Toselli, AH.; Romero Gómez, V.; Vidal Ruiz, E. (2015). Word-Graph Based Applications for Handwriting Documents: Impact of Word-Graph Size on Their Performances. En Pattern Recognition and Image Analysis. Springer. 253-261. doi:10.1007/978-3-319-19390-8_29.

Word-Graph based Applications for Handwriting Documents: Impact of Word-Graph Size on their Performances

Alejandro H. Toselli, Verónica Romero, and Enrique Vidal

PRHLT Research Centre, Universitat Politècnica de València,
Camino de Vera, s/n - 46022 Valencia - Spain
[ahector, vromero, evidal]@prhlt.upv.es

Abstract. Computer Assisted Transcription of Text Images (CATTI) and Key-Word Spotting (KWS) applications aim at transcribing and indexing handwritten documents respectively. They both are approached by means of Word Graphs (WG) obtained using segmentation-free handwritten text recognition technology based on N-gram Language Models and Hidden Markov Models. A large WG contains most of the relevant information of the original text (line) image needed for CATTI and KWS but, if it is too large, the computational cost of generating and using it can become unaffordable. Conversely, if it is too small, relevant information may be lost, leading to a reduction of CATTI/KWS in performance accuracy. We study the trade-off between WG size and CATTI & KWS performance in terms of effectiveness and efficiency. Results show that small, computationally cheap WGs can be used without loosing the excellent CATTI/KWS performance achieved with huge WGs.

1 Introduction

In recent years, large quantities of historical handwritten documents are being scanned into digital images, which are then made available through web sites of libraries and archives all over the world. Despite this, the wealth of information conveyed by the text captured in these images remains largely inaccessible (no plain text, difficult to read even for researchers). Given the amount of text, automated methods are needed to allow the users to transcribe and/or search, and also to add value to mass-digitisation and preservation efforts of Culture Heritage institutions. To this end, the *tranScriptorium*[1] project aims at fulfil these requirements with the development of two different applications: the *Computer Assisted Transcription for Test Images* (CATTI) [1], intended to speed up transcription process, and the *KeyWord Spotting* [2] for automatic indexing of untranscribed handwritten material under the so called *Precision-Recall trade-off model*. Actually, both applications rely on what is called *word lattice* or *Word Graph* (WG), which are previouly produced by a standard HMM-based handwritten text recognition system. It is worth noting that in their own conception,

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

CATTI and KWS approaches do not require the use of WGs for performing corresponding tasks. However, usually large response times due to running CATTI and KWS without WGs would rather affect their usabilities.

A WG is a data structure proposed by several authors some decades ago during the development of Automatic Speech Recognition (ASR) technology [3]. They are generated through a natural extension of the standard dynamic programming Viterbi decoding algorithm, which determines the single best HTR hypothesis. 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 some loss of the information retained in the resulting WGs [3].

This paper studies how different sizes of WGs pruned by different IDG values impact on the effectiveness/efficiency performance 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 material using CATTI and WG-based KWS.

2 Overview of HTR and WGs Technology

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

2.1 HTR based on HMMs and N-Grams

The employed HTR technology is based on *Hidden Markov Models* (HMMs) and N-grams, and follows the fundamentals presented in [4]. This recognizer accepts a handwritten text line image, represented as a sequence of feature vectors $\mathbf{x}$, and find a most likely word sequence $\widehat{\mathbf{w}}$ according to: $\widehat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathbf{x} \mid \mathbf{w}) \cdot P(\mathbf{w})$. The conditional density $p(\mathbf{x} \mid \mathbf{w})$ is approximated by morphological word models, built by concatenating character HMMs [5],and the prior $P(\mathbf{w})$ is approximated by an N-gram language model [5].

The search (or decoding) of $\widehat{\mathbf{w}}$ is optimally carried out by using the Viterbi algorithm [5]. Moreover, rather than obtaining just a single best solution (i.e. $\widehat{\mathbf{w}}$) a huge set of best solutions can be obtained in the form of a WG as a byproduct of this decoding process. For a more detailed description of this HTR system, including text line processing, model training and decoding, refers to [1].

2.2 Word-Graphs

A WG represents very efficiently a huge number word sequence hypotheses whose posterior probabilities are large enough, according to morphological character likelihood and prior (language) models, used to decode a text line image, represented as a sequence of feature vectors. WGs also store additional important data about these hypotheses; namely, alternative word segmentations and word decoding likelihoods.

Formally, a WG is represented as a weighted directed acyclic graph whose edges are labelled with words and weighted with scores derived from the HMM (likelihood) and N-gram (prior) probabilities. 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 n is 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 $s(\nu', \nu)$ is its score, corresponding to the likelihood that the word v appears in the *image segment* delimited by frames $t(\nu') + 1$ and $t(\nu)$.

A *complete path* of a WG is a sequence of nodes starting with node ν_I and ending with a node in F. Complete paths correspond to whole line decoding hypotheses. WGs considered in this work are unambiguous; that is, no two complete paths exist in a WG which correspond to the same sequence of words.

2.3 Complexity Cost

It is well known that the computational complexity of the Viterbi algorithm is linear with 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 size, according to [6]).

On the other hand, *error correcting parsing* carried out by CATTI on a WG (see [1]) as well as the WG normalization process required by KWS, entails an extra computational cost which mainly depends on the total number of WG edges. However, according to [7], as well as to our own observations, this cost is negligible; typically less than 1% of the total cost required for generating a large WG.

3 Outline of WG-based CATTI and KWS Applications

3.1 WG-based CATTI Application

The CATTI system is presented in detail in [1]. In the CATTI framework, the human transcriber is directly involved in the transcription process since he/she is responsible of validating and/or correcting the HTR output.

The interactive transcription 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 introduces some amendment to correct the erroneous text that follows the validated prefix, producing a new prefix $\mathbf{p}$. At this point, the system takes into account the new prefix and tries to complete it by searching for a most likely suffix, $\hat{\mathbf{s}}$, according to:

$$\hat{\mathbf{s}} = \arg\max_{s} P(\mathbf{s} \mid \mathbf{x}, \mathbf{p}) \approx \arg\max_{s} p(\mathbf{x} \mid \mathbf{p}, \mathbf{s}) \cdot P(\mathbf{s} \mid \mathbf{p}) \tag{1}$$

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 conditioned by $\mathbf{p}$. The main difference is that now $\mathbf{p}$ is given. Therefore, the search must be performed over all possible suffixes $\mathbf{s}$ of $\mathbf{p}$. This search is carried out using the WGs obtained during the recognition process, achieving a very efficient, linear cost search [1]. This process is repeated until a complete and correct transcript of the input signal $\mathbf{x}$ is reached.

3.2 WG-based Handwritten Image KWS

The WG-based KWS approach presented here is *line-based*. The goal is to determine whether a given keyword is or is not in each text line image, no matter how many occurrences of the word may appear in the line. According to [2], an adequate global line-level measure $S(v, \mathbf{x})$ to score the degree of presence of a keyword v in a text line (represented by its feature vector sequence $\mathbf{x}$), without considering any specific position within the line image, is given by:

$$S(v, \mathbf{x}) \stackrel{\text{def}}{=} \max_{i} P(v \mid i, \mathbf{x}) \tag{2}$$

where the frame-level word posterior, $P(v \mid i, \mathbf{x})$, is the probability that the word v is present in the line image at position i (the index of a feature vector $\mathbf{x}$). In [2] it is shown how this probability can be directly and efficiently computed by using WGs. Specifically, it can be 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:

$$P(v \mid i, \mathbf{x}) \approx \sum_{\substack{(\nu', \nu) \in E: \\ v = \omega(\nu', \nu), \\ t(\nu') < i \leq t(\nu)}} \frac{\alpha(\nu') \cdot s(\nu', \nu) \cdot \beta(\nu)}{\beta(\nu_I)} \tag{3}$$

where $\alpha(.)$ is the *forward* and $\beta(.)$ *backward* accumulated path scores which can be efficiently computed on the WGs by dynamic programming [8, 2].

4 Experiments

To compare the performance of both the CATTI and WG-based KWS approaches 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 (time and space requirements).

To asses the effectiveness of the CATTI system we use the *word stroke ratio* (WSR). It can be defined as the number of word level user interactions necessary to achieve the reference transcription of the text image considered, divided by the total number of reference words. The WSR gives an estimation of the needed human effort to produce correct transcriptions using the CATTI system.

For effectiveness assessment of the KWS approach, we employed the popular scalar measure called *average precision* (AP) [9]. The AP is based on the standard *recall* and *interpolated precision* [10] measures, which are functions of a threshold used to decide whether a score $S(v, \mathbf{x})$ (see (2)) is high enough to assume that a word v is in $\mathbf{x}$. Actually, the AP is defined as the area under the Recall-Precision curve.

On the other hand, 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 Description

Experiments were carried out on two different corpora: "Cristo-Salvador" (CS) [11] and Parzival database (PAR) [12].[2]

CS is a XIX century Spanish manuscript which was kindly provided by the *Biblioteca Valenciana Digital* (BiVaLDi). It is composed of 50 color images text pages, written by a single writer and scanned at 300 dpi.

On the other hand, PAR is a medieval manuscript from the XIII century refered to as *St. Gall, collegiate library, cod. 857*. It is composed by 45 pages, written by multiple authors in the Middle High German language. Tab. 1 summarizes statistical information of data partitioning used for each corpus.

Table 1. Basic statistics of the partition of the CS and PAR databases. 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	29.03	–	0	14.58	12.40	–
# Lines	675	497	1 172	2 237	912	1 328	4 477
Char Lex. Size	78	78	78	90	80	82	96
Word Lex. Size	2 236	1 671	3 287	3 221	1 753	2 305	4 936

[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.

4.3 System Setup

The line images of both the CS and PAR training partitions were used to train corresponding character HMM models 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 82 for PAR), such as lowercase and uppercase letters, symbols, special abbreviations, possible spacing between words and characters, crossed-words, etc. Meta-parameters of the HTR feature extraction modules for CS [1] and PAR [13], as well as corresponding HMM models, were optimized through cross-validation on the CS training data and on the PAR validation data, respectively. The optimal number of states per HMM for CS was 14 with 16 Gaussian densities per state, and 8 states with 16 Gaussians per state for PAR.

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

For each line image of the test partition, five WGs were obtained for the following input degree values: 3, 5, 10, 20 and 40. All these WGs were generated using the HTR system [1] with the previously trained models (HMMs and 2-grams). Tab. 2 shows some statistics of the resulting WGs for different maximum input degrees (IDG), along with the *minimum word error rates* ($W(\%)$) [14] and their average computing time generation (T_{gen}).

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* ($W(\%)$) values. T_{gen} stands for the average WG generation time (minutes). Standard deviation for Nodes, Edges and Words are below $\pm 25\%$ and $\pm 35\%$ of their average values for CS and PAR respectively.

	CS					PAR				
IDG	Nodes	Edges	Words	$W(\%)$	T_{gen}	Nodes	Edges	Words	$W(\%)$	T_{gen}
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

The $W(\%)$ is a *goodness* property of WGs. For each WG, it has been computed measuring, by dynamic programming [14], the minimum edit distance from the reference to a word sequence hypothesis in the WG.

For CATTI, once the WGs were generated, the system used them directly to complete the prefixes accepted by the user. For each interaction, the decoder parsed the validated prefix $\mathbf{p}$ over the WG and then continued searching for the suffix $\mathbf{s}$ that maximizes the posterior probability according to Eq. (1).

For KWS, the WGs were normalized and the frame-level word posterior probabilities $P(v \mid i, \mathbf{x})$ were computed according to Eq. (3). Finally, word confidence scores, $S(v, \mathbf{x})$, were computed from these probabilities as described in Eq. (2).

4.4 Results and Discussion

Experiments with the WG-based application approaches described in Sec.3 were carried out for the increasingly large WGs described in Sec. 4.3. For CATTI application, in each of these five trials, the respective *word stroke ratio* (WSR) along with the average time required to load the WG in the system and the average interaction time (T_{wld} and T_{int} in seconds) are reported in Table 3. In addition, for the KWS application are shown the *Average Precision* (AP) and the average indexing time (T_{ind} in seconds).

Table 3. Left: CATTI WSR for different IDG values along with the average WG load time and word interaction times: T_{wld} and T_{int} (seconds). Right: KWS figure AP along with the average indexing time: T_{ind} (seconds). 95% confidence intervals are below ±1% in PAR and ±3% in CS for WSR, and below ±0.01 in PAR and ±0.03 in CS for AP.

	CATTI CS			CATTI PAR			KWS CS		KWS PAR	
IDG	WSR	T_{wld}	T_{int}	WSR	T_{wld}	T_{int}	AP	T_{ind}	AP	T_{ind}
3	44.3	$2\ 10^{-3}$	$4\ 10^{-4}$	20.05	$2\ 10^{-3}$	$3\ 10^{-4}$	0.699	10^{-5}	0.878	10^{-5}
5	43.7	0.007	10^{-3}	19.7	0.005	$8\ 10^{-4}$	0.715	$8.7\ 10^{-4}$	0.888	$2.2\ 10^{-4}$
10	43.4	0.046	10^{-2}	19.3	0.02	0.005	0.720	$3.9\ 10^{-2}$	0.893	$2.1\ 10^{-2}$
20	43.3	0.338	0.067	18.9	0.104	0.029	0.722	$3.1\ 10^{-1}$	0.894	$1.9\ 10^{-1}$
40	43.3	2.228	0.459	18.9	0.565	0.169	0.722	2.1	0.895	1.05

From the results, we observe that for WG input degree values larger than 10, the WSR (43.4/19.3 CS/PAR) of CATTI and the AP (0.720/0.893 CS/PAR) of KWS do not improve significantly. On the other hand, the WSR/AP only rises/drops about less than 2% by using very much smaller WGs (corresponding to an input degree of 5, with around less than 6 times the number of WG edges on average. As a result, the overall WG generation costs required using these WGs become orders of magnitude lower than the costs incurred using the WGs with largest input degree (see Tables 2 and 3).

Therefore, we conclude that an *Indeg* of 5 constitutes a very good trade-off between CATTI and KWS accuracy and computing cost.

5 Remarks and Conclusions

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

From the reported WSR and AP figures, 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 nodes and edges on the average. Such word graphs not only allow extremely fast computing of

the CATTI interactive process and the required line-level KWS word confidence scores, but also can themselves be generated with low extra computing cost with respect to the standard Viterbi decoding computing cost.

These estimates can be used to gauge the computational resources that will be needed for performing CATTI and WG-based KWS on massive collections of handwritten document images.

6 Acknowledgments

Work partially supported by the Spanish MICINN projects *STraDA* (TIN2012-37475-C02-01) and by the EU 7th FP *tranScriptorium* project (Ref: 600707).

References

1. Romero, V., Toselli, A.H., Vidal, E.: Multimodal Interactive Handwritten Text Transcription. Series in Machine Perception and Artificial Intelligence (MPAI). World Scientific Publishing (2012)
2. Toselli, A.H., Vidal, E., Romero, V., Frinken, V.: Word-graph based keyword spotting and indexing of handwritten document images. Technical report, Universitat Politècnica de València (2013)
3. 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. Volume 2. (april 1993) 119 –122
4. 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) (1999) 495–504
5. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press (1998)
6. Ström, N.: Generation and Minimization of Word Graphs in Continuous Speech Recognition. In: Proc. IEEE Workshop on ASR'95, Snowbird, Utah (1995) 125–126
7. Ortmanns, S., Ney, H., Aubert, X.: A word graph algorithm for large vocabulary continuous speech recognition. Computer Speech and Language **11**(1) (1997) 43–72
8. 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) (mar 2001) 288–298
9. 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), New York, NY, USA, ACM (2008) 689–690
10. Manning, C.D., Raghavan, P., Schutze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
11. 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. Volume 4633 of LNCS., Canada (2007) 1182–1193
12. 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. (2009) 137–142
13. Pesch, H., Hamdani, M., Forster, J., Ney, H.: Analysis of preprocessing techniques for latin handwriting recognition. In: ICFHR. (2012) 280–284
14. Evermann, G.: Minimum Word Error Rate Decoding. PhD thesis, Churchill College, University of Cambridge (1999)