Computación y Sistemas Vol. 3 No. 4 pp. 237 - 244
 ISSN 1405-5546 Impreso en México

An Efficient Representation to k-TSS Language Models*

Inés Torres and **Amparo Varona**
Dpto. Electricidad y Electrónica. Universidad del País Vasco
Apdo. 644 48080 Bilbao. Spain
{manes, amparo}@we.lc.ehu.es

Article received on august 23, 1999; accepted on january 31, 2000

Abstract

k-Testable Languages in the Strict Sense (k-TSS), which are a subclass of regular languages, are used in this work to be integrated in a Continuous Speech Recognition (CSR) system. An efficient representation of the corresponding Stochastic Finite State Automaton (SFSA), that integrates K k-TSS models into a self-contained model, is proposed in this work. The whole model is represented in a simple array of an adequate size that can be easily handled at decoding time by a simple search function. An experimental evaluation of the proposed SFSA representation was carried out over an Spanish recognition task. These experiments showed that syntactic models could be efficiently represented and integrated in CSR systems.

Keywords: Continuous speech recognition, language modeling, syntactic pattern recognition.

* Work partially supported by the Spanish CICYT under grant TIC99-0423-C06-03

1 Introduction

Statistical methods have been extensively used to generate Language Models (LM) to be integrated in Continuous Speech Recognition (CSR) Systems. They are based on the estimation of the probability of observing a given lexical unit, conditioned on the observations of N preceding lexical units (N-gram models): $P(w_i/w_1...w_{n-1})$. However, in practice, the use of this kind of models is reduced to low values of N, exclusively bigrams and trigrams in really large-vocabulary recognition tasks (Placeway *et al.*, 1993).

Since the language constraints could be better modeled under a Syntactic approach, some formalisms based on regular grammars and context free grammars have also been used in LM. However, they have not extensively used since they present computational complexity problems. Moreover, full integration with the acoustic models in a CSR task and automatic learning from samples have been considered very difficult tasks under these formalisms (Segarra, 1993).

k-Testable Language in the Strict Sense (*k*-TSS) (Bordel *et al.*, 1997) are used in this work to be integrated in a CSR system. *k*-TSS Languages are a subclass of regular languages and can be inferred from a set of positive samples by an inference algorithm (García and Vidal, 1990). *k*-TSS language models can be considered as a syntactic approach of the well-known N-grams. In fact, Finite States Networks are nowadays used to represent these statistical models. In such a case, similar probability distributions can be obtained by N-grams and by k-TSS language models (Segarra, 1993) (Varona, 2000).

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 (Varona and Torres, 1999). 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 to be well managed. Some decoding experiments over a Spanish recognition task showed that syntactic models can be efficiently represented and integrated in CSR systems.

2 The Stochastic Finite State Automaton (SFSA)

The Smoothed SFSA is a self-contained model that integrates *K k*-TSS automata, where k=1,..., *K*, in a unique automaton. It is defined by a five-tuple (Σ', Q^K, q_0, q_f, δ^K) where:

- $\Sigma' = \Sigma \cup \{\$\}$ being $\Sigma = \{w_j\}$, j = 1...$|\Sigma|$, is the vocabulary, i.e. the set of words appearing in the training corpus. The nil symbol \$ was included in the training corpus to isolate each sentence.

- 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 labeled as w_{i-k}^{i-1}. States representing the initial strings of training sentences are labeled as $\$w_{i-k}^{i-1}$ where k = 1...*K*-2 to guarantee a maximum length of *K*-1. A special state labeled as λ represents a void string of words.

- The automaton has a unique initial and final state $q_0 \equiv q_f \in Q^K$ which is labeled as \$.

- δ^K is the transition function. δ^K: $Q^K \times (\Sigma \cup \{\$\}) \rightarrow Q^K \times [0...1]$. $\delta^K(q, w_i) = (q_d, P(w_i/q))$ 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 $(q, w_i) \in Q^K \times (\Sigma \cup \{\$\})$. Each transition represents a *k*-gram, k = 1...*K*; it is labeled by its last word w_i and connects two states labeled up to with *K*-1 words. As an example, transitions corresponding to strings of words of length *K* connecting states associated to string lengths *K*-1 are defined as:

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

The whole definition of the Automaton is provided in (Varona, 2000). Figure 1 represents the K-grams $w_{i-(K-1)}w_{i-(K-1)-1}...w_{(i-1)}$ and $w_{i-(K-1)+1}w_{i-(K-1)}...w_i$ labeling two states of the automaton. When w_i is observed an outgoing transition from the first to the second state is set and labeled by w_i.

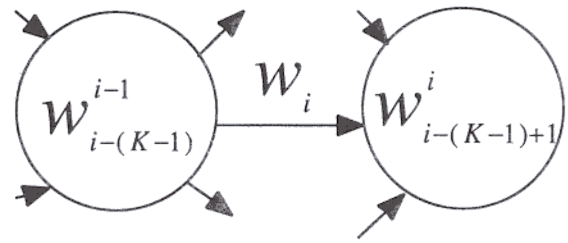

Figure 1: Two states of the SFSA representing two K-grams. Transitions are labelled by words appearing in the training sample after the *K*-gram labelling the source state.

Such a model can only associate a probability to any string of words that had been observed in the available samples, i.e., *seen events*. However, a probability need also to be associated to those events not represented in the training corpus, i.e., *unseen events*. To deal with this problem, a syntactic back-off smoothing procedure was developed in previous works (Bordel *et al.*, 1994). Under this formalism, the probabilities to be associated to the set of words appearing after the string labeling state *q* in the training corpus - the vocabulary of the state Σ_q - are explicitly estimated according to the counts *N(w/q)*, which are the number of times that word *w* appears after the string labeling state *q*. However, this counts are reduced to save some probability mass to be redistributed to the ($|\Sigma| - |\Sigma_q|$) unseen events associated to state *q*. Then, under the syntactic back-off smoothing (Bordel *et al.*, 1994), the transitions probabilities corresponding to those events not represented in the training corpus, are estimated according to more general probability distributions in *k*-TSS models, with $k<K$. These transitions can be grouped into a unique transition to a back-off state b_q associated to each state *q*. Thus, each state of the automaton $q \in Q^K$ should add a new transition to its back-off state b_q:

$$\delta^K(q,U) = (b_q, P(b_q / q)) \quad (2)$$

where *U* represents any unseen even associated to state *q*, *which* is labeled 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)-TSS submodel.

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:

$$P(w_j / q) = P(b_q / q) P(w_j / b_q) \; \forall w_j \in (\Sigma - \Sigma_q) \quad (3)$$

A complete presentation of the syntactic back off scheme can be found in (Bordel *et al.*, 1994) and (Varona, 2000). Figure 2 shows such a structure for a state q labeled as $w_{i-(K-1)}^{i-1}$. The transitions labeled by the $|\Sigma_q|$ words observed at training time after the K-gram labeling q connect it to other states in the same K-TSS submodel. Transition labeled by U connects state q to its back-off state, $w_{i-(K-1)+1}^{i-1}$ in the (K-1)-TSS submodel.

3 An Efficient Representation of the Smoothed SFSA

The main advantage of this formulation is that it leads to a very efficient representation of the model parameters learned at training time, that are probability distribution and model structure. In this Section the procedure to obtain such a representation is presented. In a first step a finite network represented the structure of the Smoothed SFSA.

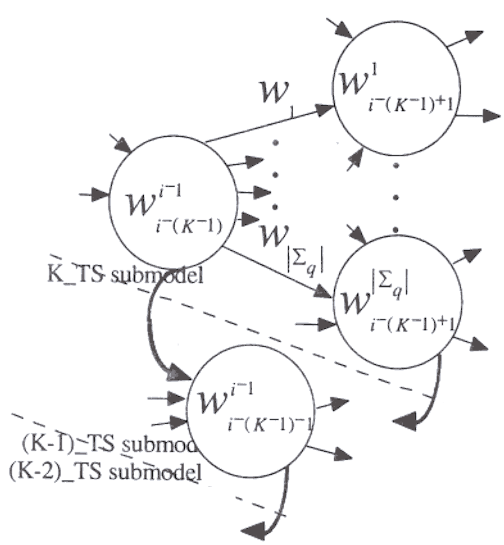

Figure 2: Transitions labelled by seen events($w_j \in \Sigma_q$) connect each state to states in the same K-TSS submodel. Transition labelled by unseen events connects it to its back-off state in the (K-1)-TSS submodel.

This network was derived from an initial trie built from the training set. This procedure is shown in Section 3.1. The initial trie nature of the network let to allocate both, the structure of the Smoothed SFSA and the probability distributions in a simple array, presented in Section 3.2. To complete the representation of the Smoothed SFSA the transition function δ of the Automaton defined in Section 2 should be defined, and thus represented, for each state $q \in Q^K$ and for each $w \in \Sigma'$. This goal is achieved by a simple search function through the proposed array, which is presented in Section 3.3.

3.1 A Finite Network Representing the Structure of the Smoothed SFSA

In a first step, the data obtained from the learning sample are stored in a trie structure with *K-1* levels. Each node at each level k, with k = 1..K-1, is associated to a state of the Smoothed SFSA labeled by a string of words of length k, w_{i-k}^{i-1}, that has been observed in the training set, that is, seen events. The number of children of each node $q \equiv w_{i-k}^{i-1}$ is equal to the number of words appearing after w_{i-k}^{i-1} in the training sentences, that is, $|\Sigma_q|$. The root of the trie is associated to state λ representing a void string of words. Level one represents the 1-TSS model and consists of $|\Sigma|$ nodes corresponding to each word of the vocabulary. As mentioned in previous Section, the automaton has a unique initial and final $q_0 \equiv q_f$, which is labeled as \$. This fact leads to consecutively parser sets of sentences that are forced to start and finish with \$. This state is different to the root node of the trie, labeled as λ, since the probability $P(w_i/\lambda)$ is the estimated probability $P(w_i)$, whereas $P(w_i/\$)$ is the probability associated to w_i being the initial word in a sentence. As a consequence, $P(\$/\lambda)$ would be the probability of a sentence and would only be considered when parsing a new sentence. Then, when each sentence is individually parsed, this link can be removed. In such a way, the original trie is transformed in a bi-trie where λ and \$ are the root nodes but only \$ represents the initial state of the automaton.

The procedure to represent the Smoothed SFSA efficiently is illustrated with a simple example. A famous poem by the Spanish poet Miguel Hernández has been selected to act as the sample text corpus. Figure 3 shows this poem, the corresponding training set R^+ and vocabulary Σ.

Figure 4 shows the bi-trie for K=4 when the training set R^+ and the vocabulary Σ of Figure 3 were used. In this Figure, the corresponding nodes of the bi-trie represent all the states of the automaton. Transitions from state labeled by λ which represents a void string of words, are represented by links between the root λ and its children nodes. Links connecting node \$ to its children represent transitions from the initial state. Transitions that correspond to strings of words shorter than K are represented by $|\Sigma_q|$ links connecting each node of the k=1..K-2 levels to its $|\Sigma_q|$ children. Nodes of K-1 level represent the states associated to word strings of length equal to K-1 labeled as $w_{i-(K-1)}^{i-1}$. For the sake of clarity, transitions to the final state established for all nodes in the bi-trie each time the symbol $ appears in the training corpus are not drawn in Figure 4. Nodes representing states with transitions to node labeled as $ are outlined instead.

con tres heridas
la del amor
la de la muerte
la de la vida
con tres heridas viene
la de la vida
la del amor
la de la muerte
con tres heridas yo
la dè la vida
la de la muerte
la del amor

Miguel Hernández

$$R^{+} = \left\{ \begin{array}{l} \textit{\$llego con tres heridas \$ la del} \\ \textit{amor \$ la de la muerte \$ la de la} \\ \textit{vida \$ con tres heridas viene \$} \\ \textit{la de la vida \$ la del amor \$ la} \\ \textit{de la muerte \$ con tres heridas} \\ \textit{yo \$ la de la vida \$ la de la} \\ \textit{muerte \$ la del amor \$} \end{array} \right\}$$

$$\Sigma = \left\{ \begin{array}{l} \textit{la, de, del, amor, muerte, vida,} \\ \textit{llego, con, tres, heridas, viene, yo} \end{array} \right\}$$

Figure 3: Selected training corpus to illustrate the procedure to represent the Smoothed SFSA efficiently

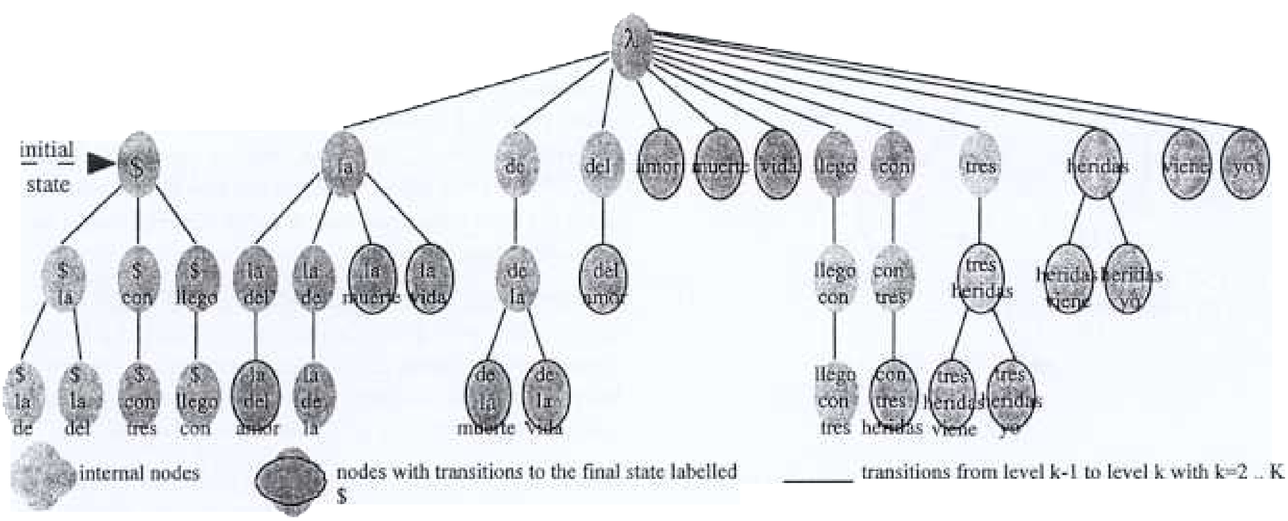

Figure 4: Bi-trie directly obtained from the training set R^{+} of Figure 3 after the first step of the procedure for K=4 model.

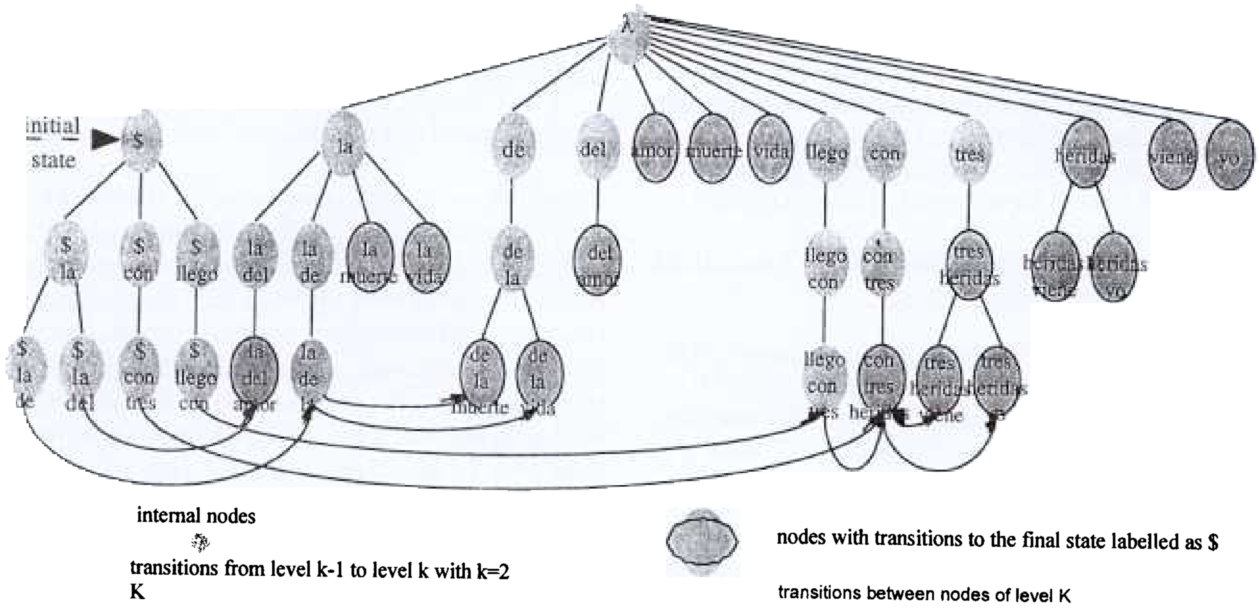

Figure 5: The network representing the structure of the non smoothed SFSA (K=4) obtained from the training set R^+ of Figure 3.

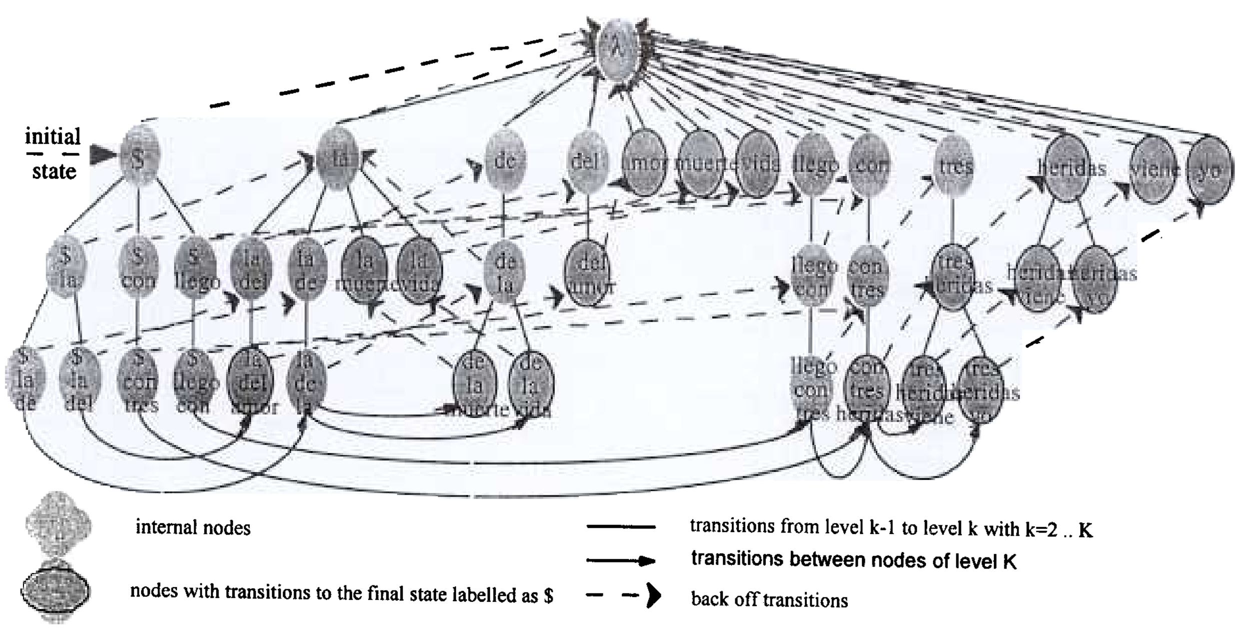

Figure 6: The complete network representing the structure of the Smoothed SFSA (K=4) obtained from the training set R^+ of Figure 3.

To complete the representation of the SFSA, transitions that correspond to strings of words of length K connecting states associated to string lengths equal to K-1 need also to be added. These transitions are represented by links between nodes of the K-1 level labeled as $w_{i-(K-1)}^{i-1}$.

The finite network obtained by including these transitions represents the structure of the complete non-smoothed SFSA. Figure 5 shows such a structure for the training corpus of Figure 3.

In a final step, the smoothing technique is applied to each node in order to obtain a network representing the

Smoothed SFSA. As a consequence, new links representing back-off transitions should be created. As mentioned above, the back-off state b associated to each state q can be found in the (k-1)-TS model. Thus, the destination node for this link should then be found in the previous level of the trie. Figure 6 shows the finite network representing the structure of the Smoothed SFSA (K=4) for the training corpus in Figure 3.

3.2 A Simple Array to Allocate 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 represents a pair (q, w) where $q \in Q^K$ and $w \in \Sigma_q \cup \{U\}$.

It consists of four elements:

- a short integer, which represents a transition labeled by a word $w \in \Sigma_q$ or by the symbol U that stands for any unseen event.
- a double, which represents $P(w_i/q) \leftrightarrow w_i \in \Sigma_q$ 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 7 shows this table representing the Smoothed SFSA in Figure 6. In Figure 7 some additional information has also be included to clarify the meaning Of each array component: state number #q and its label, the word label and the table index. However, only four elements, #w, $P(w/q)$, $|\Sigma_q|$ and the index of the destination position, need to be handled. For the sake of clarity each transition is represented by the label of the last word of the involved string, (label(w)), instead of by its integer index (#w). In this column symbol U represents any unseen even associated to state q which is labeled by a word $w_j \in (\Sigma'-\Sigma_q)$. The destination of this transition is the corresponding back-off state b_q and the associated probability $P(b_q/q)$ was calculated according to syntactic back-off smoothing procedure (Varona and Torres., 1999).

ind	#q	label(q)	$\|\Sigma_q\|$	label(w)	P(w/q)	de	#k-TSS
1	1	[λ]	12	la	0.3333	17	
2				de	0.1333	22	
3				con	0.0667	24	
4				tres	0.0667	26	
5				heridas	0.0667	28	
6				del	0.0667	32	
7				amor	0.0667	34	
8				muerte	0.0667	36	
9				vida	0.0667	38	
10				viene	0.0222	40	
11				yo	0.0222	42	
12				llego	0.0222	44	K=1
13	2	[$]	3	la	0.6000	46	
14				con	0.1333	49	
15				llego	0.0667	5	
16				*U*	0.3461	1	
17	3	[la]	4	de	0.3158	55	
18				del	0.1579	53	
19				muerte	0.1579	57	
20				vida	0.1579	59	
21				*U*	0.3158	1	
22	4	[de]	1	la	0.8571	61	
23				*U*	0.2143	1	
24	5	[con]	1	tres	0.7500	64	
25				*U*	0.2679	1	
26	6	[tres]	1	heridas	0.7500	66	
27				*U*	0.2679	1	
28	7	[heridas]	3	$	0.1667	13	
29				viene	0.1667	70	
30				yo	0.1667	72	
31				*U*	0.5232	1	
32	8	[del]	1	amor	0.7500	74	
33				*U*	0.2679	1	
34	9	[amor]	1	$	0.7500	13	
35				*U*	0.2500	1	
36	10	[muerte]	1	$	0.7500	13	
37				*U*	0.2500	1	
38	11	[vida]	1	$	0.7500	13	
39				*U*	0.2500	1	
40	12	[viene]	1	$	0.7500	13	
41				*U*	0.2500	1	
42	13	[yo]	1	$	0.7500	13	
43				*U*	0.2500	1	
44	14	[llego]	1	con	0.5000	76	
45				*U*	0.5357	1	K=2
46	15	[$la]	2	de	0.5455	78	
47				del	0.2727	80	
48				*U*	0.3455	17	
49	16	[$con]	1	tres	0.6667	82	
50				*U*	1.3333	24	
51	17	[$llego]	1	con	0.5000	84	
52				*U*	1.0000	44	
53	18	[la del]	1	amor	0.7500	86	
54				*U*	1.0000	32	
55	19	[la de]	1	la	0.8571	88	
56				*U*	1.0000	22	
57	20	[la muerte]	1	$	0.7500	13	
58				*U*	1.0000	36	
59	21	[la vida]	1	$	0.7500	13	
60				*U*	1.0000	38	
61	22	[de la]	2	muerte	0.3750	91	
62				vida	0.3750	93	
63				*U*	0.3654	17	
64	23	[con tres]	1	heridas	0.7500	95	
65				*U*	1.0000	26	
66	24	[tres heridas]	3	$	0.1667	13	
67				viene	0.1667	99	
68				yo	0.1667	101	
69				*U*	1.0000	28	
70	25	[heridas viene]	1	$	0.5000	13	
71				*U*	1.0000	40	
72	26	[heridas yo]	1	$	0.5000	13	
73				*U*	1.0000	42	
74	27	[del amor]	1	$	0.7500	13	
75				*U*	1.0000	34	
76	28	[llego con]	1	tres	0.5000	103	
77				*U*	2.0000	24	K=3
78	29	[$ la de]	1	la	0.8571	88	
79				*U*	1.0000	55	
80	30	[$ la del]	1	amor	0.7500	86	
81				*U*	1.0000	53	
82	31	[$ con tres]	1	heridas	0.6667	95	
83				*U*	1.3333	64	
84	32	[$ llego con]	1	tres	0.5000	103	
85				*U*	1.0000	76	
86	33	[la del amor]	1	$	0.7500	13	
87				*U*	1.0000	74	
88	34	[la de la]	2	muerte	0.3750	91	
89				vida	0.3750	93	
90				*U*	1.0000	61	
91	35	[de la muerte]	1	$	0.7500	13	
92				*U*	1.0000	57	
93	36	[de la vida]	1	$	0.7500	13	
94				*U*	1.0000	59	
95	37	[con tres heridas]	3	$	0.1667	13	
96				viene	0.1667	99	
97				yo	0.1667	101	
98				*U*	1.0000	66	
99	38	[tres heridas viene]	1	$	0.5000	13	
100				*U*	1.0000	70	
101	39	[tres heridas yo]	1	$	0.5000	13	
102				*U*	1.0000	72	
103	40	[llego con tres]	1	heridas	0.5000	13	
104				*U*	2.0000	64	K=4

Figure 7: Array allocating the SFSA represented in Figure 6. Only outlined elements are required.

3.3 Transition Function δ: Searching Through the Array

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

$$\delta(q,w) = (q_d, P(w/q)) \;\; \forall q \in Q^K \wedge \forall w \in \Sigma' \;\; q_d \in Q^K \quad (4)$$

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

Figure 8: Search function through the array computing δ.

When seen events appear the destination state q_d can be found directly as the destination index of the array position (q, w) (see Figure 7). In the same way the value of $P(w/q)$, can be directly computed from training samples, according to the syntactic back-off smoothing procedure (Bordel *et al.*, 1994) for $w \in \Sigma_q$. It is represented as the probability value at the array position (w, q) (see Figure 7). 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 8, 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 $\dot{q}$ in a lower level ($k<K$), i.e., $w \in \Sigma_{\dot{q}}$. State q will be then the searched destination state q_d. The $P(w/q)$ value should be computed according to Equation (3) in this case. Thus, Equation (3) is recursively calculated while q_d is found by the search function. This procedure, that represents Equation (4), is described by Function δ in Figure 8: δ_d stands for the destination state q_d and δ_P for the probability $P(w/q)$.

4 Experimental Assessment

An experimental evaluation of the proposed Smoothed SFSA representation was carried out over a Spanish corpora. The Smoothed SFSA, represented as proposed along the paper, was integrated in a CSR system (Rodríguez *et al.*, 1999). Thus, acoustic models should be integrated with language models in a unique decoding scheme. As mentioned above, the syntactic approach allowed to deal with this problem by using the SFSA presented along the paper. In such an automaton, each transition 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 SFSA transition $\delta(q/w)$. Thus, the search function through the proposed array presented in Figure 8 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.

The corpora consists of a set of simple Spanish sentences describing visual scenes (Miniature Language Acquisition task - MLA) and was first presented in English (Feldman *et al.*, 1990). The training set consisted in 9150 sentences that were randomly generated by using a context-free model of the language. It included 14,702 words and a very limited vocabulary size (29 words). Additional data about k-TSS language models inferred from this corpus can be found in (Varona, 2000) and (Varona and Torres, 2000).

A Silicon Graphics O_2 with a R10000 processor was used to recognize 1600 sentences from the MLA task, uttered by 16 speakers. In order to reduce the computational cost a beam-search algorithm was applied with two different widths: a narrower beam factor (bf) of 0.5 and a wider one of 0.7. The beam-search algorithm eliminates the less probable paths of the trellis.

Table 1 shows the word error rates (WER) for each different values of K. This Table also shows the average time per frame (ATpF) (msec) needed to decode a sentence (seconds) and the average number of alive nodes at each frame (ANANpF) in the lattice. These nodes include acoustic and LM states. Memory requirements (in Kb) and the number of states of the SFSA $|Q^K|$ are also provided for each value of K.

Table 1: Word Error rates for integrated models: K=2. 6.

K	$\|Q^K\|$	Kb	ANANpF		ATpF (msec)		WER	
			f=0.5	bf=0.	f=0.5	bf=0.7	f=0.5	bf=0.7
2	31	3.0	39	64	3	4	5.01	4.60
3	173	11.9	47	80	3	5	4.83	4.36
4	643	35.6	50	87	4	6	4.33	3.62
5	180 8	93.0	52	92	4	7	3.64	3.26
6	451 8	218.6	54	98	5	7	3.51	2.68

Table 1 shows that the use of the proposed Smoothed SFSA representation provides additional time reductions since the search function trough the array (Figure 8) is only needed for active paths. This Table also shows that Smoothed SFSAs with high values of K need bigger amount of memory but they can be easily integrated in a CSR system with not significatively increase of the decoding time, since the average number of active nodes did not increase as $|Q^K|$ did. Recognition rates show that the CSR system performance increased with K for this task.

5 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 a 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 recognition task. These experiments showed that accurate syntactic models could be efficiently represented and integrated in CSR systems.

6 References

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

Segarra, E. "Una Aproximación Inductiva a la Comprensión del Discurso Continuo," Ph Thesis. Universidad Politécnica de Valencia, 1993.

García, P. and **E. Vidal, E.**, "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, 1990.

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

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

Feldman, J.A., G. Lakoff, A. Stolcke and **S.H. Weber**, "Miniature Language Acquisition: a touch-stone for cognitive science". Technical report, TR-90-009. ICSI, Berkeley, 1990.

Rodríguez, L.J., I. Torres, J.M. Alcaide, A. Varona, K. López de Ipiña, M. Peñagarikano, G. Bordel "An integrated system for Spanish CSR tasks". *Proc. of EUROSPEECH*, vol 2, pp 951-954, 1999.

Varona, A. and **I. Torres** "Using Smoothed *k*-TSS Language Models in Continuous Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol II, oo. 729-732, 1999.

Varona, A "Modelos k-explorables en sentido estricto integrados en un sistema de reconocimiento automático del habla". *Ph Thesis*, Universidad del País Vasco (Spain), Abril, 2000.

Varona, A. and **I. Torres** "Evaluating pruned k-TSS language models: perplexity and word recognition rates". Being published in *Pattern Recognition and Applications*, Ed. Ios Press, The Netherlands, 2000.

Inés Torres: *was born in the Basque Country, Spain. She received the degree in physics from the University of the Basque Country in 1981 and the Ph.D. degree from the same university in 1991. In 1982, she joined the Electricity and Electronics department of the University of the Basque Country. She has been teaching computer science, first as an Associate Professor and since 1996 as a Titular Professor. She has also been the head of the Automatic Speech Recognition group in the University of the Basque Country. Her areas of interest include all aspects of Automatic Continuous Speech Recognition, e.g., signal processing, search strategies and language modeling.*

Dr. Torres is a board member of the Spanish Society for Pattern Recognition and Image Analysis (AERFAI), which is an affiliate society of IAPR, and she is also a member of the International Speech Communication Association (ISCA).

Amparo Varona: *was born in the Basque Country, Spain. She received the degree in physics from the University of the Basque Country in 1993 and the Ph.D. degree from the same university in 2000. Since February 1996, she has been teaching computer science as Associate Professor in the Electricity and Electronics Department of the University of the Basque Country. Her areas of interest include all aspects of Automatic Continuous Speech Recognition, e.g., search strategies and language modeling.*

Dr. Varona is a member of the Spanish Society for Pattern Recognition and Image Analysis (AERFAI), which is an affiliate society of IAPR.