Neural Computing and Applications
Word-Graphs Size Impact on the Performance of Handwriting Document Applications
--Manuscript Draft--

Manuscript Number:	
Full Title:	Word-Graphs Size Impact on the Performance of Handwriting Document Applications
Article Type:	S.I. : IbPRIA 2015
Keywords:	computer assisted transcription of text images; keyword spotting for handwritten text; historical handwritten manuscripts; word graphs; evaluation performance; generation of word lattices
Corresponding Author:	Alejandro Héctor Toselli, Ph.D. Universitat Politècnica de València Valencia, SPAIN
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Universitat Politècnica de València
Corresponding Author's Secondary Institution:	
First Author:	Alejandro Héctor Toselli, Ph.D.
First Author Secondary Information:	
Order of Authors:	Alejandro Héctor Toselli, Ph.D.
	Verónica Romero, Ph.D.
	Enrique Vidal, Ph.D.
Order of Authors Secondary Information:	
Funding Information:	Generalitat Valenciana grant PrometeoII (2014/030) — Not applicable
	Seventh Framework Programme (600707) — Not applicable
	STraDa (TIN2012-37475-C02-01) — Not applicable
Suggested Reviewers:	Laurence Likforman-Sulem TELECOM ParisTech laurence.likforman@telecom-paristech.fr Experience in handwritten documents recognition. Paris, France.
	Bill Barrett Brigham Young University barrett@cs.byu.edu Experience in Document Recogntion and Understanding. Provo, Utah, USA.
	Ogier Jean-Marc Universite de La Rochelle jean-marc.ogier@univ-lr.fr Experience in Document Analysis and Pattern recognition. La Rochelle, France
	Bertrand Coüasnon IRISA Institur de Recherche - Rennes Bertrand.Couasnon@irisa.fr Experience in Document Analysis and Understanding. Rennes, France.
	Umapada Pal Indian Statistical Institute, Kolkata umapada@isical.ac.in Experience in Handwritten Document Recognition, Word Spotting and Document Image Analysis. Kolkata, India.

	Venu Govindaraju University at Buffalo - The State University of New York venu@cubs.buffalo.edu Experience in Machine Learning, Biometrics, Language Technologies. Amherst, New York, USA.

Noname manuscript No.
(will be inserted by the editor)

Word-Graphs Size Impact on the Performance of Handwriting Document Applications

Alejandro H. Toselli · Verónica Romero · Enrique Vidal

Received: date / Accepted: date

Abstract Two document processing applications are considered: *Computer Assisted Transcription of Text Images* (CATTI) and *Key-Word Spotting* (KWS), for transcribing and indexing handwritten documents, respectively. Instead of working directly on the handwriting images, both of them employ meta data structures called Word Graphs (WG), which are obtained using segmentation-free handwritten text recognition technology based on N-gram Language Models and Hidden Markov Models. A WG contains most of the relevant information of the original text (line) image required by CATTI and KWS but, if it is too large, the computational cost of generating and using it can becomes unaffordable. Conversely, if it is too small, relevant information may be lost, leading to a reduction of CATTI or KWS performance. We study the trade-off between WG size and performance in terms of effectiveness and efficiency of CATTI and KWS. Results show that small, computationally cheap WGs can be used without loosing the excellent CATTI and KWS performance achieved with huge WGs.

1 Introduction

In recent years, huge amounts of historical handwritten documents have been scanned into digital images, which are then made available through web sites of libraries and archives all over the world. However, the wealth of information conveyed by the text captured in these images remains largely inaccessible (no plain text, difficult to read even for researchers). Therefore, automated methods are needed to add value to mass-digitization and preservation efforts of Culture Heritage institutions, in order to provide adequate access to the *contents* of the preserved collections of handwritten text documents. To this end, the *tranScriptorium*[1] project [17] aims to fulfill these needs

Alejandro H. Toselli, Verónica Romero and Enrique Vidal
PRHLT Research Centre, Universitat Politècnica de València,
Camino de Vera, s/n - 46022 Valencia - Spain
E-mail: [ahector, vromero, evidal]@prhlt.upv.es

[1] `http://www.transcriptorium.eu`

with the development of two different applications: *Computer Assisted Transcription for Test Images* (CATTI) [16], intended to speed up transcription processes, and *Keyword Spotting* [24] for automatic indexing of untranscribed handwritten material under the so called *Precision-Recall trade-off model*. Actually, both applications rely on *word lattices* or *Word Graphs* (WG).

A WG is a data structure proposed by several authors some decades ago during the development of Automatic Speech Recognition (ASR) technology [10]. Nowadays, WGs are also being used in the fields of *machine translation* (MT) [26], and lately in *handwritten text recognition* (HTR) [16,23]. In HTR, WGs are obtained through a natural extension of the standard dynamic programming Viterbi decoding algorithm, which determines the single best HTR hypothesis. A WG represents very efficiently a huge number of word sequence hypotheses whose posterior probabilities are large enough, according to the morphological character likelihood and the prior (language) models, used to decode a text line image. WGs also store additional important data about these hypotheses; namely, alternative word segmentations and word decoding likelihoods.

An important shortcoming of WGs is the large computing cost entailed by their generation, often very much larger than the cost of the basic Viterbi decoding process itself. WG generation cost depends on many factors, including the input sequence length and decoding vocabulary size. But a major factor is, by far, a parameter known as *maximum node input degree* (IDG), which specifies the amount of information retained at each node during the WG generation process. In addition to reducing IDG, other pruning techniques, such as *beam-search*, *histogram pruning*, etc. can also be used to accelerate the WG generation process at the expense of some loss of the information retained in the resulting WGs [10,18,30].

This work, which extends the one presented in [20], studies how different sizes of WGs, pruned by different IDG values, impact on the effectiveness/efficiency of both CATTI and KWS applications. This study will serve as a reference for making good enough estimations of required space-time resources for tasks entailing the processing of massive handwritten images using WG-based CATTI and KWS.

2 Overview of HTR and WG Technology

This section is devoted to introduce the basics of the *handwritten text recognition system* (HTR) used to generate WGs required by CATTI and KWS.

2.1 HTR based on HMMs and *N*-Grams

Holistic, segmentation-free HTR technology is used in this work to produce WGs. It is based on *Hidden Markov Models* (HMMs) and *N*-grams, and follows the fundamentals presented in [1] further developed in [27,21], among others. This kind of recognizer accepts a handwritten text line image, represented as a sequence of *D*-dimensional feature vectors $\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, $\mathbf{x}_i \in \Re^D$, and find a most likely word

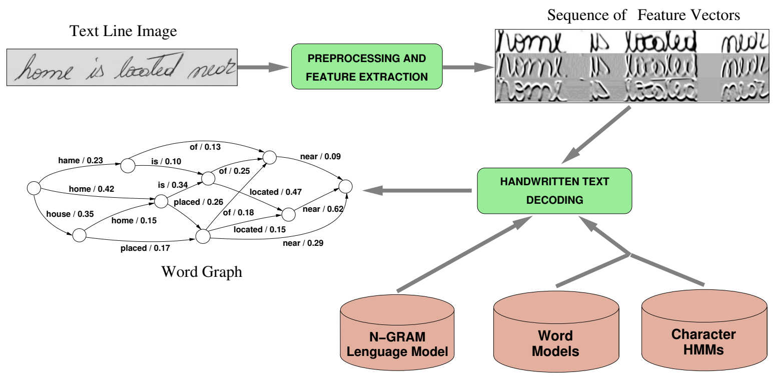

Fig. 1 Diagram of the HTR decoding process. Starting from a text line image of the text "home is located near", a visual representation of the obtained feature vector sequence is shown: average grey-level and horizontal & vertical components of the grey-level gradient. Finally, the corresponding WG is delivered by the decoding process, using the provided knowledge sources: morphological character HMMs, lexicon word models and N-gram language model.

sequence $\widehat{\mathbf{w}} = \widehat{w}_1 \widehat{w}_2 \ldots \widehat{w}_l$, according to:

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

The conditional density $p(\mathbf{x} \mid \mathbf{w})$ is approximated by morphological word models, built by concatenating character HMMs [5,13], and the prior $P(\mathbf{w})$ is approximated by an N-gram language model [5]. Two main modules comprise the HTR process illustrated in Fig. 1: preprocessing and feature extraction, and decoding. Preprocessing generally entails line image enhancement and basic geometry corrections, including slant normalization, while feature extraction obtains an image representation in terms of a sequence of feature vectors. A simple feature extraction method based on grey levels and grey-level gradients [22,21] is illustrated in Fig. 1.

The search (or decoding) of $\widehat{\mathbf{w}}$ is optimally carried out by the Viterbi algorithm [5] also referred to as *token-passing* [31,9]. This dynamic-programming decoding process can yield not only a single best solution, as in Eq. (1), but also a huge set of best solutions compactly represented into a WG. For a more detailed description of this HTR system, including text line processing, model training and decoding, the reader is referred to [16].

2.2 Word-Graphs

A WG (also called *word lattice*) is a weighted directed acyclic graph whose edges are labelled with words and weighted with scores derived from the HMM and N-gram probabilities computed during the line image decoding process. It is defined as a finite set of nodes Q and edges E, including an initial node $\nu_I \in Q$ and a set of final nodes $F \subseteq (Q - \nu_I)$. Each node ν is associated with a horizontal position of $\mathbf{x}$, given by $t(\nu) \in [0,n]$, where $t(\nu_I) = 0$ and $\forall_{\nu_F \in F}\; t(\nu_F) = n$, being n the length of $\mathbf{x}$. For an edge $(\nu', \nu) \in E$ $(\nu' \neq \nu, \nu' \notin F, \nu \neq \nu_I)$, $v = \omega(\nu', \nu)$ is its associated word and

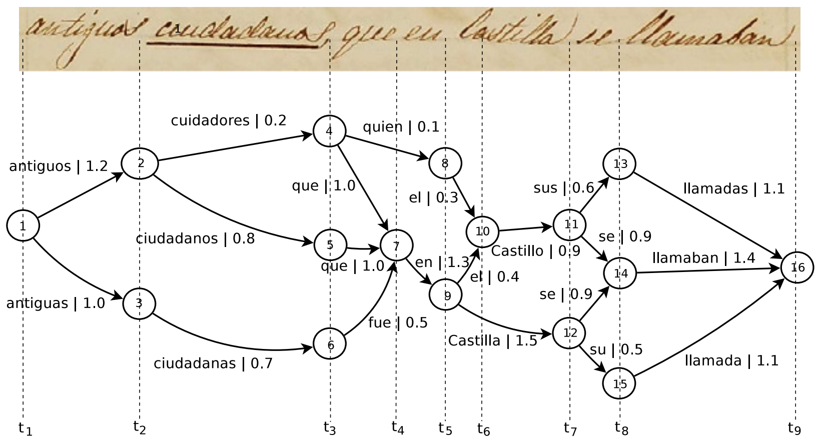

Fig. 2 Illustrative, simplified example of a WG that would be obtained from the decoding of a line image of the Spanish handwritten text: "antiguos ciudadanos, que en Castilla se llamaban", represented by a sequence of feature vectors, **x**, of length n. Each edge (v', v) is labelled with the corresponding word, $\omega(v', v)$, and weighted with its score $s(v', v)$. Node positions, $t_v \equiv t(v)$, corresponding to different word segmentations, are also shown on the bottom and in the image itself.

$s(v', v)$ is its score, corresponding to the likelihood that the word v appears in the *image segment* delimited by frames $t(v') + 1$ and $t(v)$, computed during the Viterbi decoding of **x**.

A *complete path* of a WG is a sequence of nodes starting with node v_I and ending with a node in F. Complete paths correspond to whole line decoding hypotheses.

Fig. 2 shows a small, illustrative example of a WG obtained by decoding the text line image also shown in this figure.

2.3 WG Generation Overview

In this work, WGs are generated using the HTK toolkit [30], whose decoder implements the Viterbi-like *token-passing* algorithm [31,9]. The underlying idea is that each word-level HMM state at each time holds a "moveable token", which propagates (updating its information) to same or a new state for the following time stamp along the decoding process. Among other things, tokens contain partial log probabilities and path identifiers to allow path trace-back. A path identifier is actually a pointer to a record of word boundary information called *Word Link Record* (WLR). During token propagation through word-level transitions (i.e. external HMM connections between words), potential word boundaries are recorded in a linked list of WLRs. On decoding completion, the path identifier held in the token with the maximum score provides the WLR linked list containing the best matching word sequence and also the corresponding word boundary locations.

A WG is straightforwardly obtained by recording, not only the best token, but the N-best tokens emitted at each syntactically distinct word boundary. Thus, upon decoding completion, the WLR linked lists can be easily converted into a WG by applying the *Chart* parser [29]. It is important to remark that WGs generated in this way are unambiguous; that is, no two complete paths exist in a WG which correspond to the same sequence of words.

Many other methods to generate WGs have been proposed, most of them closely related to specific types of decoders. For instance, in [12] a method based on automata composition is described for a (spoken signal) decoder based on *weighed finite state transducers* [8].

2.4 Computational Cost of WG Generation

It is well known that the computational complexity of the Viterbi algorithm is linear in the length of $\mathbf{x}$ and the cost can be made largely independent of the lexicon size and the overall size of the models used by means of well known pruning techniques such as *beam-search* [5]. However, when the decoding process includes WG generation, the overall computing cost is observed to grow very fast with the WG size (exponentially with the WG density, according to [18]). All in all, the asymptotic cost of generating a WG for a line image of length n can be expressed as $O(\varGamma \cdot n)$, where $\varGamma$ is a (generally large) constant which depends on the WG size. Nevertheless, it should be taken into account that this process is carried out only once, and by choosing adequate WG sizes, reasonable WG generation time can be achieved in practice, as it will be shown later on in this paper.

3 Outline of WG-based CATTI and KWS

3.1 WG-based CATTI

Interactive computer assisted transcription of text images (CATTI) is presented in detail in [16]. In this framework, the human transcriber is directly involved in the handwritten text transcription process and he/she is responsible of validating and/or correcting the HTR output.

The interactive process starts when the HTR system proposes a full transcript of a feature vector sequence $\mathbf{x}$, extracted from a handwritten text line image. In each interaction step the user validates a prefix of the transcript which is error free and makes some amendment(s) to correct the erroneous text that follows the validated prefix, producing a new correct prefix $\mathbf{p}$. The new, extended prefix is used by CATTI to search for a new most likely suffix, $\hat{\mathbf{s}}$, according to:

$$\hat{\mathbf{s}} = \arg\max_{s} P(\mathbf{s} \mid \mathbf{x}, \mathbf{p}) \approx \arg\max_{s} p(\mathbf{x} \mid \mathbf{p}, \mathbf{s}) \cdot P(\mathbf{s} \mid \mathbf{p}) \qquad (2)$$

Equation (2) is very similar to Equation (1), being $\mathbf{w}$ the concatenation of $\mathbf{p}$ and $\mathbf{s}$. As in conventional HTR, $p(\mathbf{x} \mid \mathbf{p}, \mathbf{s})$ can be approximated by HMMs and $P(\mathbf{s} \mid \mathbf{p})$ by and N-gram model; but now the N-gram is conditioned by $\mathbf{p}$, which is given. Therefore, the search must be performed over all possible suffixes $\mathbf{s}$ of $\mathbf{p}$, rather than over complete transcripts as in Eq. (1).

This search can be carried out through an extension of the conventional Viterbi algorithm, which directly uses the original N-gram, and HMMs to incrementally process the given vector sequence $\mathbf{x}$ [22, 19]. However, the computational cost of this approach becomes prohibitive for the very short response time generally needed for adequate interactive operation.

As shown in [16], much more efficient search can be achieved using the WG obtained during the plain Viterbi decoding of the whole image representation $\mathbf{x}$, as outlined in Sec. 2. In each interaction step, the decoder parses the previously validated prefix $\mathbf{p}$ over the WG. This parsing procedure will end defining a set of nodes Q_p corresponding to paths from the initial node whose associated word sequence is $\mathbf{p}$. Then, the decoder continues searching from any of the nodes in Q_p for a suffix $\mathbf{s}$ that maximizes the posterior probability according to Eq. (2).

It may happen that some prefix given by the user can not be exactly found in the word-graph. In this case, an error-correcting parsing procedure is carried out. Rather than looking for the exact validated prefix $\mathbf{p}$, a best-match, "approximate" prefix is searched for over all the possibles prefixes existing in the WG [16]. This approximate search procedure can be efficiently carried out using dynamic programming and it can be further improved by visiting the states in WG in topological order [4].

This process is repeated until a complete and correct transcript of the input image is obtained. A key point of this interactive process is that, at each user-system interaction, the system can take advantage of the prefix validated so far to attempt to improve its prediction.

The computational costs of these WG procedures can be divided into two phases: *initialization* and *prediction*. For each line image, first the corresponding WG must be stored in memory and several data needed at each successive interaction step can be pre-computed. Then, at each interaction step, the cost of prefix matching and suffix prediction should be considered. Both costs can be seen to be roughly linear in the number of WG edges but, thanks to the pre-computation phase, the more critical suffix prediction costs can be reduced very significantly. As will be shown later, for reasonably small WG sizes, both of these costs can be kept sufficiently small, as required by the real-time constraints imposed by interactive operation.

3.2 WG-based Handwritten Image KWS

The WG-based KWS approach presented here is *"query by string"* and *line-based*. The goal is to determine whether a textually given keyword is likely to appear in each text line image, no matter how many occurrences of the word may appear in the line. According to [24,25], an adequate line-level measure $S(v,\mathbf{x})$ to score how likely is that a keyword v appears in a line image, without considering any specific position within the image, is:

$$S(v,\mathbf{x}) \overset{\text{def}}{=} \max_i P(v \mid i,\mathbf{x}) \tag{3}$$

where $\mathbf{x}$ is the given vector sequence representation of the image and i is an index or "frame" of $\mathbf{x}$.

$P(v \mid i,\mathbf{x})$, is the probability that the word v appears in some segment of the line image $\mathbf{x}$ such that i lies within this segment. As shown in [24], this probability can be easily and efficiently computed by using WGs. More specifically, it is obtained by considering the contribution of all the WG edges labelled with v, which correspond to segmentation hypotheses that include the frame i; that is:

$$P(v \mid i, \mathbf{x}) \;\approx \sum_{\substack{(v',v)\in E:\\ v=\omega(v',v),\\ t(v')<i\leq t(v)}} \varphi(v',v)\,, \qquad \varphi(v',v) = \frac{\alpha(v')\cdot s(v',v)\cdot\beta(v)}{\beta(v_I)} \tag{4}$$

where the so called "edge posterior" probability $\varphi(v',v)$ is computed using the *forward* $\alpha(.)$ and *backward* $\beta(.)$ accumulated path scores which, in turn, can be very efficiently computed on the WGs by dynamic programming [28,24]. Fig. 3 shows a version of the WG shown in Fig. 2 where edge scores are *"normalized"* in this way.

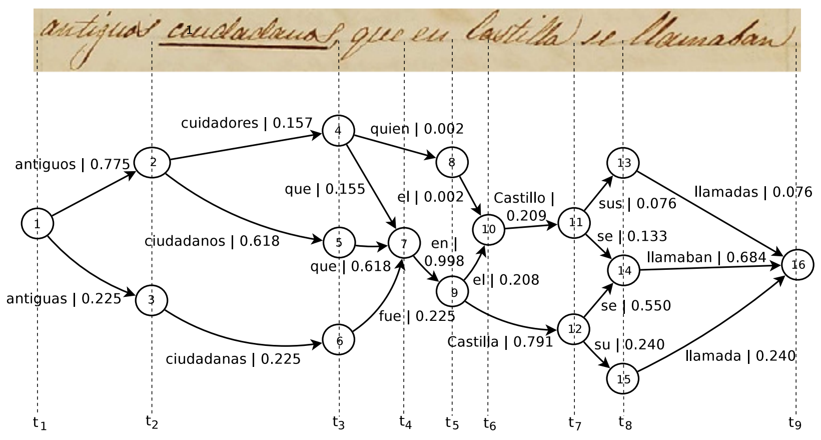

Fig. 3 Example of an "edge posterior" *normalized* WG. The original, unnormalized WG is shown in Fig. 2. Each edge (v',v) is labelled with the word $\omega(v',v)$, and weighted with the edge posterior $\varphi(v',v)$. Note that, for any horizontal image position i, the sum of the weights of all the edges encompassing i is 1.

The costs entailed by the computation of the confidence measure $S(v,\mathbf{x})$, based on the frame word-posteriors $P(v \mid i,\mathbf{x})$ and the corresponding WG normalization (Eqs. (3,4)), depend linearly on the total number of WG edges and on the length, n, of line image representation, $\mathbf{x}$. As will be see in Sec. 4.4, these costs are practically negligible in comparison with the cost of WG generation.

4 Experiments

To compare the performance of WG-based CATTI and KWS for different WG sizes, several experiments were carried out. The evaluation measures, corpora, experimental setup and the results are presented next.

4.1 Evaluation Measures

WG sizes effect on CATTI and KWS performances are assessed in terms of effectiveness (accuracy) and efficiency (computational time and space requirements).

To asses the effectiveness of CATTI we use the *word stroke ratio* (WSR), defined as the number of word-level interaction steps needed to achieve the reference transcript of the text image considered, divided by the total number of reference words. The WSR gives an estimation of the human effort required to produce correct transcriptions using the CATTI system.

On the other hand, KWS effectiveness was measured by means of the standard *recall* and *interpolated precision* [7] curve, which are obtained by varying a threshold to decide whether a score $S(v,\mathbf{x})$ (Eq. (3)) is high enough to assume that a word v appears in $\mathbf{x}$. More specifically, the *average precision* (AP) [32,14] was used as a single scalar performance measure. In addition, we provide the *maximum recall* achieved (MxRcl), defined as the recall value which can be achieved if the decision threshold is set to zero.

Finally, the computing times required for efficiency assessment are reported in terms of total *elapsed* times needed using a dedicated single core of a 64-bit Intel Core Quad computer running at 2.83GHz.

4.2 Corpora

Two historical manuscripts: *CS* [15] and PAR[2] [3] were used in the experiments.

CS is a XIX century Spanish manuscript, entitled *"Noticia histórica de las fiestas con que Valencia celebró el siglo sexto de la venida a esta capital de la milagrosa imagen del Salvador"* (referred to as "Cristo Salvador"), which was kindly provided by the *Biblioteca Valenciana Digital* (BiVaLDi)[3]. It is composed of 50 color images of text pages, written by a single writer and scanned at 300 dpi. Some page examples are shown in Fig. 4.

On the other hand, PAR is a XIII-century epic poem, by Wolfram von Eschenbach, identified as *"St. Gall, collegiate library, cod. 857"* (and often referred to as "Parzival"). It is composed by 47 pages written in the Middle High German language. While written by multiple hands, all the writing styles are very similar. Fig.5 show some page examples of this manuscript.

Tab. 1 summarizes information of data partitioning used for both datasets. The percentage of different words of the test partition that do not appear in the training partition is shown in the row "Running OOV(%)" (out of vocabulary words).

For the KWS application evaluation, it was followed the procedure described in [25,23] for keyword selection, where the whole *training* vocabulary was used as keyword sets: 2 236 for CS and 3 221 for PAR respectively.

[2] CS and PAR are publicly available for research purposes from `prhlt.iti.upv.es/page/data` and `www.iam.unibe.ch/fki/databases`, respectively.

[3] `http://bv2.gva.es`

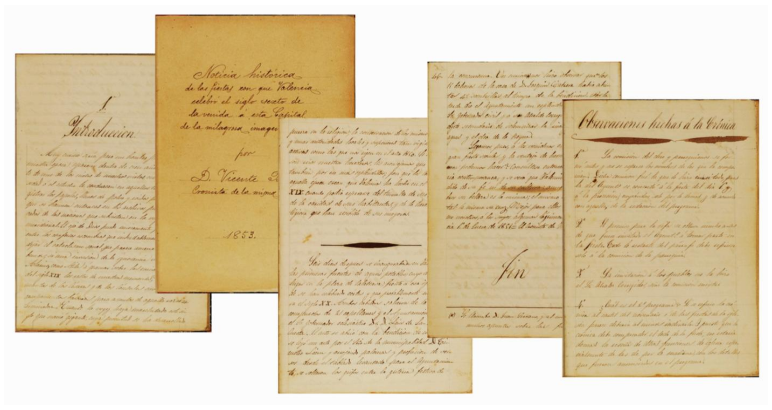
Fig. 4 Examples of page images from the CS manuscript.

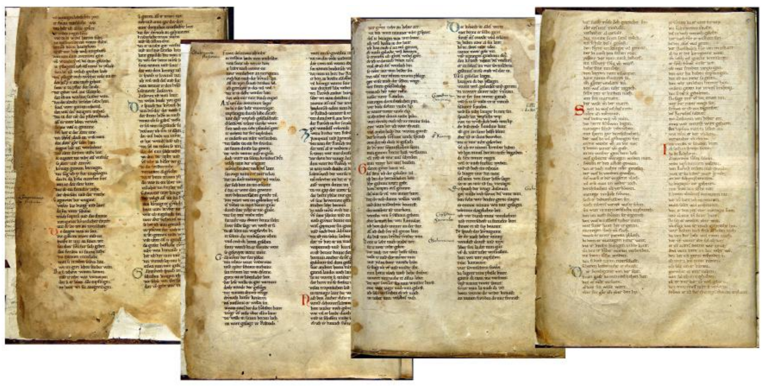
Fig. 5 Examples of page images from the PAR manuscript.

Table 1 Basic statistics of the CS and PAR datasets and the corresponding partitions. OOV stands for Out-Of-Vocabulary words.

	CS			PAR			
	Training	Test	Total	Training	Valid	Test	Total
Running Chars	35 863	26 353	62 216	64 436	26 211	38 339	128 986
Running Words	6 223	4 637	10 860	14 042	5 671	8 407	28 120
Running OOV(%)	0	29.0	–	0	14.6	12.4	–
Number of Lines	675	497	1 172	2 237	912	1 328	4 477
Number of Pages	29	21	50	–	–	–	47
Character Set Size	78	78	78	90	80	82	96
Word Lexicon Size	2 236	1 671	3 287	3 221	1 753	2 305	4 936

4.3 System Setup

Each line image was represented as a sequence of feature vectors. For CS an approach based on smoothed grey levels and grey-level gradients was used (see [16] and Fig. 1), while a technique just based on grey level PCA analysis [11] was adopted for PAR.

The line image feature-vector sequences of both the CS and PAR training partitions were used to train corresponding character HMMs, using the standard embedded Baum-Welch training algorithm [5]. A left-to-right HMM was trained for each of the elements appearing in the training text images (78 for CS and 92 for PAR). This included lowercase and uppercase characters, symbols, special abbreviations, possible spacing between words and characters, crossed-words, etc. Meta-parameters of both HTR feature extractions and HMM models were optimized through cross-validation on the training data for CS and on the validation data for PAR. The optimal HMM meta-parameters were 14 states with 16 Gaussian densities per state for CS, and 8 states with 16 Gaussians per state for PAR.

The training set transcripts of both corpora were used to train the respective 2-grams with Kneser-Ney back-off smoothing [6] (for the PAR final evaluation, the language model training includes also the validation data).

For each test line image, six WGs were obtained for several input degree (IDG) values using the HTR systems [16] based on the previously trained HMMs and 2-grams. The following IDG values were considered: 1, 3, 5, 10, 20 and 40, where the value 1 corresponds to a degenerate WG representing only the 1-best transcript. Tab. 2 shows relevant statistics of the resulting WGs, along with the *minimum word error rates* ($W(\%)$) [2] and the average generation computing time (T_{gen}). The $W(\%)$ values are "Oracle" WERs obtained by computing, for each WG, a path (word sequence) which best matches the corresponding reference transcript,

Table 2 Statistics of the CS and PAR WGs obtained for different IDG values. All the figures are numbers of elements, averaged over all the generated WGs, with exception of the *minimum word error rates* (W) which are percentages. T_{gen} stands for the average WG generation time in minutes.

	CS					PAR				
IDG	Nodes	Edges	Words	$W(\%)$	T_{gen}	Nodes	Edges	Words	$W(\%)$	T_{gen}
1	13	12	12	48.9	1.2	10	9	9	23.1	0.5
3	66	182	31	40.9	3.2	38	102	22	18.8	1.5
5	175	796	57	38.7	4.2	80	346	38	16.3	2.0
10	670	6 008	128	36.7	7.1	224	1 831	77	15.1	3.8
20	2 416	42 990	279	34.9	15.0	618	9 751	153	14.3	6.8
40	7 600	272 850	530	33.6	36.3	1 643	51 951	297	13.6	16.5

Once the WGs were generated, they were directly used by CATTI to complete the prefixes accepted by the (simulated) user. In each interaction step, the decoder parsed the validated prefix over the WG and then continued searching for a suffix which maximizes the posterior probability according to Eq. (2).

For KWS, the WGs were normalized by computing edge posteriors and used to obtain the frame-level word posterior probability according to Eq. (4). Finally, word confidence scores were computed from these probabilities according to Eq. (3).

4.4 Results and Discussion

Experiments with the WG-based systems outlined in Sec.3 were carried out for increasingly large WGs, as described in Sec. 4.3. For increasing IDG values, the corresponding CATTI WSR, along with the average WG load and interaction times (T_{wld} and T_{int} in milliseconds) are reported in Tab. 3. T_{wld} not only includes the time required to load the word-graph, but also the time required to initialize the data structures for error correcting parsing and efficient suffix search. T_{int}, on the other hand, corresponds to the time required to compute a suffix prediction, as needed in each successive word-level interaction step.

Table 3 CATTI WSR for different IDG values, along with average WG load and interaction times: T_{wld} and T_{int} (milliseconds). WSR 95% confidence intervals are below $\pm 3\%$ in CS and $\pm 1\%$ in PAR.

IDG	CS			PAR		
	WSR	T_{wld}	T_{int}	WSR	T_{wld}	T_{int}
1	48.9	0.01	–	23.1	0.01	–
3	44.3	2	0.4	20.1	2	0.3
5	43.7	7	1	19.7	5	0.8
10	43.4	46	10	19.3	20	5
20	43.3	338	67	18.9	104	29
40	43.3	2228	459	18.9	565	169

For KWS, on the other hand, Tab. 4 shows the *Average Precision* (AP) and maximum achieved recall (MxRcl) along with the average normalization time (T_{nor}, in milliseconds) and total indexing time (T_{ind}, in seconds), for increasing IDG. Here T_{nor} includes the values of T_{wld} reported in Tab. 3 plus the average time needed for WG normalization and computation of the KWS scores according to Eqs. (3-4). T_{ind}, on the other hand, is determined by adding T_{nor} to the corresponding WG generation time, T_{gen}, given in Tab. 2.

Table 4 KWS AP and MxRcl along with the average normalization and total indexing times per line: T_{nor} (in milliseconds) and T_{ind} (in seconds). 95% confidence intervals are below ± 0.03 or ± 0.02 in CS and ± 0.01 or ± 0.003 in PAR, for AP or MxRcl, respectively.

IDG	CS				PAR			
	AP	MxRcl	T_{nor}	T_{ind}	AP	MxRcl	T_{nor}	T_{ind}
1	0.430	0.768	0.01	72	0.722	0.897	0.01	30
3	0.699	0.888	1	192	0.878	0.955	1	90
5	0.715	0.918	9	252	0.888	0.967	5	120
10	0.720	0.944	39	426	0.893	0.979	21	228
20	0.722	0.965	310	900	0.894	0.987	190	408
40	0.722	0.973	2100	2180	0.895	0.992	1050	991

From the results, we observe that for WG IDG values larger than 10, the CATTI WSR and the KWS AP do not improve significantly (43.4 or 19.3 WSR and 0.720 or

0.893 AP, for CS or PAR, respectively). It is worth noting that, above IDG $= 10$, the WGs become huge (more than two orders of magnitude larger for IDG $= 40$. On the other hand, in both datasets, the WSR or the AP only degrade less than 2% by using the WGs obtained with IDG $= 5$, which are $5-7$ times smaller on the average. For full comparison, Tabs. 3 and 4 also include results for IDG $= 1$, which is equivalent to just using the HTR 1-best transcription hypothesis. As it can be observed, the effectiveness of both CATTI and KWS degrade very significantly in this case.

It is worth mentioning that the MxRcl is not typically used to assess KWS performance, being AP generally the main criterion. As can be seen in Tab. 4, for IDG equal to or larger than 5, the MxRcl is already large enough to not have significant influence on the AP achieved. Accordingly, the observed tendency of MxRcl to diminish with the IDG should not be considered problematic.

With respect to efficiency, the computing time results of Tabs. 2, 3 and 4 clearly show that WG generation dominates all the costs. For KWS, which is intended to process and indexing thousands or millions of page images without any supervision, it is just this WG generation time the one which matters.

In the case of CATTI, usually aimed to semi-automatically transcribe documents with hundreds of pages, WG generation time is much less critical, as it is only spent in a preparatory phase. However, during interactive operation, large WGs may require prohibitively large load time (T_{wld}), which negatively affect the interactive experience in the first interaction step for each line image. And, for the very large WGs (IDG $= 40$), also the successive, word-level interaction steps may become compromised because of the large increase of prediction time (T_{int}), thereby significantly hindering the overall usability of CATTI.

Taking into account this discussion, we conclude that IDG $= 5$ constitutes a very good trade-off between accuracy and computing cost, both for CATTI and KWS.

5 Remarks and Conclusions

Performance of two applications, CATTI and KWS, of handwritten document image processing based on word graphs is studied in this paper. In both applications the word graphs are generated during the decoding process of text line images using optical character HMMs and N-Gram language models. The work presented in this paper focuses on how the performance of these applications is affected by using WGs of increasing sizes, where WG size is controlled by limiting the node maximum input degree during WG generation.

From the reported performance results, no significant differences are observed for WG input degrees equal to or larger than 5. For this input degree, the word graphs are really small, in the order of hundred of edges on the average. Such word graphs not only allow extremely fast computing of CATTI predictions and line-level KWS word confidence scores, but also can themselves be generated with low extra computing cost over the standard Viterbi decoding computing cost.

The estimates reported in the paper can be used to gauge the computational resources that will be needed for performing WG-based CATTI and KWS on large collections of handwritten document images.

Acknowledgments

Work partially supported by the Generalitat Valenciana grant *PrometeoII*/2014/030; by the Spanish MECD as part of the Valorization and I+D+I Resources program of VLC/CAMPUS in the International Excellence Campus program, and under the STraDa (TIN2012-37475-C02-01) research projects; and by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 600707 - *tranScriptorium*.

References

1. Bazzi, I., Schwartz, R., Makhoul, J.: An Omnifont Open-Vocabulary OCR System for English and Arabic. IEEE Transactions on Pattern Analysis and Machine Intelligence **21**(6), 495–504 (1999)
2. Evermann, G.: Minimum word error rate decoding. Ph.D. thesis, Churchill College, University of Cambridge (1999)
3. Fischer, A., Wuthrich, M., Liwicki, M., Frinken, V., Bunke, H., Viehhauser, G., Stolz, M.: Automatic transcription of handwritten medieval documents. In: Virtual Systems and Multimedia, 2009. VSMM '09. 15th International Conference on, pp. 137–142 (2009). DOI 10.1109/VSMM.2009.26
4. J.C.Amengual, E.Vidal: Efficient Error-Corecting Viterbi Parsing. IEEE Trans. on Pattern Analysis and Machine Intelligence **Vol.PAMI-20, No.10**, 1109–1116 (1998)
5. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press (1998)
6. Kneser, R., Ney, H.: Improved backing-off for N-gram language modeling. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP '95), vol. 1, pp. 181–184. IEEE Computer Society, Los Alamitos, CA, USA (1995)
7. Manning, C.D., Raghavan, P., Schutze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
8. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. Computer Speech & Language **16**(1), 69–88 (2002)
9. Odell, J.J., Valtchev, V., Woodland, P.C., Young, S.J.: A one pass decoder design for large vocabulary recognition. In: Proceedings of the Workshop on Human Language Technology, HLT '94, pp. 405–410. Association for Computational Linguistics, Stroudsburg, PA, USA (1994)
10. Oerder, M., Ney, H.: Word graphs: an efficient interface between continuous-speech recognition and language understanding. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 119 –122 (1993). DOI 10.1109/ICASSP.1993.319246
11. Pesch, H., Hamdani, M., Forster, J., Ney, H.: Analysis of preprocessing techniques for latin handwriting recognition. In: ICFHR, pp. 280–284 (2012)
12. Povey, D., Hannemann, M., Boulianne, G., Burget, L., Ghoshal, A., Janda, M., Karafiat, M., Kombrink, S., Motlcek, P., Qian, Y., Riedhammer, K., Vesely, K., Vu, N.T.: Generating exact lattices in the wfst framework. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP) (2012)
13. Rabiner, L.: A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition. Proceedings IEEE **77**, 257–286 (1989)
14. Robertson, S.: A new interpretation of average precision. In: Proc. of the International ACM SIGIR conference on Research and development in information retrieval (SIGIR '08), pp. 689–690. ACM, New York, NY, USA (2008). DOI http://doi.acm.org/10.1145/1390334.1390453
15. Romero, V., Toselli, A.H., Rodríguez, L., Vidal, E.: Computer assisted transcription for ancient text images. In: Proc of the International Conference on Image Analysis and Recognition, *LNCS*, vol. 4633, pp. 1182–1193. Canada (2007)
16. Romero, V., Toselli, A.H., Vidal, E.: Multimodal Interactive Handwritten Text Transcription. Series in Machine Perception and Artificial Intelligence (MPAI). World Scientific Publishing (2012)
17. Sánchez, J., Mühlberger, G., Gatos, B., Schofield, P., Depuydt, K., Davis, R., Vidal, E., de Does, J.: tranScriptorium: an European project on handwritten text recognition. In: DocEng, pp. 227–228 (2013)
18. Strm, N.: Generation and Minimization of Word Graphs in Continuous Speech Recognition. In: Proc. IEEE Workshop on ASR'95, pp. 125–126. Snowbird, Utah (1995)

19. Toselli, A., Romero, V., i Gadea, M.P., Vidal, E.: Multimodal Interactive Transcription of Text Images. Pattern Recognition **43**(5), 1814–1825 (2010)
20. Toselli, A., Romero, V., Vidal, E.: Word-graph based applications for handwriting documents: Impact of word-graph size on their performances. In: R. Paredes, J.S. Cardoso, X.M. Pardo (eds.) Pattern Recognition and Image Analysis, *Lecture Notes in Computer Science*, vol. 9117, pp. 253–261. Springer International Publishing (2015)
21. Toselli, A.H., Juan, A., Keysers, D., Gonzlez, J., Salvador, I., H. Ney, Vidal, E., Casacuberta, F.: Integrated Handwriting Recognition and Interpretation using Finite-State Models. Internationa Journal of Pattern Recognition and Artificial Intelligence **18**(4), 519–539 (2004)
22. Toselli, A.H., Romero, V., Rodríguez, L., Vidal, E.: Computer Assisted Transcription of Handwritten Text. In: 9th Int. Conf. on Doc. Analysis and Recog. (ICDAR 2007), pp. 944–948. IEEE Computer Society, Curitiba, Paraná (Brazil) (2007)
23. Toselli, A.H., Vidal, E.: Fast HMM-Filler approach for Key Word Spotting in Handwritten Documents. In: Proc. of the 12th International Conference on Document Analysis and Recognition (ICDAR'13). IEEE Computer Society, Washington, DC, USA (2013)
24. Toselli, A.H., Vidal, E., Romero, V., Frinken, V.: Word-graph based keyword spotting and indexing of handwritten document images. Tech. rep., Universitat Politècnica de València (2013)
25. Toselli, A.H., Vidal, E., Romero, V., Frinken, V.: Hmm word-graph based keyword spotting in handwritten document images (2015). Submitted to International Journal of Information Sciences
26. Ueffing, N., Ney, H.: Word-Level Confidence Estimation for Machine Translation. Computatinal Linguistic **33**(1), 9–40 (2007). DOI 10.1162/coli.2007.33.1.9
27. Vinciarelli, A., Bengio, S., Bunke, H.: Off-line recognition of unconstrained handwritten texts using HMMs and statistical language models. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(6), 709–720 (2004)
28. Wessel, F., Schluter, R., Macherey, K., Ney, H.: Confidence measures for large vocabulary continuous speech recognition. IEEE Transactions on Speech and Audio Processing **9**(3), 288–298 (2001)
29. Winograd, T.: Language as a cognitive process,, vol. 1: Syntax. Addison-Wesley (1983)
30. Young, S., Odell, J., Ollason, D., Valtchev, V., Woodland, P.: The HTK Book: Hidden Markov Models Toolkit V2.1. Cambridge Research Laboratory Ltd (1997)
31. Young, S., Russell, N., Thornton, J.: Token passing: a simple conceptual model for connected speech recognition systems. Tech. rep. (1989)
32. Zhu, M.: Recall, Precision and Average Precision. Working Paper 2004-09 Department of Statistics & Actuarial Science - University of Waterloo (2004)