LAYERED MARKOV MODELS: A NEW ARCHITECTURAL APPROACH TO AUTOMATIC SPEECH RECOGNITION

Mikel Penagarikano and German Bordel
Department of Electricity and Electronics
University of the Basque Country, 48940 Leioa, Spain
E-mail: mpenagar@we.lc.ehu.es, german@we.lc.ehu.es
Web: gtts.ehu.es

Abstract. This paper presents the theoretical basis of *layered Markov model*s (LMM), which integrate all the knowledge levels commonly used in automatic speech recognition (acoustic, lexical and language levels) in a single model. Each knowledge level is represented by a set of Markov models (or even hidden Markov models) and all these sets are arranged in a layered structure. Given that common supervised training and recognition paradigms can be also expressed as simple Markov models, they can be formalized and integrated into the model as an extra knowledge layer. In addition, it is shown that hidden Markov models (HMM) and newer HMM2 can be considered as particular instances of LMM.

INTRODUCTION

In state-of-the-art automatic speech recognition (ASR) systems, different models corresponding to different knowledge levels are integrated in order to get a joint probability distribution which is used to obtain the most probable pronounced sentence. Most of used models can be described as Markov models (implemented as weighted finite-state automata) and hidden Markov models. The purpose of this paper is to introduce a novel architectural approach, layered Markov models (LMM), whom formalism allows the integration of all such knowledge levels into a single model.

First, the LMM formalism is introduced and next, HMM and HMM2 are studied as particular instances of LMM. Afterward, the way to insert different recognition-training paradigms into a LMM is presented, unifying all of them and reducing the problem to the standard recognition procedure using just one model. In the end, some conclusions are presented.

FORMAL DEFINITION

A layered Markov model (LMM) consists of a number of layers, each of them composed by a finite set of Markov models (see Fig.1). Such a set of Markov models represents a knowledge level that modelizes it's units in terms of lower level units. For example, the set of pronunciation models of an ASR system modelizes words in terms of phonemes, diphonemes or other lower level units. In fact, one layer is connected to the underlying one in the sense that bottom layer's models (classes) correspond to upper layer's alphabet. Each Markov model can be represented by a weighted finite-state automaton (WFA), and thus the layered model can be described in terms of such models.

First some notation: the quintuple $m \equiv \left(Q, q^I, Q^F, \Sigma, \delta\right)$ refers to a WFA implementation of a Markov model, being Q the set of states, q^I the initial state, Q^F the set of final (accepting) states, Σ the symbol alphabet, and $\delta : Q \times \Sigma \rightarrow Q \times \mathbb{R}$ the weighted transition function ($\acute{q} = next\left(q, \alpha\right)$ refers to the destination state given a source state $q \in Q$ and a symbol $\alpha \in \Sigma$, whereas $p\left(\alpha \mid q\right)$ is the probability of such transition, thus, $\delta\left[q, \alpha\right] = \left[next\left(q, \alpha\right), p\left(\alpha \mid q\right)\right]$)[1]. A knowledge layer of the LMM is defined by a set $L \equiv \{m \in M\}$ of Markov models sharing the same alphabet Σ_L, and all the layers are arranged on a vector $\mathbb{L} = \left[L_1, L_2, \ldots, L_N\right]$ [2], which last element (the top layer) must contain only one Markov model. Each layer is connected to the underlying one by means of the function $\gamma : \Sigma_{L_i} \rightarrow L_{i-1}$ that maps ith layer's alphabet into *i-1*th layer's models. Therefore, a LMM consists of a vector of layers and such a mapping function: $\Gamma \equiv \left(\mathbb{L}, \gamma\right)$.

The whole LMM (see Fig.1) can be seen as a new WFA, with a number of *metastates* $\mathbf{q} \in \mathbf{Q}$, described by a vector $\mathbf{q} = \left[\left(m_1, q_1\right), \ldots, \left(m_N, q_N\right)\right]$ made up of a model-state pair per layer. The alphabet of the whole LMM will be the bottom layer's alphabet $\Sigma = \Sigma_{L_1}$ and a global weighted transition function ξ is implicit in the model description. As will be pointed later, the weighted transition function (and therefore, the resulting WFA) will be nondeterministic: $\xi : \mathbf{Q} \times \Sigma_{L_1} \rightarrow \{\mathbf{Q} \times \mathbb{R}\}$.

LMM transitions

Admitting that the LMM is nondeterministic, as we will later see, the destination of a transition is formalized as a function $next\left(\mathbf{q}, \alpha\right)$ that returns the destination metastate set $\acute{\mathbf{Q}} = \{\acute{\mathbf{q}}_{\mathbf{q},\alpha} = \left[\left(\acute{m}_1, \acute{q}_1\right), \ldots, \left(\acute{m}_N, \acute{q}_N\right)\right]\}$ given a source metastate $\mathbf{q} = \left[\left(m_1, q_1\right), \ldots, \left(m_N, q_N\right)\right]$ and a symbol $\alpha \in \Sigma = \Sigma_{L_1}$, while the transition probability is formalized as the probability distribution function $p\left(\acute{\mathbf{q}}, \alpha \mid \mathbf{q}\right)$. Next, computing of both functions will be presented:

[1] The final state paradigm can be understood as an outgoing transition, and therefore integrated as a particular case: $\delta\left[q, \alpha_{out}\right] = \left[null, p\left(\alpha_{out} \mid q\right)\right]$, where $p^F\left(q\right) = p\left(\alpha_{out} \mid q\right)$ would be the probability of being final. All the posible transitions (inners and outgoings) must sum to a total probability of 1, that is, $\sum_{\alpha \in \Sigma} p\left(\alpha \mid q\right) + p^F\left(q\right) = 1$.

[2] We will refer to L_1 as the *bottom layer*, and L_N as the *top layer*, in order to set up an intuitive graphical image of the model.

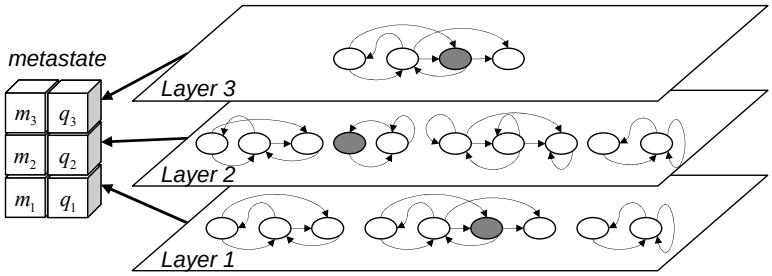

Figure 1: A schematic image of a *3layer*-LMM. Only one model must be placed at top level. A metastate is a vector with a model-state pair per layer.

first a very simple case will be considered, later a more complex one and in the end the general one.

If the lowest state of the metastate is not a final one (at its own model), the transition only alters the lowest layer state:

$$\acute{q_1} = next_{m_1}(q_1, \alpha)$$

In such a case, we will say that the transition happens at bottom layer, that is, only bottom layer model is involved and therefore only the lowest layer state changes. As all models in the layers are deterministic, in this case there will be only one destination metastate:

$$\acute{\mathbf{Q}} = next_{L_1}(\mathbf{q}, \alpha) = \acute{\mathbf{q}}_{\mathbf{q},\alpha} = [(m_1, \acute{q_1}), \ldots, (m_N, q_N)]$$

and the transition probability is straightforward:

$$p(\acute{\mathbf{q}}, \alpha \mid \mathbf{q}) = p(\alpha \mid \mathbf{q}) = p_{m_1}(\alpha \mid q_1)$$

But in a more complex case, transition might happen at higher layers and lead to more than one destination metastates. For instance, if the lowest state is final, transitions at second layer must be considered: if second layer state is not final, transition will happen at second layer. As only lowest layer symbols are observed, all possible $\alpha_2 \in \Sigma_{L_2}$ symbols (and therefore transitions) must be taken into account.[3] Each transition at this layer fixes the destination state at layer 2, and the emitted symbol α_2 is mapped to a Markov model of the underlying layer:

$$\acute{q_2} = next_{m_2}(q_2, \alpha_2)$$
$$\acute{m}_1 = \gamma(\alpha_2)$$

The destination state at layer 1 arises from the transition given the initial state of the mapped Markov model and the observed symbol,

$$\acute{q_1} = next_{\acute{m}_1}(q^I, \alpha)$$

[3]By $\alpha_2 \in \Sigma_{L_2}$, we don't mean the complete alphabet Σ_{L_2}, but only those symbols related to existing transitions from state q_2.

and the destination metastate set, given all possible transitions at the upper layer can be stated as:

$$\acute{\mathbf{q}}_{\mathbf{q},\alpha}\left(\alpha_2\right) = \left[\left(\acute{m}_1, \acute{q_1}\right), \left(m_2, \acute{q_2}\right), \left(m_3, q_3\right) \ldots, \left(m_N, q_N\right)\right]$$

$$next_{L_2}\left(\mathbf{q}, \alpha\right) = \left\{\acute{\mathbf{q}}_{\mathbf{q},\alpha}\left(\alpha_2\right)\right\}_{\alpha_2 \in \Sigma_{L_2}}$$

Therefore, there exist more than one possible transitions for each metastate (as much as transitions at layer 2), justifying our previous definition of the LMM as a nondeterministic weighted finite-state automaton. Moreover, if state q_1, as well as being final, could transit at model m_1 and observable α, such a transition should be added as considered for the first case:

$$\acute{\mathbf{Q}} = next_{L_1}\left(\mathbf{q}, \alpha\right) \cup\ next_{L_2}\left(\mathbf{q}, \alpha\right)$$

The probability of each metastate transition in the set $next_{L_2}\left(\mathbf{q}, \alpha\right)$ is the joint probability of q_1 being final (layer 1), transition $q_2 \rightarrow \acute{q_2}$ (layer 2) and transition $q_I \rightarrow \acute{q_1}$ (layer 1):

$$p\left(\acute{\mathbf{q}}_{\mathbf{q},\alpha}\left(\alpha_2\right), \alpha \mid \mathbf{q}\right) = p^F_{m_1}\left(q_1\right) \cdot p_{m_2}\left(\alpha_2 \mid q_2\right) \cdot p_{\acute{m}_1}\left(\alpha \mid q^I\right)$$

In a more general case, states $q_1 \ldots q_{i-1}$ of the metastate $\mathbf{q}$ could be final states. Hence, the transition will happen at layer i and, once again, all possible transitions from the source state q_i should be processed to obtain destination states $\acute{q_i}$. Each emitted symbol $\alpha_i \in \Sigma_{L_i}$ is then mapped to a model $\acute{m}_{i-1} = \gamma\left(\alpha_i\right)$ at layer $i-1$. If layer $i-1$ is not the bottom one (or, $i > 2$), there is no fixed symbol, and therefore all possible transitions from mapped models initial state must be taken into account. All the symbols $\alpha_{i-1} \in \Sigma_{L_{i-1}}$ of such transitions are mapped again to models at layer $i-2$ and the process is repeated until the bottom layer is reached. At bottom layer, only transitions from initial states and observed symbol α can be performed:

$$\acute{q_i} = next_{m_i}\left(q_i, \alpha_i\right)$$

$$\acute{q_j} = next_{\acute{m}_j = \gamma(\alpha_{j+1})}\left(q^I, \alpha_j\right) \quad 1 < j < i$$

$$\acute{q_1} = next_{\acute{m}_1 = \gamma(\alpha_2)}\left(q^I, \alpha\right)$$

$$\acute{\mathbf{q}}_{\mathbf{q},\alpha}\left(\alpha_i, \ldots, \alpha_2\right) = \left[\left(\acute{m}_1, \acute{q_1}\right), \ldots, , \left(m_i, \acute{q_i}\right), \ldots, \left(m_N, q_N\right)\right]$$

$$next_{L_i}\left(\mathbf{q}, \alpha\right) = \left\{\acute{\mathbf{q}}_{\mathbf{q},\alpha}\left(\alpha_i, \ldots, \alpha_2\right)\right\}_{\alpha_j \in \Sigma_{L_j}}$$

Once again, as states $q_1\ \ldots\ q_{i-1}$, as well as being finals, could lead to transitions, the destination metastate set is the union of transitions at each layer:

$$\acute{\mathbf{Q}} = \bigcup_{j=1}^{i} next_{L_j}\left(\mathbf{q}, \alpha\right)$$

The probability of each transition in the set $next_{L_i}\left(\mathbf{q}, \alpha\right)$ is the joint probability of being final for $q_1\ \ldots\ q_{i-1}$ (layers $1\ \ldots\ i-1$), transition $q_i \rightarrow \acute{q_i}$ (layer i) and initial transitions $q^I \rightarrow \acute{q_{i-1}}\ \ldots\ q^I \rightarrow \acute{q_1}$ (layers $i-1\ \ldots\ 1$):

$$p\left(\acute{\mathbf{q}}_{\mathbf{q},\alpha}\left(\alpha_i, \ldots, \alpha_2\right), \alpha \mid \mathbf{q}\right) = \prod_{j=1}^{i-1} p^F_{m_j}\left(q_j\right)\ \cdot\ p_{m_i}\left(\alpha_i \mid q_i\right)\ \cdot\ \prod_{k=i-1}^{1} p_{\acute{m}_k}\left(\alpha_k \mid q^I\right)$$

Final states

A metastate $\mathbf{q} = [(m_1, q_1), \ldots, (m_N, q_N)]$ will be accepting or final, if and only if all the states q_i are finals. Thus there exists a set $\mathbf{Q}_F$ of final metastates and their final probabilities are the joint final probabilities:

$$p^F\left(\mathbf{q} \in \mathbf{Q}^F\right) = \prod_{i=1}^{N} p^F_{m_i}\left(q_i\right)$$

As stated for Markov models, LMM final metastate paradigm can be understood as an *outgoing* transition, and thus all transitions, inner and outgoing, must sum to a total probability of 1:

$$p^F\left(\mathbf{q}\right) + \sum_{\alpha \in \Sigma} \sum_{\acute{\mathbf{q}} \in next(\mathbf{q}, \alpha)} p\left(\acute{\mathbf{q}}, \alpha \mid \mathbf{q}\right) = 1$$

Initial states

In contrast to common WFA and in order to maintain coherence with the formalism about LMM transitions explained before, there exists no such an only metastate $\mathbf{q} = [(m_1, q_1), \ldots, (m_N, q_N)]$ that could be the initial one.[4] Indeed, there exists a set $\mathbf{Q}^I$ of initial metastates, and all of them can be obtained in a similar way that was presented for transitions.

Starting at top level layer, all possible transitions from its only initial state (and the corresponding symbols α_N) lead to all possible destination states $\acute{q_N}$. Once again, each symbol is mapped to an underlying model at layer $N-1$, and all possible transitions starting at initial states at layer $N-1$ should be taken into account. The recursion ends at bottom level, where just initial states are considered, due to the fact that there is not any $\alpha = \alpha^{L_1}$ symbol to process. To summarize, the initial metastates are of the form:

$$\acute{q_N} = next_{m_N}\left(q^I, \alpha_N\right)$$

$$\acute{q_i} = next_{\acute{m}_i = \gamma(\alpha_{i+1})}\left(q^I, \alpha_i\right) \quad 1 < i < N$$

$$\acute{q_1} = q^I \in Q_{\acute{m}_1 = \gamma(\alpha_2)}$$

$$\mathbf{q}^I\left(\alpha_N, \ldots, \alpha_2\right) = \left[\left(\acute{m}_1, q^I\right), \ldots, \left(\acute{m}_{N-1}, \acute{q_{N-1}}\right), \left(m_N, \acute{q_N}\right)\right]$$

$$\mathbf{Q}^I = \left\{\mathbf{q}^I\left(\alpha_N, \ldots, \alpha_2\right)\right\}_{\alpha_j \in \Sigma_{L_j}}$$

[4] Actually, we could define a special state $\mathbf{q} = \left[(null, null), \ldots, (null, null), \left(m, q^I\right)\right]$ which's top layer's state should be the initial state of the Markov model and other layer's model and states would be *null*. Such state could easily lead to correct initial transitions, just starting to process them from the top layer down to the bottom one instead of as explained for other metastates (starting at bottom, up to the transition layer, then down to the bottom). Anyway, that should require to adapt the metastate transition formalism to include this exception.

Furthermore, each initial metastate owns an initial probability, which is the joint initial transition probability at each layer, and all initial metastates sum to a total probability of 1:

$$p^I\left(\mathbf{q} \in \mathbf{Q}^I\right) = \prod_{i=N}^{2} p_{\acute{m}_i}\left(\alpha_i \mid q^I\right) \qquad \sum_{\mathbf{q}\in\mathbf{Q}^I} p^I\left(\mathbf{q}\right) = 1$$

LMM-S AND HMM-S

In spite of the LMM formalism is versatile enough to add as much knowledge layers as we would like to, there is a fundamental restriction that should be stated: layers must contain Markov models. However, in state-of-the-art automatic speech recognition [2, 1, 5, 4, 7], hidden Markov models (HMM) are the uncontested model for the temporal decoding stage, mostly due to the fact that they are able to face up to temporal variations. Therefore, the novel architecture would be useless unless HMMs could be integrated in them.

One could suggest to relax the constraint about Markov models, just allowing nondeterministic WFA to be part of a layer (HMM should be handled as such an automaton). But a much more interesting solution can be stated: a HMM can be actually represented as a 2-layer LMM and a HMM set can be also translated into 2 knowledge layers, which can be straight added under LMMs' formalism.

HMM as a 2-layer LMM

A HMM [3] can be defined as a sextuple $h \equiv (Q, \Sigma, A, B, \Pi, \Phi)$, where Q is the set of states and Σ the output alphabet, $A = \{a_{q\to\acute{q}} = p\left(\acute{q} \mid q\right)\}$ refers to transition probabilities, $B = \{b_q\left(\alpha\right) = p\left(\alpha \mid q\right)\}$ to output (emission) probabilities, $\Pi = \left\{\pi_q = p^I\left(q\right)\right\}$ to initial probabilities and $\Phi = \left\{\phi_q = p^F\left(q\right)\right\}$ refers to final probabilities.

In essence, a HMM is a metaphor of a process whose inner states cannot be observed but some other indirect observables. The statistical dependence among these observables and the states is given by output probability distributions. Equivalently, in a LMM the only observables are bottom layer's symbols. Hence, given a sequence of such symbols we cannot determine a sequence of metastates (or states in the case of HMM), and must work out a sequence which is optimal in some meaningful sense [6].

The equivalent 2 *layer* LMM of a given $h \in H$ HMM (see Fig. **??**) is obtained simply splitting transitions and emissions: the former to the top layer and the latter to the bottom one. Thus, *layer* 2 (L_2) contains the knowledge about h topology, whereas the knowledge related to emissions is contained at *layer* 1 (L_1). Next, both layers will be briefly described.

The HMM and the LMM are both equivalents, so their symbol alphabets

should be the same. Thus, L_1's alphabet is h's alphabet:

$$\Sigma = \Sigma_{L_1} = \Sigma_H$$

Each emission distribution at states $q \in Q_h$ can be represented in L_1 by a 2 *states* Markov model, one state being initial and the other final. Therefore, there will be one model at L_1 per each state at model h, that is, there exists a mapping $\lambda : Q_h \rightarrow L_1$ that connects h states and L_1 models:

$$L_1 \equiv \{m = \lambda(q)\}_{q \in Q_h}$$

In each Markov model $m = \lambda(q)$ at L_1, the transition set $q^I \rightarrow q^F$ is equivalent to the emission probability distribution $b_q(\alpha)$ at state $q \in Q_h$:

$$\forall \alpha \in \Sigma\,, \quad next_{m=\lambda(q)}\left(q^I, \alpha\right) = q^F$$

$$p_{m=\lambda(q)}\left(\alpha \mid q^I\right) = b_q(\alpha)$$

At *layer* 2, there is only one Markov model $m \in L_2$ with all the knowledge about h's topology. One more state than those at h is needed: the extra state is the initial one, q^I, and there exists a mapping $\omega : Q_h \rightarrow Q_m - \left\{q^I\right\}$ among h states and m states (except q^I). All transitions that end at a mapped state $q_m = \omega(q_h)$ generate the same symbol $\alpha_2 \in \Sigma_{L_2}$, and this symbol is related to the Markov model at L_1 that modelizes the emission of the state q_h:

$$\forall q_m \in Q_m \quad \forall q_h \in Q_h\,, \quad next_m(q_m, \alpha_2) = \omega(q_h) \Leftrightarrow \gamma(\alpha_2) = \lambda(q_h)$$

Hence, transitions destination function is

$$next_m(q, \alpha) = \omega\left(\lambda^{-1}(\gamma(\alpha))\right)$$

and transition weights will have the value of HMM probabilities π_q and $a_{q \rightarrow \acute{q}}$, depending on source state at L_2 being initial or not:

$$p_m(\alpha \mid q) = \begin{cases} \pi_{\lambda^{-1}(\gamma(\alpha))}\,, & q = q^I \\ a_{\omega^{-1}(q) \rightarrow \lambda^{-1}(\gamma(\alpha))}\,, & q \neq q^I \end{cases}$$

Finally, the set of accepting (final) states will be the set of states related to h's final states, having the same probability:

$$Q_m^F = \{q = \omega(q_h)\}_{\phi_{q_h} > 0} \qquad p_m^F(q = \omega(q_h)) = \phi_{q_h}$$

The previous equivalence between a HMM and a 2 *layer* LMM can be generalized to a set of HMM. Each emission pattern at HMMs states is modelized by a two-state Markov model at *layer* 1, whereas each topology is represented by another model at *layer* 2; all transitions ending at the same state at *layer* 2 are related to the same symbol $\alpha_2 \in \sum$, which represents an emission pattern, that is, a Markov model at *layer* 1. These layers can be directly added as two knowledge levels into a LMM. In ASR, for example, acoustic models are usually HMMs, whereas at upper knowledge levels

(lexical or pronunciation models, language models, etc.) Markov models are typically used. Thus, using LMMs, a single model integrating all the knowledge can be defined. Such a model would be bottom-up formed by: two layers of acoustic knowledge (derived by a HMM set), one layer or lexical knowledge (pronunciation models) and a language model (n-grams, for example).

HMM2 as a 3-layer LMM

HMM2 [8, 9] extends HMM framework to simultaneously accommodate complex constraints in both the temporal and frequency domains. On this approach, multi-gaussians typically used in standard HMMs are replaced by frequency based HMM which perform frequency warping and integration. That is, a frequency based HMM is associated with each (temporal) HMM-state.

As stated for HMMs, a HMM2 can be converted into a LMM too. In this case 3 layers are needed: *layer* 3 represents the topology knowledge of the HMM2, whereas *layer* 2 and *layer* 1 arise from the already described decomposition of emission-HMMs into 2 knowledge layers (topology of emission-HMMs plus the emissions themselves). Therefore, HMM2 approach differs from traditional HMM in the sense that they add an extra knowledge layer: the frequency warping.

The fact that both HMM and HMM2 can be considered as particular instances of LMM, clarifies and suggests the search of alternative models based on the LMM paradigm.

ADDING TRAINING AND RECOGNITION PARADIGMS

In ASR, during the recognition and also the training[5] of the models, all knowledge sources are usually integrated on a single system. But as well as the models commonly considered to take part on such integrations, sometimes some extra knowledge is added to the system. For instance, when phonetic HMMs are trained from labeled (there is some information about what has been said) but unsegmented data (there is no information about when each phoneme starts or ends), some restrictions can be stated: the order of phonemes and the possibility of silence between them, for example. Even in a much simpler situation, at acoustic-phonetic decodification, the way phonetic units can be combined must be fixed : all phonetic strings could be equally probable, less probable as they are longer, and so on.

Whenever there is more than one class (model) at the top knowledge level of such a system, some extra information about those classes strings' probabilities must be added. This information can be easily formalized as an extra layer (containing only one Markov model) and integrated into a LMM, and thus, such recognition tasks can be converted to the standard recognition procedure using just one LMM. Next, some usual cases will be presented.

[5]By training, we mean the supervised recognition stage used for sentence segmentation.

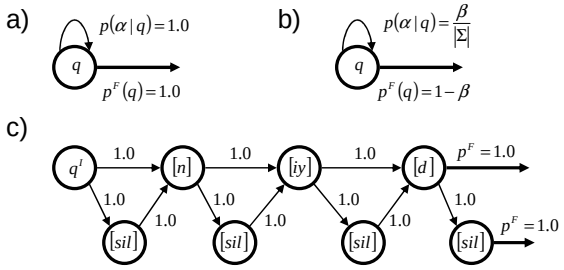

Figure 2: Recognition paradigm can be achieved adding a simple Markov model at LMM's top layer. Three different recognition paradigms are shown, *a)* equiprobable, *b)* length-dependent equiprobable and *c)* supervised (with silences insertion) for the word *need* ([*n*] [*iy*] [*d*]).

Equiprobable recognition

If there is no *a priori* knowledge about the classes' combination probability, all strings should be equally probable, $P(S_i) = P(S_j) = \frac{1}{|\{S\}|}$. As all possible strings range $|\{S\}|$ depends on the input-data maximum size bound, an easy solution is to consider an unnormalized distribution such as $P(S_i) = P(S_j) = 1.0$. This distribution can be achieved by a one state Markov model, with as much transitions over itself as classes to combine. All probabilities, transition and final, are 1.0 valued:

$$next(q, \alpha) = q \qquad p(\alpha \mid q) = 1.0 \qquad p^F(q) = 1.0$$

Length-dependent equiprobable recognition

Many times, the equiprobable paradigm is modified in such a way that only strings of same length are equiprobable, and the probability decreases as the string length goes longer. Such a distribution can be achieved by another one state Markov model, where all transitions have the same probability and final probability is just the necessary to add up to one. Actually, this is the most used paradigm at unsupervised recognition, may be due to the fact that it is the simplest normalized case to implement:

$$next(q, \alpha) = q \qquad p(\alpha \mid q) = \frac{\beta}{|\Sigma|} \qquad p^F(q) = 1 - \beta$$

Supervised recognition for training segmentation

In ASR, when some models (acoustic HMMs, for instance) are trained, a portion of the labeled training data can be manually segmented in order to initialize those models and afterwards use themselves to segment the rest of the data for more training iterations. But in absence of manually segmented data, that is, starting with random models, trained parameters hardly converge to

good values unless recognition constraints are enough relaxed. A valid solution is to allow the insertion of silence between any labeled phonemes (silences are supposed not to be labeled) and let all possible results be equiprobables (once again, an unnormalized distribution will be useful). Nevertheless, in both cases there is a supervised recognition in order to obtain a segmentation for parameter training. Such a supervision can be easily achieved by a simple Markov model with at least as much states as labeled phonemes (or twice for the silence introduction case).

CONCLUSION

A new architectural approach to automatic speech recognition has been proposed. Both HMM and HMM2 have been presented as particular instances of LMM, suggesting the possibility of alternative models based on the LMM paradigm. Finally, it has been stated that recognition paradigms, also used for model training, can be integrated as an extra knowledge layer, so, they are all reduced to the one model standard recognition procedure.

References

[1] L. Bahl, F. Jelinek and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," **IEEE Trans. Pattern. Anal. Machine Intell.**, vol. PAMI-5, pp. 179–190, 1983.

[2] J. K. Baker, "The dragon system – An overview," **IEEE Trans. Acoust. Speech Signal Processing**, vol. ASSP-23, no. 1, pp. 24–29, 1975.

[3] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," **Ann. Math. Stat.**, vol. 37, pp. 1554–1563, 1966.

[4] S. E. Levinson, "Structural methods in automatic speech recognition," **Proceedings of the IEEE**, vol. 73, no. 11, pp. 1625–1650, 1985.

[5] S. E. Levinson, L. R. Rabiner and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," **Bell Sys. Tech. J.**, vol. 62, no. 4, pp. 1035–1074, 1983.

[6] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," **Proceedings of the IEEE**, vol. 77, no. 2, pp. 237–240, 1989.

[7] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," **EEE Magazine on Accoustics, Speech and Signal Processing**, vol. 3, no. 1, pp. 4–16, 1986.

[8] K. Weber, S. Bengio and H. Bourlard, "HMM2- A novel approach to HMM emission probability estimation," in **Proceedings of the International Conference on Speech and Language Processing**, Beijing, China, 2000.

[9] K. Weber, S. Ikbal, S. Bengio and H. Bourlard, "Robust Speech Recognition and Feature Extraction Using HMM2," **Computer, Speech and Language**, vol. 17, no. 2-3, pp. 195–211, 2003.