USING SMOOTHED K-TSS LANGUAGE MODELS IN CONTINUOUS SPEECH RECOGNITION *

A. Varona, I. Torres

Dpto. Electricidad y Electrónica. Universidad del País Vasco
Apdo. 644 48080 Bilbao. SPAIN
E-mail (amparo@we.lc.ehu.es, manes@we.lc.ehu.es)

ABSTRACT

A syntactic approach of the well-known N-grams models, the K-Testable Language in the Strict Sense (K-TSS), is used in this work to be integrated in a Continuous Speech Recognition (CSR) system. The use of smoothed *K*-TSS regular grammars allowed to obtain a deterministic Stochastic Finite State Automaton (SFSA) integrating *K k*-TSS models into a self-contained model. An efficient representation of the whole model in a simple array of and adequate size is proposed. This structure can be easily handled at decoding time by a simple search function through the array. This formulation strongly reduced the number of parameters to be managed and thus the computing complexity of the model. An experimental evaluation of the proposed SFSA representation was carried out over an Spanish recognition task. These experiments showed important memory saving to allocate K-TSS Language models, more important for higher values of K. They also showed that the decoding time did not meaningfully increased when K did. The lower word error rates for the Spanish task tested were achieved for K=4 and 5. As a consequence the ability of this syntactic approach of the N-grams to be well integrated in a CSR system, even for high values of K, has been established.

1. INTRODUCTION

Statistical methods have been extensively used to generate Language Models (LM) to be integrated in large-vocabulary and/or Continuous Speech Recognition (CSR) Systems. They are based on the estimation of the probability of observing the N preceding lexical units (N-gram models): $P(w_1/w_1...w_{n-1})$. However in practice the use of this kind of models is reduce to low values of N. Nevertheless, it has been shown [1] that the probability distribution obtained through an N-gram model is strictly equivalent to the distribution obtained by a stochastic grammar generating a certain subclass of regular languages called K-Testable Language in the Strict Sense (*K*-TS) (*K* stands for the same meaning as N in N-grams). This probability distribution [2] [3] obtained from a training set must be modified to also consider those events not seen in the training corpus. So that, a syntactic back-off smoothing performing the integration of K *K*-TS models in a unique self-contained smoothed model was proposed in [4].

The aim of this work was to show the ability of this syntactic approach to be well integrated in a CSR system. For this purpose an efficient representation of the model in a simple array is proposed in this paper. This structure can be easily handled at decoding time by a simple search function through the array allowing models with high values of K be integrated in a CSR system.

The use of stochastic automata for language modeling has also been recently proposed [5] [6] [7] with the same aim to handle accurate language models in a one-step decoding procedure. However, the stochastic automata proposed in [5] and [7] are finite networks where the recognizer has to explore several word hypotheses due to their non deterministic nature. The use of K-TSS regular grammars allowed us to obtain a deterministic Stochastic Finite State (SFSA) that leads to get a simply maximum likelihood estimation of the probability of each transition.

The definition of the SFSA is provided in Section 2. In Section 3 the syntactic back-off smoothing technique is applied to the SFSA. In Section 4 the smoothed SFSA was represented in an efficient way as a simple array, easy to be managed in a CSR system. Finally, in Section 5 memory requirements and recognition experiments are shown.

2. The Stochastic Finite State Automaton (SFSA)

The SFSA is a self-contained model that integrates *K k*-TS automata [6], where $k=1,...,K$, in a unique automaton. Thus, the model was composed by *k* submodel, each one representing a *k*-TS LM. The whole automaton is defined by a five-tuple $(\Sigma, Q^K, q_0, q_f, \delta^K)$ where: $\Sigma = \{w_j\}$, $j = 1...|\Sigma|$ is the vocabulary, that is the set of words appearing in the training corpus; Q^K is the state set of the automaton (Each state represents a string of words $w_{i-k}w_{i-(k-1)}...w_{i-1}$, $k = 1...K$-1, with a maximum length of *K*-1, where *i* stands for a generic index in any string $w_1...w_i...$ appearing in the training corpus, such a state is labelled as w_{i-k}^{i-1}, see Figure 1); q_0 and $q_f \in Q^K$ are the initial and final states and δ^K is the transition function $\delta^K : Q^K \times (\Sigma) \rightarrow Q^K \times [0...1]$. $\delta^K(q, w_i) = (q_d, P(w_i/q))$ defines a destination state $q_d \in Q^K$ and a probability $P(w_i/q) \in [0...1]$ to be assigned to each element $(q, w_i) \in Q^K \times \Sigma$. Each transition represents a *k*-gram, $k = 1...K$; it is labelled by its last word w_i and connects two states labelled up to with *K*-1 words. As an example, transitions corresponding to strings of words of length *K* connecting states associated to string lengths equal to *K*-1 are defined as:

$$\delta^K(w_{i-(K-1)}^{i-1}, w_i) = (w_{i-(K-1)+1}^{i}, P(w_i / w_{i-(K-1)}^{i-1})) \qquad (1)$$

The whole and detailled definition of the Automaton is provided in [3]. Figure 1 represents the K-grams $w_{i-(K-1)}w_{i-(K-1)-1}...w_{(i-1)}$ and $w_{i-(K-1)+1}w_{i-(K-1)}...w_i$ labelling two states of the automaton. When w_i is observed an outgoing transition from the first to the second state is set and labelled by w_i .

* Work partially supported by the Spanish CICYT under grant TIC-95-0884-CO4-03

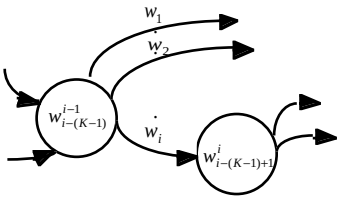

Figure 1: Two states of the SFSA representing two K-grams. Transitions are labelled by words appearing in the training sample after the *K*-gram labbelling the source state.

The model defined above is a deterministic and hence unambiguous stochastic finite state automaton [3]. The unambiguity of the automaton allows to obtain a simply maximum likelihood estimation of the probability of each transition $\delta^K(w_{i-(K-1)}^{i-1}, w_i)$ as:

$$P(w_i / w_{i-(K-1)}^{i-1}) = \frac{N(w_i / w_{i-(K-1)}^{i-1})}{\sum_{\forall w_j \in \Sigma} N(w_j / w_{i-(K-1)}^{i-1})} \quad (2)$$

where $N(w_j / w_{i-(K-1)}^{i-1})$ is the number of times the word w_j appears at the end of the *K*-gram $w_{i-(K-1)}...w_{i-1}w_j$, that is the count associated to the transition labelled by w_j coming from state labelled as $w_{i-(K-1)}^{i-1}$. This probability distribution must be modified to also consider the events not seen in the training corpus

3. The Smoothed SFSA

In previous works [4] a syntactic back-off smoothing procedure was developed. Under this formalism the probability $P(w_i/q)$ to be associated to a transition $\delta^K(q, w_i) = (q_d, P(w_i/q))$ is estimated according to:

$$P(w/q) = \begin{cases} \dfrac{N(w/q)}{N(q)+|\Sigma_q|} & w \in \Sigma_q \\ \dfrac{|\Sigma_q|}{N(q)+|\Sigma_q|} \dfrac{P(w/b_q)}{1-\sum_{\forall w' \in \Sigma_q} P(w'/b_q)} & w \in (\Sigma - \Sigma_q) \end{cases} \quad (3)$$

where: Σ_q is the vocabulary associated to state q and consists of the set of words appearing after the string labelling state q in the training corpus, i.e. words labelling the set of *seen* outgoing transitions from state q; $N(w/q)$ is the number of times that word w appears after the string labelling state q; $N(q) = \sum_{\forall w \in \Sigma_q} N(w/q)$, $|\Sigma_q|$ is the size of Σ_q and $P(w_i/b_q)$ is the estimated probability associated to the same event in the (k-1)-TS submodel; thus, if state q is labelled as w_{i-k}^{i-1} then state b_q is labelled as w_{i-k+1}^{i-1}.

The smoothing function (Equation (3)) estimates $|Q^K| \times |\Sigma|$ parameters that need large amount of space to be allocated, so that, it is prohibitive even for small vocabulary tasks. Nevertheless, the Smoothed SFSA can be represented in such a way that only transitions seen at training time need to be explicitly represented.

In Equation (3) the probabilities to be associated to the set of words appearing after the string labelling state q in the training corpus - the vocabulary of the state Σ_q - are explicitly estimated calculating $N(w/q)$ and $N(q)$. The remaining $|\Sigma| - |\Sigma_q|$ transition probabilities corresponding to those events not represented in the training corpus, are estimated according to more general probability distributions in k-TSS models, with $k<K$. However these transitions do not need to be explicitly estimated nor represented at each state. The structure of the previous automaton along with the back-off smoothing technique leads to group them into a unique transition to the back-off state b_q, which can be found in a more general sub-model. Taking into account that the stochastic condition must be satisfied, the probability to be assigned to the transition from each state to its back-off state, $P(b_q/q)$, can be easily estimated as:

$$P(b_q / q) = \frac{|\Sigma_q|}{N(q)+|\Sigma_q|} \frac{1}{1-\sum_{\forall w' \in \Sigma_q} P(w'/b_q)} \quad (4)$$

This transition connects each state q with its back-off state b_q which represents the same event in the (k-1)-TS model. Then, the probability to be associated to each event not represented in the training corpus $P(w_j/q)$ $\forall w_j \in (\Sigma' - \Sigma_q)$ is estimated according to:

$$P(w_j / q) = P(b_q / q)P(w_j / b_q) \ \forall w_j \in (\Sigma - \Sigma_q) \quad (5)$$

This formulation reduces the number of parameters to be handled from $|Q^K| \times |\Sigma|$ to $|Q^K| \times (|\Sigma_q|+1)$. Finally, this smoothing approach is incorporated to the previous definition of the SFSA to obtain a Smoothed SFSA. The Smoothed SFSA is defined by a five-tuple $(\Sigma, Q^K, q_0, q_f, \delta^K)$ where only δ^K needs to be modified. Each state of the automaton $q \in Q^K$ should add a new transition to its back-off state b_q:

$$\delta^K(q, U) = (b_q, P(b_q / q)) \quad (6)$$

where U represents any unseen even associated to state q which is labelled by a word $w_j \in (\Sigma - \Sigma_q)$. The back-off state b_q associated to each state q can be found in the (k-1)-TS submodel.

Figure 2 shows such a structure for a state q labelled as $w_{i-(K-1)}^{i-1}$. The transitions labelled by the $|\Sigma_q|$ words observed at training time after the K-gram labelling q connect it to other states in the same K-TS submodel. Transition labelled by U connects state q to its back-off state, $w_{i-(K-1)+1}^{i-1}$ in the (K-1)-TSS submodel.

4. An efficient representation of the SFSA

The main advantage of the previous formulation is that it leads to a very efficient representation of the model parameters learned at training time.

4.1.- An array allocating the Smoothed SFSA.

A simple array was used to allocate all the parameters of the model. A state of the automaton is represented by $|\Sigma_q|+1$ array rows, each of one representing an outgoing transition. Each position of the array represents a pair (q,w) where $q \in Q^K$ and

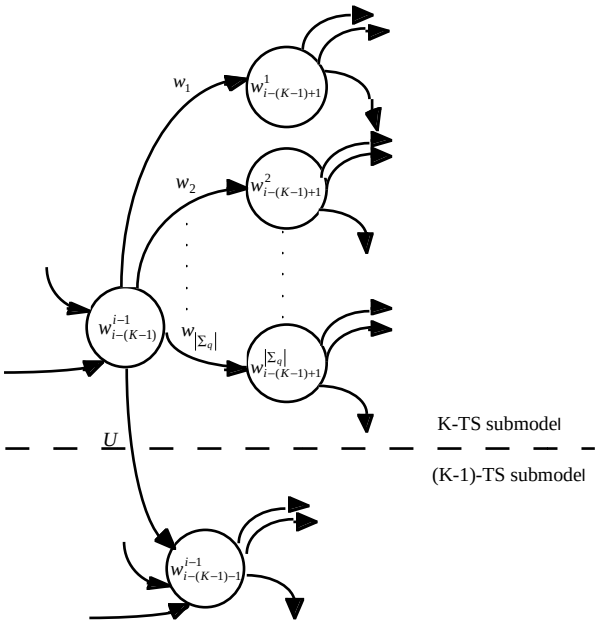

Figure 2: Transitions labelled by seen events connect each state to states in the same K-TSS submodel. Transition labelled by unseen events connnect it to its back-off state in the (K-1)-TSS submodel.

$w \in \Sigma_q \cup \{U\}$. It consists of four elements:

- a short integer, which represents a transition labelled by a word $w \in \Sigma_q$ or by the symbol U that stands for any unseen event.

- a double, which represents $P(w_i/q)$ $\forall w_i \in \Sigma_q$ or $P(b_q/q)$ for unseen events.

- a short integer, which represents the value of $|\Sigma_q|$.

- a short integer, which represents the explicit link to the first child of q or to its back-off state b_q

Figure 3 shows the allocation of the portion of the automaton showed in Figure 2. In this table the destination states corresponding to seen events can be found in positions corresponding to states of the same K-TSS submodel, down in the array (103, 105, etc.), whereas the back-off destination state is found in a position corresponding to a state in the (K-1)-TSS submodel, up in the array (76).

4.2.- The transition function δ as a simple search function through the array.

To complete the representation of the Smoothed SFSA the transition function δ of the Automaton defined in Section 3 should be proposed, and represented, for each state $q \in Q^K$ and for each $w \in \Sigma$. This transition function defines a destination state q_d and a probability $P(w/q)$ associated to each pair (q,w) $\forall q \in Q^K$ and $\forall w \in \Sigma'$:

$$\delta(q,w) = (q_d, P(w/q)) \;\; \forall q \in Q^K \wedge \forall w \in \Sigma' \;\; q_d \in Q^K \qquad (7)$$

When seen events appear, the destination state q_d can be found directly as the destination index of the array position (w,q). In the same way the value of $P(w/q)$, computed according to Equation (3) for $w \in \Sigma_q$, can be directly found as the probability

pos #q	label(q)	$\lvert\Sigma_q\rvert$	label(w)	P(w/q)	destination
.					
.					
84 32	$[w_{i-(K-1)}, w_{i-(K-1)+1}, \ldots, w_{i-1}]$	6	w_1	$P(w_1/q)$	103
85			w_2	$P(w_2/q)$	105
.					
.					
89			w_6	$P(w_6/q)$	113
90			U	P(U/q)	76
.					
.					

Figure 3: The array allocation of the portion of the Smoothed SFSA in Figure 2. Only the outlined elements need to be handled. However, some additional columns have also be included to clarify the meaning of each component.

value at the array position (w,q). However when unseen events appear both q_d and $P(w/q)$ values are not directly found in the array and a simple search function through the array is required.

This function, represented in Figure 4, search backwards across the back-off states, i.e. transitions through the U symbol, until the word w is found as a seen event for a state q in a lower level ($k<K$), i.e., $w \in \Sigma_q$. State q will be then the searched destination state q_d. The $P(w/q)$ value should be computed according to Equation (8) in this case. Thus, Equation (3) is recursively calculated while q_d is found by the search function.

```
Function δ (q∈ Q^K; w∈Σ): (dt∈ Q^K; P∈[0...1]);
var  q_aux∈ Q^K; P_aux∈[0...1];
          begin
if  w∈Σ_q  then   δ_d←array_dest[q,w];   {seen events}
                  δ_P←array_prob[q,w]
            else  q_aux←array_dest[q,U]; {unseen events}
                  P_aux←array_prob[q,U]
                  while w∉Σ_q_aux do
                    q_aux←array_dest[q_aux,U];
                    P_aux← P_aux × array_prob[q_aux,U]
                  end_while
                  δ_d ←array_dest[q_aux,w];
                  δ_P ←array_prob[q_aux,w]
end_if
end_δ.
```

Figure 4: A simple search function through the proposed array to compute Equation (7). d_d stands for the destination state q_d and d_P for the probability $P(w/q)$.

5.- Experimental assessment.

An experimental evaluation of the proposed Smoothed SFSA representation was carried out over a Spanish corpora. In Section 5.1, an evaluation of the memory requierements of the model is presented. Then the model was integrated in a CSR system (Section 5.2) showing word error rates and decoding time requiered by several K-TSS models.

For these experiments a task-oriented Spanish corpus (BDGEO) [8] consisting in 82,000 words and a vocabulary of 1,213 words was used. This corpus represents a set of queries to a Spanish geography database. This is a specific task designed to test integrated systems (acoustic, syntactic and semantic modelling) in automatic speech understanding. The training corpora consisted in 9150 sentences.

5.1.- Smoothed SFSA allocation

Some experimental measurements were carried out to evaluate the save of memory achieved with the proposed structured to

allocate the Smoothed SFSA. For comparison purposes the direct structure represented by a full array of $|Q^K| \times |\Sigma|+1$ positions was also considered. In such a case each position consisted of two elements representing the pair (destination state, transition probability associated). Table 1 shows the memory requirements for several values of K. The number of states of the SFSA, $|Q^K|$, is also provided.

Table 1: Memory required to allocate the Smoothed SFSA for BDGEO application task and different Ks.

K	$\|Q^K\|$	full array	proposed array
2	1,213	17.6 Mb	0.13 Mb
3	7,479	108.8 Mb	0.43 Mb
4	21,551	313.6 Mb	0.95 Mb
5	42,849	623.7 Mb	1.69 Mb
6	69,616	1013.3 Mb	2.55 Mb

Table 1 shows important reductions of the memory requirements when the proposed structure was used. These reductions were more important for higher values of *K*.

5.2.- The Smoothed SFSA in a CSR system.

The Smoothed SFSA, represented as proposed along the paper, was integrated in a CSR system. Each transition of the automaton was replaced by a chain of Hidden Markov models representing the acoustic model of each phonetic unit of the word. Then, the decoding scheme was performed by using the time-synchronous Viterbi algorithm. In the lattice the transition through each word of the vocabulary should be evaluated each time the system considers a LM state transition $\delta(q/w)$. Thus, the search function through the proposed array presented in Figure 3 was used to obtain for state q the following state q_d and the associated probability $P(w/q)$ for all the words in the vocabulary.

A Sylicon Graphics O_2 with a R10000 processor was used to recognice 600 sentences from the BDGEO task, uttered by 12 speakers. Recognition experiments using the full table are prohibitive, thus the comparison is not possible and only experiments with the proposed array were carried aut. In order to reduce the computational cost the beam-search algorithm was applied with two different widths: a narrower beam factor (bf) of 0.5 and a wider one of 0.7. Table 2 shows the word error rates (WER) for each different values of K. This Table also shows the average time needed to decode a sentence (seconds) and the average number of alive nodes at each frame in the lattice, thus they include acoustic and LM states.

Table 2: Word Error rates for different K-TSS models.

K	average number of active nodes		Time per sentence (sec)		WER	
	bf=0.5	bf=0.7	bf=0.5	bf=0.7	bf=0.5	bf=0.7
2	218.21	526.28	5.6	12.2	15.95	14.29
3	179.01	467.66	4.3	10.1	10.85	9.45
4	177.99	469.60	4.3	10.4	10.12	8.58
5	180.57	478.33	4.6	10.7	10.25	8.72
6	182.32	490.50	4.8	11.3	10.66	9.07

Table 2 shows that the use of the proposed Smoothed SFSA representation provides additional time reductions since the use of the function that manage the array is only needed for active paths. Table 1 shows that Smoothed SFSAs with high values of K need bigger amount of memory but the recognition rates in Table 2 prove that they can be easily integrated in a CSR system with not significatively increse of the decoding time. For the Spanish task used in the work the lower word error rates were achieved for *K*=4 and *K*=5 *K*-TSS models.

6.- Concluding remarks.

The use of smoothed *K*-TSS regular grammars allowed to obtain a deterministic, and hence unambiguous, Stochastic Finite State Automaton (SFSA) integrating *K k*-TSS models into a self-contained model. After applying a back-off smoothing technique, unseen events have an unique transition to a back-off state in a more general K-TSS model. So that, the Smoothed SFSA can be easily represented in an efficient way, where the probability distribution and transitions were represented in a simple array of an adequate size. A search function has been developed to manage the proposed representation allowing the final integration of the K-TSS model into a CSR system.

An experimental evaluation of the proposed Smoothed SFSA representation was carried out over a Spanish corpora. These experiments showed a great reduction of the number of parameter to be handled and, hence a very important memory saving. These reductions were more important for higher values of K. Then the Smoothed SFSA was integrated in a CSR system showing that the time required in recognition using Smoothed SFSA with high values of K is not significatively higher than for low values. The lowest word error rates for the Spanish task tested, were achieved when K= 4 and K= 5 K-TSS models were used. As a consequence the ability of this syntactic approach of the N-grams to be well integrated in a CSR system, even for high values of K, has been established.

7. REFERENCES

[1] Segarra, E. (1993): "Una Aproximación Inductiva a la Comprensión del Discurso Continuo," Ph Thesis. Universidad Politécnica de Valencia.

[2] García, P. and Vidal, E. (1990): "Inference of k-testable languages in the strict sense and application to syntactic pattern recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol 12, nº 9, pp. 920-925.

[3] Bordel, G., Varona, A. and Torres, Y. (1997): "K-TLSS(S) Language Models for Speech Recognition". *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol 2, pp 819-822.

[4] Bordel, G., Torres, I. and Vidal, E. (1994): "Back-off smoothing in a syntactic approach to Language Modelling". *Proc. International Conference on Speech and Language Processing*, pp. 851-854.

[5] Placeway, P., Schwartz, R., Fung, P. and Nguyen, L. (1993): "The estimation of powerful Language Models from small and large corpora," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* , vol. II, pp. 33-36.

[6] Zhao, R., Kenny, P., Labute, P. and O'Shoughnessy, D. (1993): "Issues in Large Scale Statistical Language Modeling". *Proc. of Eurospeech'93*, pp. 965-968.

[7] Riccardi, G., Pieraccini, R. and Bocchieri, E. (1996): "Stochastic automata for language modeling". *Computer Speech and Language* 10, pp. 265-293.

[8] Diaz, J. E., Rubio, A. J., Peinado, A. M., Segarra, E., Prieto, N. and Casacuberta, F. (1993); "Development of Task Oriented Spanish Speech Corpora," *Proceedings of EUROSPEECH 93.*