based models present for out of vocabulary (OOV) keywords: since there is no available information about them in the lexicon or the language model, null scores are assigned. OOV keywords may have a significant impact on the global performance of KWS systems, as we show. By using a CL approach, which does not suffer from the previous problem, to estimate the OOV scores, we take advantage of both models, using the speed and accuracy that WGs provide for in-vocabulary keywords and the flexibility of the CL approach. This combination improves significantly both average precision and mean average precision over the two methods.**

I. Introduction

Keyword Spotting in unsegmented handwritten text line images has already been addressed in [1]–[3], among others. In such approaches, a user query consists of a *keyword* and a more or less directly specified *confidence threshold*. For such a query, the system hypothesizes whether the keyword is present in each text line image with a confidence greater than the given threshold.

Following this idea, in [3], [4] a KWS approach is proposed based on Word-Graphs (WGs). A WG is produced during the Viterbi decoding of a text line image, using a standard handwritten text recognition (HTR) system based on *Hidden Markov Models* (HMMs) and *N-gram Language Models* (LMs). Using the multiple word-level segmentation and probabilistic information contained in a WG, adequate confidence scores are computed for all the words in the LM vocabulary. These scores are then used to generate a word index, which allows for extremely fast, confidence-level controlled look-up for word queries. Moreover, the precision-recall performance of the mentioned approach has been shown to be very competitive with state-of-the-art KWS approaches, including BLSTM KWS [1], which is perhaps the best HTR KWS method nowadays if its high computing training costs are not taken into account.

Clearly, the good performance of WG-based KWS comes from the rich contextual lexical and syntactic information which is more or less explicitly retained in the WGs. However, an important drawback of this approach is that out-of-vocabulary (OOV) keywords (that is, words not included in the LM lexicon) just get a null score, making the approach completely useless for OOV queries.

On the other hand, lexicon-free KWS approaches, such as the HMM-*Filler* [2] and also BLSTM [1], only rely on character-level processing and do not therefore suffer from this problem. Because of the prohibitive training costs of BLSTM, here we focus only on HMM-*Filler*. This approach uses the same character HMM models used in the WG-based approach. However, lacking a lexicon and a LM, the KWS accuracy of HMM-*Filler* often falls short of that of the WG approach. In addition, for each keyword (specified as an arbitrary character sequence), the standard HMM-*Filler* approach must perform a Viterbi decoding on the whole set of lines of the handwritten document collection. The computing time required for such an on-the-fly search often becomes prohibitive for large collections of handwritten images (for one million images, for instance, a single keyword query could require days or weeks of intensive computing).

In a recent work [5] we have shown that this prohibitive computing cost can be reduced as much as about two orders of magnitude by using *Character Lattices* (CLs) to compute the scores needed by the HMM-*Filler* approach, thereby making the HMM-*Filler* approach feasible for practical use. CLs, like WGs, are also obtained during Viterbi decoding of text line images, but using, as a LM, just a trivial concatenation of characters (i.e., the so called *filler* model).

According to this state of the affairs, practical applications involving large image collections call for using a hierarchy of spotting methods. In the first level, a lexicon-based index (obtained by means of WG-based KWS) provides very fast and accurate spotting results for hopefully usual queries. Then, for any new, non-indexed keyword, an affordable lexicon-free method, such as the CL-based HMM-*Filler*, can be used to provide reasonable spotting results. In real-world

KWS-based search and retrieval, supporting OOV keywords is crucial, since it is very likely that, over time, many queries will fall into this category.

With this in mind, this paper proposes a KWS approach which combines the WGs-based approach for spotting in-vocabulary keywords and the CL-based HMM-*Filler* method for handling OOV queries. According with empirical results on the well-known IAMDB dataset, the proposed combination achieves relative improvements of Average Precision and Mean Average Precision, as high as 11% and 19%, respectively.

The paper is organized as follows: Sec. II introduces basic concepts of HTR and WGs. Then, both the WG-based and the CL-based approaches are summarized in Sec. III. The proposed KWS combination method is presented in Sec. IV. In Sec. V are explained the performance assessment metrics used, database, experimental setup and obtained results. Finally, remarks and conclusions are drawn in Sec. VI.

II. HTR and WG Framework Review

To better understand the principles of both the WG-based and the lexicon-free CL-based approaches, this section reviews some basic concepts of the HTR technology based on HMMs and N-grams, as well as the required details about WGs/CLs. Fig. 1 shows an overview of the whole HTR process for a given line image.

A. HTR based on HMMs and N-Grams

Both the WG-based and CL-based approaches to KWS rely on line-level HTR processing. Text line images can be obtained using well-known text line detection and segmentation techniques [6]. For each line image, a sequence of d-dimensional feature vectors, $\mathbf{x} = \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n,\ \vec{x}_i \in \Re^d$, is obtained after applying different normalization and feature extraction techniques (see details in [7]).

The fundamentals of the standard HTR technology based on HMMs and N-gram LMs were originally presented in [8] and further developed in [9], [10], among others. Given an input image represented by $\mathbf{x}$, the problem is to find a most likely word sequence, $\widehat{\mathbf{w}} = \widehat{w}_1\widehat{w}_2\ldots\widehat{w}_l$, according to:

$$\widehat{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{x}) \;= \arg\max_{\mathbf{w}} p(\mathbf{x}|\mathbf{w}) \cdot P(\mathbf{w}) \qquad (1)$$

The conditional density $p(\mathbf{x}|\mathbf{w})$ is modeled by morphological word models, built by concatenating character HMMs [11], [12], and the prior $P(\mathbf{w})$ by an N-gram language model [11].

In contrast with the WG approach, in the CL-based method HTR is performed at character-level, rather than at word-level. However, the same Eq. (1) applies by interchanging the search for a most likely word sequence $\widehat{\mathbf{w}}$ with the search for a most likely character sequence $\hat{\mathbf{c}}$. In this case, the Viterbi score, $S(\mathbf{x})$, associated with $\hat{\mathbf{c}}$ is:

$$S(\mathbf{x}) = \max_{\mathbf{c}} p(\mathbf{x}|\mathbf{c}) \cdot P(\mathbf{c}) \qquad (2)$$

where $p(\mathbf{x}|\mathbf{c})$ is approximated by the morphological character HMMs previously trained and used for the WG approach. The prior distribution, $P(\mathbf{c})$, is now a trivial uniform distribution of character occurrences, as in the standard HMM-*Filler* method [2] (see Sec. III-B).

Eqs. (1–2) can be solved by means of Viterbi decoding [11]. As a by-product, a huge set of most likely word (or character) sequences, along with their corresponding likelihoods and segmentation hypotheses, can be obtained and compactly represented in the form of a *"word-graph"* [13] (see Fig. 1).

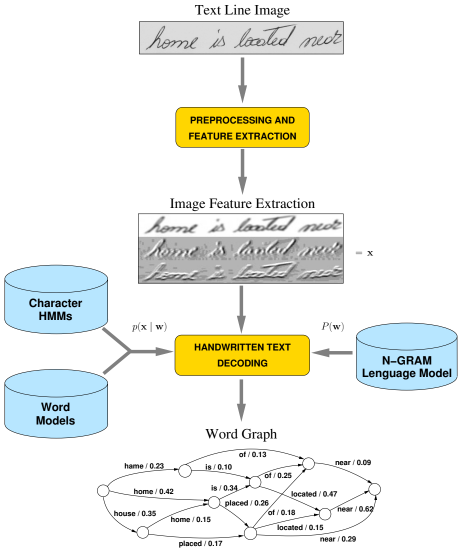


Figure 1. Diagram of the HTR decoding process. The input image is processed and different features are obtained: average grey-level and horizontal & vertical components of the grey-level gradient. The decoding step uses character HMMs, lexicon word models and N-gram language models to build the corresponding WG.

B. Word-Graphs

A WG (or CL) of a vector sequence $\mathbf{x}$ is a weighted directed acyclic graph $G(\mathbf{x}) = (Q, q_0, F, \tau, \omega, \delta)$, with initial node $q_0 \in Q$ and a set of final nodes $F \subseteq (Q - q_0)$. Each node q has associated an integer given by $\tau(q_i) \in [0, n]$, where n is the length of the sequence $\mathbf{x}$ (it represents a horizontal position of the image represented by $\mathbf{x}$). For every edge $(q, q') \in E$ $(q \neq q', q \notin F, q' \neq q_0)$, $\omega(q, q') = v$ associates a word (character) v to the edge and $\delta(q, q')$ is a score, corresponding to the likelihood that the word (character) $\omega(q, q')$ is written in the *image segment* represented by vectors ("frames") $x_{\tau(q)+1}, \ldots, x_{\tau(q')}$. These words (characters), segmentation marks and likelihoods are given by the Viterbi decoding process.

The best hypothesis $\widehat{\mathbf{w}}$ of Eq. (1) (and/or the best score $S(\mathbf{x})$ as in Eq. (2)) can be obtained by searching for a best complete path in the corresponding WG (or CL); that is, a sequence of connected edges from q_0 to some $q_F \in F$, such that the accumulated edge score is maximum.

III. Overview of WG-based and CL-Based KWS

A. WG-based KWS

The KWS approach for handwritten text line images we proposed in [4] is used here. For each keyword v and each text line represented by $\mathbf{x}$, a score $S_G(v,\mathbf{x})$ is computed as:

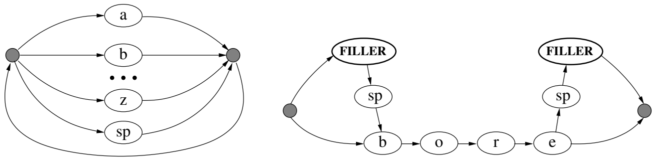
$$S_G(v,\mathbf{x}) \overset{\text{def}}{=} \max_{1\leq i\leq n} P(v|\mathbf{x},i) \tag{3}$$

where $P(v|\mathbf{x},i)$, called *frame-level word posterior*, is the probability that the word v appears at position i of $\mathbf{x}$. As shown in [3], [4], $P(v|\mathbf{x},i)$ can be very accurately approximated by considering the probability contributions of all the edges of $G(\mathbf{x})$, labeled with the keyword v whose segmentation hypotheses include the frame i.

$S_G(v,\mathbf{x})$ is bounded within $[0,1]$ and measures the system's confidence about the statement: "keyword v is in line image $\mathbf{x}$". It can be computed for all line images of the document collection in a *preparatory phase*, just after the WG of each line is obtained. The resulting scores can be organized into an adequate word-based index so that, in the *searching phase*, honoring any query involving an indexed word can be very fast and computationally cheap.

B. CL-based KWS

In the classical HMM-*Filler* KWS approach, as presented in [2], character HMMs are used to build both a *"filler"* model, f, and a *keyword-specific* model, k_v, for each individual keyword v to be spotted, as shown in Fig. 2. Each of these models correspond to a different prior probability $P(\mathbf{c})$ in Eq. (2). The filler, f, assigns a uniform probability to any equal-length unrestricted sequence of characters. From these sequences, k_v assigns a null probability to those which do not contain the word v.



(a) Character Filler HMM: f

(b) Keyword HMM: k_v

Figure 2. (a) "Filler HMM" (f) and (b) "keyword HMM" (k_v) built for the keyword v = "bore".

In a *preparatory phase*, f is used once for each line image $\mathbf{x}$ to compute the Viterbi decoding score $S(\mathbf{x})$ (Eq. (2)). Similarly, in the *searching phase*, for each line image $\mathbf{x}$, the Viterbi decoding score $S_v(\mathbf{x})$ is computed for each keyword v to be spotted, using the keyword-specific model k_v. A HMM-*Filler* spotting score $S_F(v,\mathbf{x})$ is then defined as:

$$S_F(v,\mathbf{x}) \overset{\text{def}}{=} \frac{\log S_v(\mathbf{x}) - \log S(\mathbf{x})}{l_v} \tag{4}$$

where l_v is the length of v in number of frames between the word detected borders.

In the *preparatory phase*, computing $S(\mathbf{x})$ for all lines is generally affordable; but the cost of the *searching phase* can become prohibitive, since $S_v(\mathbf{x})$ has to be computed for every spotted keyword v. To deal with this problem, we recently proposed the CL-based method, which computes $S_v(\mathbf{x})$ using CLs generated in the preparatory phase as a by-product of computing $S(\mathbf{x})$ [5]. The idea is simple: a CL obtained by decoding $\mathbf{x}$ with f contains a huge number of character sequences that may explain the input image $\mathbf{x}$. If the keyword v is (well) written in the image, v will be a proper sub-sequence of some (or many) of these character sequences. Among these, the one with maximum likelihood can be used to very accurately approximate $S_v(\mathbf{x})$ [5].

IV. Back-off combination of WG and CL KWS

As commented in Sec. I, an important drawback of WG-based KWS is that it is completely useless for OOV keywords. However, at query time, it is trivial to tell apart whether a given keyword is OOV or not. If *not* (i.e., if it is indexed), the query can be immediately honored using the precomputed indexed scores; otherwise, the CL-based HMM-*Filler* can be used as a "back-off" approach. Accordingly, a combined spotting score $S(v,\mathbf{x})$ can be written as:

$$S(v,\mathbf{x}) = \begin{cases} S_G(v,\mathbf{x}) & v \in V \\ \exp(S_F(v,\mathbf{x}))^{\eta} & \text{otherwise} \end{cases} \tag{5}$$

where V is the LM vocabulary. It is important to note that while S_G is properly normalized in the $[0-1]$ interval (and can be properly interpreted probabilistically), S_F can not be directly interpreted in probabilistic terms and their (negative) range is unbounded. Therefore, for practical use along with S_G, S_F needs to be monotonically mapped into an adequate interval by means of the exponential function and the weight parameter η. Observe that the distribution of scores given by $\exp(S_F)$ and S_G may differ substantially, thus η is required to tune the KWS performance measured in terms of *average precision*. This parameter is tuned using a development set.

The proposed combination scheme is reminiscent of the well-known *Back-off smoothing* used for language modeling: when the information about an event (keywords vs. n-grams) is not available at a certain level, it is estimated using lower level models (characters vs. $(n-1)$-grams). Thus, we will further refer to this combination as B-Comb.

V. Experimental Setup and Results

A. Corpus Description

For the sake of fair comparison and to easy reproducibility, the well-known, publicly available IAMDB corpus, commonly used in HTR benchmarks, has been used to assess the proposed approach. Other datasets have also been used in the past, like the *George Washington* or the *Parzival* [1], [2], [14]–[16], but these works are not comparable with the work presented here, because of the choice of keywords discussed in V-C. Therefore, we restrict ourselves to the IAMDB.

IAMDB consists of modern English handwritten text samples from many writers, compiled by the FKI-IAM Research Group on the base of the Lancaster-Oslo/Bergen Corpus (LOB). The last released version (3.0) is composed of 1 539 scanned text pages, handwritten by 657 different writers and partitioned into writer-independent training, validation and test sets. The line segmentation provided with the corpus [17] is used here. Statistics of the IAMDB corpus appear in Table I. The "text data" information for this corpus refers to three external text corpora (LOB, Brown, and Wellington, collectively called "LBW") which were employed for compiling the 20K-word lexicon and for training the IAMDB bi-gram language model [18] used in the WG-based KWS setting.

Table I
BASIC STATISTICS OF THE IAMDB AND THE CORRESPONDING PARTITIONS. "TEXT DATA" SUMMARIZES THE DATA USED TO TRAIN THE LANGUAGE MODEL.

		Training	Validation	Test	Total
Image data	Running chars	269 270	39 318	39 130	347 718
	Char set size	72	69	65	81
	Running words	47 615	7 291	7 197	62 103
	Lexicon size	7 778	2 442	2 488	9 809
	Lines	6 161	920	929	8 010
Text data	Lexicon size	19 892	2 442	2 488	20 832

B. Evaluation Measures

The standard *recall* and *interpolated precision* measures [19] are used here. *Interpolated precision* is widely used in the literature to avoid cases in which plain precision can be ill-defined. Since precision and recall are functions of a threshold used to determine whether the score $S(v, \mathbf{x})$ is high enough to assume that v appears in $\mathbf{x}$, they can be plotted as a so-called *recall-precision* (R-P) curve. This curve actually shows the interrelated trade-off between recall and precision. The overall R-P behavior is summarized into a single scalar measure known as *average precision* (AP) [20] [21], defined as the area under the R-P curve.

In addition, we use the *mean average precision* (MAP), which is also very often adopted in the KWS literature. It is computed by averaging the individual *average precision* of each keyword.

C. Set of Keywords to be Spotted

Two main, rather opposite criteria are often adopted for establishing a set of keywords which are adequate to assess KWS performance. The first one is to spot only *relevant keywords*; that is, words which are known to appear at least once in the test images. The vast majority of KWS works in OCR and HTR literature adopt this criterion. The other criterion is to take the query set from the vocabulary observed in a set of training images.

Here, in contrast to recent literature on KWS [1]–[5] the first criterion is adopted, since the second one ignores the problem of spotting OOV keywords. More specifically, we chose all the words which appear in the test set, after filtering out *stop words*, such as articles, prepositions, etc. (the same criterion was adopted for the validation partition).

Let M be the number of selected keywords and N the number of test line images. The score of the M keywords has to be computed for each of the N images, resulting in $M \cdot N$ line-query events in total. From these, only a small number will be *relevant* (the keyword is actually present in the image). Finally, note that the proposed KWS combination is challenged by the fraction of relevant line-query events involving OOV keywords (since the WG approach underestimate their scores). Tab. II summarizes these statistics.

Table II
DETAILS OF THE SELECTED SETS OF KEYWORDS FOR THE IAMDB.

	Validation	Test
# Line images: N	920	929
# Query words: M	2 134	2 209
# Line-query events: $M \cdot N$	1 963 280	2 052 161
# OOV Line-query events	400 200	405 973
# Relevant line-query events	3 384	3 446
# Relevant OOV line-query events	497	496

D. Proposed System Setup

The IAMDB training partition was used to train the character HMMs. A left-to-right HMM was trained for each of the elements appearing in the training text images, such as lowercase and uppercase letters, symbols, special abbreviations, possible spacing between words and characters, crossed-words, etc. Details about the meta-parameters employed for the line-image preprocessing, writing style attribute normalization, feature extraction and HMM training setup (all which were optimized on the validation data) can be found in [2].

In the *preparatory phase* of the WG-based and CL-based KWS approaches, the WG and CL of each line image was generated. WGs were obtained employing a bi-gram LM trained with the external text corpora LWB with a 20K-word lexicon [18]. Both the WGs and the CLs were generated using the HTK toolkit [22], setting values of 40 and 30 respectively for the parameter which specifies the maximum node input degree (NID). The amount of nodes and edges of the CLs is much larger in spite of using a lower NID and a reduced vocabulary size corresponding to the set of 81 characters. This is due to the lack of modeling restrictions that, in the case of WGs, were supplied by the LM and lexicon. This results in big graphs with more than 850 nodes, 22K edges and 10^{21} paths, for the WGs, and 37K nodes, 1M edges and 10^{307} paths for the CLs, on average.

Once the WGs and CLs were generated, they were processed as discussed in Sec. III. Spotting scores, $S_G(v, \mathbf{x})$, for all in-vocabulary words, v, were computed as discussed in Sec. III-A. On the other hand, the corresponding OOV query scores $S_F(v, \mathbf{x})$ were computed as explained in Sec. III-B. Finally, the proposed back-off smoothed spotting score $S(v, \mathbf{x})$, was obtained according to Eq. (5), after optimizing the parameter η on the IAMDB validation set with the corresponding set of queries (value set to 2.2).

E. Results

Fig. 3 shows the *recall-precision* (R-P) curve of each method and Table III shows the corresponding average and mean average precision results. Baseline HTR is the performance when the transcription provided by a HTR system is indexed to perform KWS. This is equivalent to the WG approach with NID equal to 1. It was first surprising that the AP of this method was higher than the Baseline CL. However, CL-based methods suffer from a normalization problem in their scores, which greatly affects the AP. Eq. (4) introduced l_v to mitigate this problem, but it is not completely solved (Fig. 4(d) also shows this problem). Moreover, the CL-Filler can not benefit from the contextual information provided by the word Language Model.

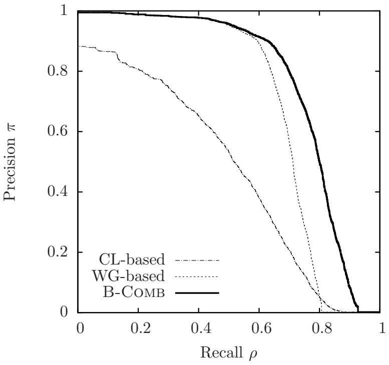


Figure 3. Recall-Precision curves of the three KWS methods on the IAMDB test set.

Table III
RESULTS OF THE DIFFERENT KWS MODELS ON THE IAMDB TEST SET.

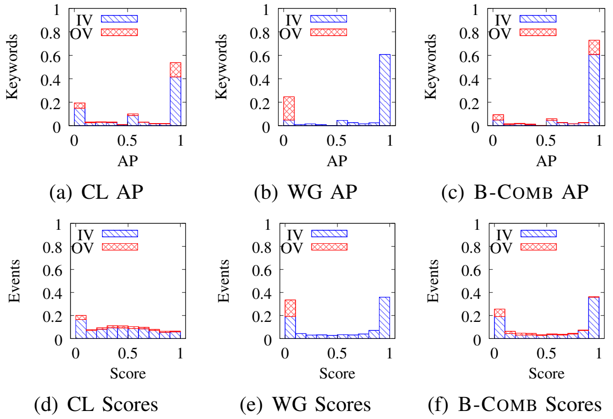
Model	AP	MAP
Baseline HTR	0.513	0.524
Baseline CL	0.467	0.665
Baseline WG	0.691	0.688
B-COMB	**0.769**	**0.822**

The best baseline model was the WG-based, with an AP of 0.691 and a MAP equal to 0.688. With the proposed combined method, the AP increases to 0.769 and the MAP to 0.822; that is, an absolute improvement of 7.8 percent points in AP and 13.4 in MAP.

As shown in Fig. 4(b), the WG-based individual AP of OOV keywords is close to zero, while the in-vocabulary keywords achieve a good AP generally. By using the back-off combination, the OOV distribution of individual AP (Fig. 4(c)) is that of the CL-based (Fig. 4(a)), which results in much better MAP global performance.

The same occurs in the distribution of the scores of the individual events, which affects the AP. Ideally, all relevant events should have a score close to 1 and the non-relevant ones to 0. However, the distribution for relevant events varies on each method, thus the need of η introduced in Eq. (5). The CL-based distributes the scores rather uniformly (Fig. 4(d)), which does not affect the MAP but damages the AP (see Tab. III). However, in the case of the relevant OOV events, the WG-based model does much worse, since all of them have a null score. The combination results in a better global distribution, which leads to the superior global AP observed.

This analysis explains how B-COMB achieves the observed important improvements: It takes advantage of the capabilities of both models by using the lexical information provided by WGs, when it is available, or using the flexibility of the CL-based method for OOV keywords.



(a) CL AP (b) WG AP (c) B-COMB AP

(d) CL Scores (e) WG Scores (f) B-COMB Scores

Figure 4. Histograms of AP and KWS scores. Each bars includes two parts stacked: The contribution of in-vocabulary keywords (IV) in blue and that of out-of-vocabulary (OOV) in red.

VI. REMARKS AND CONCLUSIONS

An approach which combines word-graph and character-lattice techniques for keyword spotting in handwritten documents has been presented. It solves the main problem exhibited by WG-based KWS for OOV keywords: since they are not modeled by the underlying word LM, a null score is assigned to them leading to a uselessly low AP.

Character-level models, like the CL-based HMM-Filler approach, provide a better modeling of OOV keywords, since they do not rely on any given word lexicon. However, they generally offer a worse global performance.

The presented work uses the strengths of both models by using the WG-based scores for in-vocabulary keywords and the HMM-Filler CL-based scores for OOV. This simple approach offers an absolute improvement of 7.8% on the AP (11.3% relative) and 13.4% on the MAP (19.5% relative).

Clearly, when the amount of OOV events is not negligible (as in our experiments), improving the OOV score estimates has an important impact in the global performance of the system, as reflected both in terms of AP and MAP. Additionally, we observed that a scaling parameter is needed to equalize WG-based and CL-based score distributions. This has no effect on MAP but significantly affects AP results.

Lastly, the proposed approach is reminiscent of the well-know *Back-off method* used in language modeling. This suggests that other combination techniques, like linear interpolation, and the usage of other character models, like BLSTM, may give also significant improvements in global KWS performance.

Acknowledgments

This work was partially supported by the Spanish MEC 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] 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.

[2] 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.

[3] A. H. Toselli, E. Vidal, V. Romero, and V. Frinken, "Word-Graph Based Keyword Spotting in Handwritten Document Images," 2013, under review.

[4] ——, "Word-graph based keyword spotting and indexing of handwritten document images," Universidad Politécnica de Valencia, Tech. Rep., 2013.

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

[6] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," *Intl. Journal on Document Analysis and Recognition*, vol. 9, pp. 123–138, April 2007.

[7] V. Romero, A. H. Toselli, and E. Vidal, *Multimodal Interactive Handwritten Text Transcription*, ser. Series in Machine Perception and Artificial Intelligence (MPAI). World Scientific Publishing, 2012, http://www.worldscientific.com/worldscibooks/10.1142/8394.

[8] I. Bazzi, R. Schwartz, and J. Makhoul, "An Omnifont Open-Vocabulary OCR System for English and Arabic," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 495–504, 1999.

[9] A. Vinciarelli, S. Bengio, and H. Bunke, "Off-line recognition of unconstrained handwritten texts using HMMs and statistical language models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 709–720, june 2004.

[10] A. H. Toselli, A. Juan, D. Keysers, J. González, I. Salvador, H. Ney, E. Vidal, and F. Casacuberta, "Integrated Handwriting Recognition and Interpretation using Finite-State Models," *Intl. Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 4, pp. 519–539, 2004.

[11] F. Jelinek, *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[12] L. Rabiner, "A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition," *Proceedings IEEE*, vol. 77, pp. 257–286, 1989.

[13] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, mar 2001.

[14] R. Manmatha and W. Croft, "Word spotting: indexing handwritten archives," *Intelligent multimedia information retrieval*, pp. 43–64, 1997.

[15] T. M. Rath, R. Manmatha, and V. Lavrenko, "A search engine for historical manuscript images," in *Proc. of the Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, New York, NY, USA, 2004, pp. 369–376.

[16] T. Rath and R. Manmatha, "Word spotting for historical documents," *Intl. Journal on Document Analysis and Recognition*, vol. 9, pp. 139–152, 2007.

[17] 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.

[18] R. Bertolami and H. Bunke, "Including language model information in the combination of handwritten text line recognisers," in *Proc. of the Intl. Conf. on Frontiers in Handwriting Recognition (ICFHR'08)*, 2008.

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

[20] M. Zhu, "Recall, Precision and Average Precision," Working Paper 2004-09 Department of Statistics & Actuarial Science - University of Waterloo, August 26 2004.

[21] S. Robertson, "A new interpretation of average precision," in *Proc. of the Intl. ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '08)*. New York, NY, USA: ACM, 2008, pp. 689–690.

[22] 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.