An Experimental Study of Pruning Techniques in Handwritten Text Recognition Systems*

Daniel Martín-Albo, Verónica Romero, and Enrique Vidal

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València, Spain
{dmartinalbo,vromero,evidal}@dsic.upv.es
http://prhlt.iti.upv.es/

Abstract. Handwritten Text Recognition is a problem that has gained attention in the last years mainly due to the interest in the transcription of historical documents. However, the automatic transcription of handwritten documents is not error free and human intervention is typically needed to correct the results of such systems. This interactive scenario demands real-time response. In this paper, we present a study comparing how different pruning techniques affect the performance of two freely available decoding systems, HTK and iATROS. These two systems are based on Hidden Markov Models and n-gram language models. However, while HTK only considers 2-gram language models, iATROS works with n-grams of any order. In this paper, we also carried out a study about how the use of n-grams of size greater than two can enhance results over 2-grams. Experiments are reported with the publicly available ESPOS-ALLES database.

Keywords: Handwritten Text Recognition, pruning techniques, language models, n-grams.

1 Introduction

Lately, the paradigm for Pattern Recognition (PR) has been shifting from fully automatic systems to systems where the user interacts with the system to obtain the final result [11]. One remarkable pattern recognition example where this interaction can be successfully used is in handwritten document transcription [10]. This task is becoming an important research topic, specially because of the increasing number of digital libraries publishing large quantities of digitized legacy documents.

Given that in this scenario the user is constantly interacting with the system, response within strict time constraints is needed. Long delays can be a cause of

* This work was partially supported by the Spanish MEC under the STraDA research project (TIN2012-37475-C02-01), the MITTRAL (TIN2009-14633-C03-01) project, the FPU scholarship AP2010-0575, by the Generalitat Valenciana under the grant Prometeo/2009/014, and through the EU 7th Framework Programme grant tranScriptorium (Ref: 600707).

J.M. Sanches, L. Micó, and J.S. Cardoso (Eds.): IbPRIA 2013, LNCS 7887, pp. 559–566, 2013.

major frustration leading the user to believe that the system is not functioning, or that an input has been ignored. Responsiveness is therefore considered an essential usability issue for computer-human interaction.

Most of the time spent by the recognition systems is used in seeking for the optimal decoding in a search network. Thus, one way to improve the responsiveness of a decoding system is to reduce the size of this network by pruning parts that provide little information.

In this work we studied how pruning techniques affects the performance of two freely available decoding systems, HTK [12] and iATROS [4]. HTK was developed at the Cambridge University Engineering Department (CUED). On the other hand, iATROS was developed by the Pattern Recognition and Human Language Technology group (PRHLT) of the Universitat Politècnica de València (UPV). Both systems are based on Hidden Markov Models (HMMs) [1] and n-gram language models [1]. However, while HTK only considers 2-grams language models, iATROS works with n-grams of any order. Furthermore, we also carried out a study about how the use of n-grams of size greater than two can enhance results over 2-grams. Experiments are reported with the publicly available ESPOSALLES database [7].

2 General Formulation of the Decoding Problem

The traditional decoding problem can be formulated as the problem of finding the most likely word sequence, $\hat{\mathbf{w}} = (w_1 w_2 \ldots w_l)$, for the given handwritten text image represented by a sequence of any number of feature vectors $\mathbf{x} = (x_1 x_2 \ldots x_D)$, i.e., $\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \Pr(\mathbf{w} \mid \mathbf{x})$. Using the Bayes' rule, we can decompose $\Pr(\mathbf{w} \mid \mathbf{x})$ into two probabilities, $\Pr(\mathbf{x} \mid \mathbf{w})$ and $\Pr(\mathbf{w})$, representing morphological-lexical and syntactic knowledge, respectively:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \Pr(\mathbf{w} \mid \mathbf{x}) = \arg\max_{\mathbf{w}} \Pr(\mathbf{w}) \cdot \Pr(\mathbf{x} \mid \mathbf{w}) \tag{1}$$

The linguistic grammar knowledge $\Pr(\mathbf{w})$, is typically modelled by a n-gram language model [1]. On the other hand, $\Pr(\mathbf{x} \mid \mathbf{w})$ is typically approximated by concatenated character models, usually Hidden Markov Models (HMMs) [1].

Each character class is modelled by a continuous density left-to-right HMM, characterized by a set of states and a Gaussian mixture per state. Each lexical word is modelled by a stochastic finite-state automaton (SFS), which represents all possible concatenations of individual characters to compose a word. By embedding the character HMMs into the edges of this automaton, a *lexical* HMM is obtained. Finally, text lines are modelled using n-grams.

Given that all these models (HMM character, word and line) can be represented by SFS networks, they can be easily integrated into a single global SFS network by replacing each word character of the n-gram model by the corresponding HMM. The search, involved in (1) to decode the sequence $\mathbf{x}$ into the most likely output $\hat{\mathbf{w}}$, is performed over this global SFS, which leads to:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \Pr(\mathbf{w}) \cdot \sum_{\mathbf{s}} \Pr(\mathbf{x}, \mathbf{s} \mid \mathbf{w}) \tag{2}$$

where $\mathbf{s}$ is any HMM state sequence being emitted by the word sequence $\mathbf{w}$. It should be noted that the maximization problem stated in (2) is NP-hard. Nevertheless, this search problem can be approximated by the Viterbi algorithm [1]:

$$\hat{\mathbf{w}} \approx \arg\max_{\mathbf{w}} \Pr(\mathbf{w}) \cdot \max_{\mathbf{s}} \Pr(\mathbf{x}, \mathbf{s} \mid \mathbf{w}) \tag{3}$$

However, the time complexity of the Viterbi algorithm grows with the number of vocabulary words. Therefore, an exhaustive search through the network in large vocabulary tasks can be infeasible given the real-time requirements of an interactive system. To make the search feasible, we need to compromise the optimality of the search by introducing pruning.

2.1 Viterbi Pruning Strategies

As explained above, the aim of pruning is to reduce the size of the search network, to only permit the computation of the most promising paths. In this paper, we focus on the pruning techniques that are implement in both recognizers, HTK and iATROS. These are the beam pruning [5] and histogram pruning [8].

Beam pruning is probably the most important pruning criteria. It is often referred as global beam search pruning, as it can be applied to all states of the search network. This heuristic retains only those paths whose likelihood score is close to the best current hypothesis. This proximity is defined using a threshold called *beam width* (f_B). The value of f_B is predefined and has no semantic interpretation. An improper selection of f_B could conduct to the survival of too many hypotheses (making the pruning process useless) or too few (if all paths that leads to a word-end are pruned this causes a search error).

Histogram pruning is a technique to prevent this slippage in the number of active hypotheses. This pruning method introduces an upper limit (H) to the number of active hypotheses. If the value is larger than H, only the best H hypotheses are retained. This technique is called histogram pruning because in practice a histogram of the scores of the active hypotheses is used.

3 Experimental Details

A comparative set of experiments were conducted using HTK (version 3.4) and iATROS (version 0.1) toolkits. The aim was to compare the performance of these two systems for different parameter values of the aforementioned pruning techniques. The details of the corpora, the preprocessing, feature extraction, the configuration of the models and the assessment measures are given below.

3.1 Corpora

The comparison experiments were performed on the ESPOSALLES[1] database [7]. Here, we used the LICENSES part, which was compiled from a marriage license book conserved at the Archives of the Cathedral of Barcelona.

[1] The corpus is publicly available from: `http://www.cvc.uab.es/5cofm/groundtruth`

The corpus was written by only one person between 1617 and 1619 in old Catalan. The LICENSES part used in our experiments has a total of 173 pages. These pages contain 5,447 lines grouped in 1,747 licenses. The whole manuscript was transcribed line by line by an expert palaeographer. The complete annotation of LICENSES contains around 60,000 running words from a lexicon of around 3,500 different words. More information can be found at [7].

Here we used the standard partition proposed in [7], consisting of seven consecutive blocks of 25 pages. Table 1 summarizes the statistics of the LICENSES part of the ESPOSALLES database.

Table 1. Basic statistics of the different partitions for the LICENSES part of the ESPOSALLES database. The number of running words for each partition that do not appear in the other six partitions is shown in the out-of-vocabulary (OOV) row.

	P0	P1	P2	P3	P4	P5	P6
Pages	25	25	25	25	25	25	23
Lines	827	779	786	768	771	773	743
Run. words	8,893	8,595	8,802	8,502	8,506	8,799	8,610
OOV	426	374	368	340	329	373	317
Lexicon	1,119	1,096	1,106	1,036	1,046	1,078	1,011
Characters	48,464	46,459	47,902	45,728	46,135	47,529	46,012

3.2 Preprocessing and Feature Extraction

In the preprocessing module, the pages were separated into text line images using a method based on the horizontal projection profile of the input image. Then, a conventional noise reduction method was applied on each line image [2]. Finally, slant correction and size normalization were applied. A more detailed description about this preprocessing can be found in [9,6].

Then, each preprocessed line image was represented as a sequence of feature vectors. To do this, the feature extraction module applies a grid to divide each text line image into squared cells. For each cell, three smoothed features are calculated: normalized gray level, the horizontal and vertical components of the grey level gradient. At the end of this process, a sequence of 60-dimensional feature vectors is obtained. More information can be found in [9].

3.3 Models

As mentioned before, each character is modelled by a continuous density left-to-right HMM with six states and 64 Gaussians mixture components per state. These values have been proven to work well in previous handwriting recognition experiments. The HMMs have been trained from line images from the training set, without any kind of segmentation, accompanied by the correct transcription into the corresponding sequence of characters. This training process is carried out using a well-known instance of the EM algorithm called Baum-Welch algorithm.

On the other hand, lines are modelled using 2-grams and 3-grams, with Kneser-Ney back-off smoothing [3], estimated from the training transcriptions of the text images.

3.4 Assessment Measures

Two measures were adopted to assess the performance of both decoders. The quality of the transcriptions was measured by means of the word error rate (WER). It is defined as the minimum number of words that need to be substituted, deleted, or inserted to match the recognition output with the corresponding reference ground truth, divided by the total number of words in the reference transcriptions. However, since we may not have a transcription for every sentence due to searching errors caused by the pruning, we assumed many inserts as necessary to achieve the reference. Moreover, to allow the comparison between both systems all the transcriptions obtained, as well as the reference, were transformed to uppercase. Regarding WER results, confidence intervals were computed using bootstrapping.

The word decoding time (WDT) was used to assess the response time. It is defined as the time in seconds to decode a set of sentences divided by the total number of words in these sentences. WDT is similar to the real time factor (RTF) metric of measuring the speed of an automatic speech recognition system. The RTF is defined as the ratio between the time that takes to process an input and its duration. In this case, it was not possible to use the RTF since we are dealing with text images, instead of voice.

The different nature of WER and WDT makes difficult to achieve a completely accurate comparison when we try to compare two results. However, we established that the best result is the one that minimizes both WER and WDT, but placing emphasis on the WDT. That is, we want a *good* WER in the shortest WDT possible.

4 Experimental Results

The aim of the performed experiments was twofold: 1) test how different values of the previous described pruning techniques can affect the performance of both recognizers; and, 2) check how much affects to the result the use of 3-grams over 2-grams, that is, having a more informed language model. To address these questions we used the seven different partitions described in Sect. 3 aimed at performing cross-validation experiments.

We defined first a baseline scenario, without pruning. HTK allows disabling any pruning by setting to zero that parameter. In contrast, this is not possible in iATROS. For this reason, we used *conservative* pruning values ($H = 10^4$ and $f_B = 10^{30}$) as a baseline scenario in iATROS. Table 2 displays the baseline results for HTK and iATROS.

An examination of Table 2 shows that, comparing 2-grams, iATROS achieves slightly better values of WER. In contrast, HTK scores substantially lower WDT

Table 2. Baseline scenario word error rate (WER) and word decoding time (WDT) for HTK and iATROS (2-grams and 3-grams). WER is expressed in % and WDT in seconds. Intervals at $\alpha = 0.95\%$.

	HTK	iATROS	
	2-grams	2-grams	3-grams
WER	15.9 ± 0.5	15.4 ± 0.4	14.4 ± 0.5
WDT	1.7	4.1	3.8

than iATROS. We think that this is due to the use of more efficient data structures in HTK as it only considers 2-gram language models. Moreover, iATROS using 3-gram language model achieves significantly better WER than the other two options.

To investigate the effect of the pruning techniques on the recognition accuracy and decoding speed, different experiments were conducted. First, an experiment was carried out varying the value of the histogram pruning threshold (H) for each system. Fig. 1 (left) summarizes the results of this comparison using a 2-gram language model. A good compromise between WER and WDT was achieved for HTK, approximately, at H = 1500 by getting a WDT of 0.7s and a WER of 16.5% (60% faster and 4% less accurate than its baseline result). Regarding iATROS, a good value was obtained, approximately, at H = 2000 with a WDT of 1.7s of and a WER of 16.4% (59% faster and 7% less accurate than its baseline). Comparing these two performances, HTK obtained a similar result in WER, being 59% faster than iATROS.

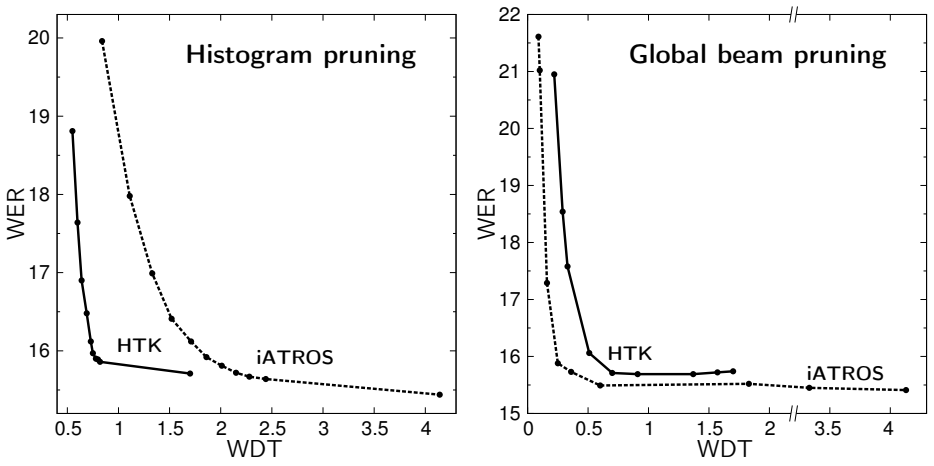

Fig. 1. Word error rate (WER) and word decoding time (WDT) comparison using HTK and iATROS with 2-gram language models. Showing results for different values of histogram pruning threshold (H) (left) and global beam pruning threshold (f_B) (right). The rightmost point for each curve on the abscissa represents the baseline showed on Table 2.

A similar experiment was also conducted varying the value of the beam width (f_B) (Fig. 1 (right)). A good value for HTK was attained, approximately, at $f_B = 1200$. In this case the WDT was 0.5s and the WER was 16.0% (40% faster than its baseline result). Regarding iATROS, a good value was obtained at $f_B = 800$ with a WDT of 0.3s and a WER of 15.9% (94% faster and 3% less accurate than its baseline). Comparing the best configuration of the two systems, both obtained similar WER being iATROS 46% faster than HTK.

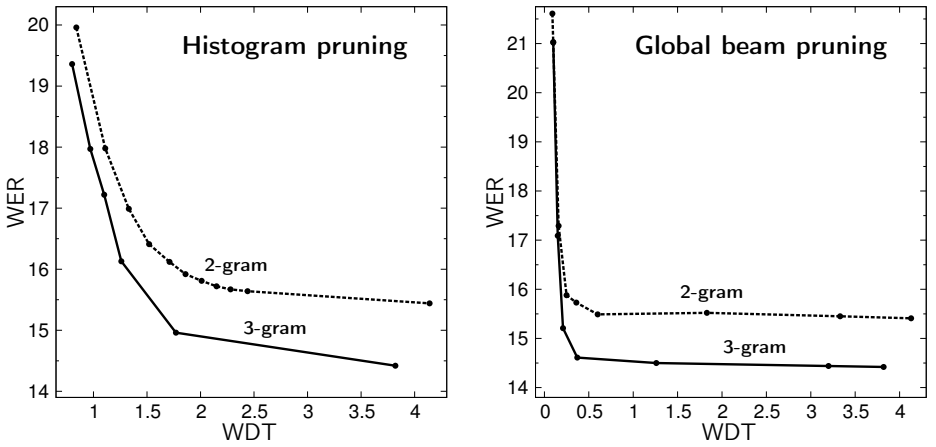

Fig. 2. Performance comparison of iATROS using 2-grams and 3-grams. Word error rate (WER) and word decoding time (WDT) for different values of histogram pruning threshold (M) (left) and for different values of global beam pruning threshold (f_B) (right). The rightmost point for each curve on the abscissa represents the baseline showed on Table 2.

After comparing the performance of the pruning techniques in both HTK and iATROS systems, the following experiment intended to verify how results can be boosted by exploiting the advantages of 3-grams over 2-grams. In Fig. 2 we can see a comparison between iATROS using a 2-gram language model and a 3-gram language model. A good value using histogram pruning was achieved at H = 3000 by getting a WDT of 1.7s and a WER of 15.0%. These values represent an improvement of 8% in WER over iATROS using 2-grams.

Regarding global beam pruning a good value was $f_B = 1000$ with a WDT of 0.4s and a WER of 14.6 %. These values represent a 90% reduction of WDT in comparison with the baseline and an improvement of 8% in WER over iATROS using 2-grams.

Finally, Table 3 summarizes the results obtained for both decoders and types of pruning. As we can see, the differences of the results using 2-grams in iATROS and HTK are not statistical significant. However, iATROS with 3-grams outperforms compared to using iATROS or HTK both with 2-grams.

Table 3. Best Word error rate (WER) and word decoding time (WDT) using HTK and IATROS by employing Histogram and Beam pruning. WER in % and WDT in seconds. Intervals at $\alpha = 0.95\%$.

	HTK (2-grams)		iATROS (2-grams)		iATROS (3-grams)	
	Histogram	Beam	Histogram	Beam	Histogram	Beam
WER	16.5 ± 0.5	16.1 ± 0.4	16.4 ± 0.4	15.9 ± 0.4	15.0 ± 0.4	14.6 ± 0.4
WDT	0.7	0.5	1.7	0.3	1.7	0.4

5 Conclusions

In this paper, we have presented a set of experiments to compare the pruning techniques of HTK and iATROS. We established that HTK and iATROS word error rate (WER) results using histogram pruning are comparable, being HTK faster. On the other hand, iATROS performs much better in terms of speed than HTK using global beam pruning. Moreover, the iATROS decoder has performed quite better when using 3-grams. We recall that iATROS, unlike HTK, allows the use of n-grams of size greater than two.

References

1. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press (1998)
2. Kavallieratou, E., Stamatatos, E.: Improving the quality of degraded document images. In: Proc. of 2nd IEEE Int. Conf. on Document Image Analysis for Libraries, Washington DC, USA, pp. 340–349 (2006)
3. Kneser, R., Ney, H.: Improved backing-off for n-gram language modeling. Proc. of the ICASSP 1995, pp. 181–184 (1995)
4. Luján-Mares, M., Tamarit, V., Alabau, V., Martínez-Hinarejos, C.D.: i Gadea, M.P., Sanchis, A., Toselli, A.H.: iATROS: A speech and handwritting recognition system. In: V Jornadas en Tecnologías del Habla, pp. 75–78 (2008)
5. Ney, H., Mergel, D., Noll, A., Paeseler, A.: Data driven search organization for continuous speech recognition. Trans. Sig. Proc. 40(2), 272–281 (1992)
6. Romero, V., Pastor, M., Toselli, A.H., Vidal, E.: Criteria for handwritten off-line text size normalization. In: Proc. of the 5th Int. Conf. on Visualization, Imaging and Image, Spain (2006)
7. Romero, V., Fornés, A., Serrano, N., Sánchez, J.A., Toselli, A.H., Frinken, V., Vidal, E., Lladós, J.: The ESPOSALLES database: An ancient marriage license corpus for off-line handwriting recognition. Pattern Recognition (in press, 2013)
8. Steinbiss, V., Tran, B.H., Ney, H.: Improvements in beam search. In: ICSLP (1994)
9. Toselli, A.H., et al.: Integrated Handwriting Recognition and Interpretation using FS Models. Int. Journal on Pat. Rec. and Artif. Intel. 18(4), 519–539 (2004)
10. Toselli, A.H., Romero, V., Pastor, M., Vidal, E.: Multimodal interactive transcription of text images. Pattern Recognition 43(5), 1814–1825 (2010)
11. Vidal, E., Rodríguez, L., Casacuberta, F., García-Varea, I.: Interactive pattern recognition. In: Popescu-Belis, A., Renals, S., Bourlard, H. (eds.) MLMI 2007. LNCS, vol. 4892, pp. 60–71. Springer, Heidelberg (2008)
12. Young, S.J., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P.: The HTK Book Version 3.4. Cambridge University Press (2006)