STUDIA PHONOLOGICA XXIV (1990)

Automatic Extraction of Phonotactics based on Hidden Markov Models and Language Identification

Seiichi NAKAGAWA and Yoshio UEDA

ABSTRACT

Natural languages were modeled popularly by Markov models. In this paper, they were modeled by HMM (Hidden Markov Model). As applications, the identification of language and the prediction for phoneme/syllable/word-category were performed using HMM.

The results show that the HMM extracts automatically the phonotactics and it has also a performance better than the first order Markov model (bigram) and almost the same as the second order Markov model (trigram) on the entropy.

From the results, we believe that HMM is useful not only speech recognition but also natural language processing.

1 INTRODUCTION

For speech recognition, an accurate word recognition system needs certain linguistic knowledges such as syntax, semantics and pragmatics, because it is difficult to recognize words using only their acoustical characteristics. So, the system performance becomes better with linguistic knowledges according to the decrease of a search space. In other words, the higher correct prediction rate corresponds to the smaller search space.

First of all, we use the ergodic HMM (Hidden Markov Model) for modeling the alphabetical sequence of natural language and investigate whether the HMM extracts automatically the phonotactics of the language or not.

Next, we describe a language identification method and a prediction method of phoneme/syllable/word-category using HMM to correct recognition errors, and compare HMM with traditional Markov models (bigram, trigram) on the entropy and the performance.

The ways of a hidden Markov modeling of the languages, they are same either identification or prediction, refer to the papers of N. Huang [1] and R. Cave and

Seiichi NAKAGAWA (中川聖一): Professor, Department of Information and Computer Sciences, Toyohashi Univeristy of Technology, Toyohashi, 441 Japan.
Yoshio UEDA (上田佳央): Graduate Student, Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, 441 Japan
The authors have done the research under the direction of Dr. Shuji Dohshita, Professor of Information Science, Kyoto University

L. Neuwirth [2].

A language identification has been studied by many researchers. P. Henrich [3] studied to identify words into three language (Germany, English, French) with rules. He obtained almost the same performance as the result using a neural network. A. House and E. Neuburg [4] used eight phonetic texts which were reduced to 4-character alphabets from 26 alphabetieal letters and these samples were used to form N-state statistical models of each language. However, they did not experiment on the identification. R. Cole, et al. [5] studied one by using a neural network with aeoustical distribution of stop consonants.

M. Nakamura and K. Shikano [6] have been studied a word category prediction with a N-gram neural network. They obtained almost the same performance as the results using statistical trigram model.

We obtain a result of correct language identification rate and correct phoneme/syllable/word-category prediction rate by using a 7-state HMM. This result is as same as that using a traditional second order Markov model on the entropy and as same as that using a traditional first order Markov model on the correct rate. The more number of states, the better result will be obtained.

2 TEXT DATABASE

For the extraction of phonotactics, identification of language and prediction for phonemes, texts of six languages (English, French, Germany, Italian, Japanese, Spanish) were used. For each language, there were about 30000 characters for training data and 1000 characters for test data with an alphabet (26 letters).

For the prediction of syllable, only Japanese text was used. In almost cases, Japanese syllable consists of a vowel and a syllable. So there were 15000 syllables for training data and 500 syllables for testing data with 109 kinds of syllables.

For the prediction of word-category, there were 1024 sentences (about 24 word per sentence) for training data and another 1024 sentences for testing extracted from the Brown Corpus which was English text database. The words had been classified into 89 kinds of category.

3 HMM TOPOLOGY

In this study, HMM is used a full structured (ergodic) model that any state can transit to all states as shown in Fig. 1.

For each number of states $S=2,3,5,7,10$, HMM was trained for the model of a phoneme/syllable/word-category sequence using many sentences for each language, and experimented in the identification and prediction about other sentences. The Baum-Welch (Forward-Backward) algorithm was used to train them. And the Forward path algorithm was used to identify and predict them. The HMM with

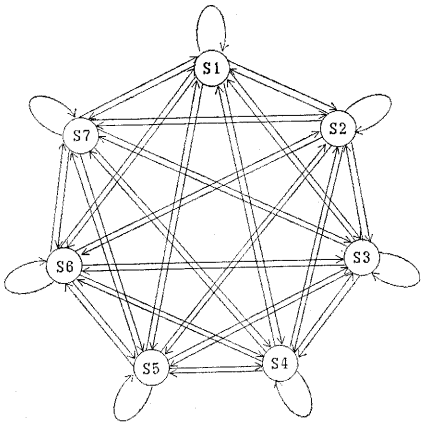

Fig. 1. A seven state ergodic hideen Markov model.

S-state consists of an S*S transition matrix, an (S*S)*(number of symbols) output probability matrix and an S*1 stationary state probability matrix (vector) for the model parameters.

4 Automatic Extraction of Phonotactics

Table 1 and Fig. 2 show the trained HMMs' parameters for English and Japanese phonemes, respectively. Here show HMMs with only the number of states S=2,3. HMMs for other langugaes and S=5,7 are not shown because of too many tables and figures (Note that for lower probabilities less than 0.000001 are represented by '——', and Japanese special syllables are represented by '*x*' for the syllabic nasal and by '*q*' for the choked sound.) There are some interesting results on these trained HMMs.

First, for output probabilities, arcs are classified into about two categories of vowels and consonants. This tendency is found for all languages. For example, for the 3-state HMM for Japanese, the occurrence probabilities of consonant are 0.9949 on the transition from the state-2 to the state-1, and 0.9936 on the transition from the state-2 to the state-0, respectively, on the other hand, those of vowel are 0.0051 and 0.0064, respectively.

Secondly, HMMs for Japanese have a clear structure. For the 2-state HMM for Japanese, only transition probabilities of between the state-0 and state-1 are higher, because Japanese usually appears vowel and consonant alternately, that is, almost all Japanese syllables consist of a consonant and a vowel. For the 3-state HMM for Japanese from the syllabic point of view, the state-2 may be a starting point. In other words, the transition to the state 2 occurs at the syllable boundary. Therefore almost all Japanese syllables are represented by the state transition for 2→0→2 or 2→1→2 as consonant(C)-vowel(V), for 2→0→0→2 as C-y (contracted sound)-V and for 2→2 as syllabic nasal and choked sound, here notice that we

Table 1. Parameters for trained HMM.

(a) 2-state HMM for Japanese

Stationary State Probabilities

state	0	1
probability	.5261	.4739

Transition Probabilities

from\to	0	1
0	.1375	.8625
1	.9119	.0881

Output Probabilities

alphabet	transition			
	0→0	0→1	1→0	1→1
a	.0049	.0004	.3203	.2448
b	-----	.0253	-----	-----
c	-----	-----	-----	-----
d	.0010	.0527	-----	-----
e	.0596	-----	.1247	.1191
f	-----	.0003	-----	-----
g	-----	.0374	-----	-----
h	-----	.0623	-----	.0001
i	.3225	.0033	.1848	.0816
j	-----	-----	-----	-----
k	-----	.1324	-----	-----
l	-----	-----	-----	-----
m	-----	.0822	-----	-----
n	-----	.1312	-----	-----
o	.0065	.0016	.2300	.3054
p	-----	.0037	-----	-----
q	.2048	-----	-----	-----
r	-----	.1070	-----	-----
s	-----	.1029	-----	-----
t	-----	.1672	-----	-----
u	.1927	.0003	.1402	.0568
v	-----	-----	-----	-----
w	-----	.0400	-----	-----
x	.2080	.0003	-----	-----
y	-----	.0247	-----	.1922
z	-----	.0246	-----	-----

(c) 2-state HMM for English

Stationary State Probabilities

state	0	1
probability	.4599	.5401

Transition Probabilities

from\to	0	1
0	.4034	.5966
1	.5952	.4048

Output Probabilities

alphabet	transition			
	0→0	0→1	1→0	1→1
a	-----	.2518	-----	.0525
b	.0334	.0006	.0305	.0009
c	.0086	-----	.0496	-----
d	.0012	-----	.0925	.1178
e	.1334	.3269	-----	.0033
f	.0121	-----	.0462	.0263
g	.0003	.0003	.0473	.0194
h	.3715	-----	.0057	-----
i	.0210	.1477	-----	.0583
j	.0002	-----	.0019	-----
k	.0070	.0048	.0132	.0040
l	.0406	-----	.0369	.1044
m	.0119	-----	.0741	.0014
n	.0031	-----	.0696	.2375
o	.0206	.2203	-----	.0249
p	.0185	.0000	.0187	.0246
q	.0009	-----	.0027	-----
r	.0551	-----	.0902	.0975
s	.0284	-----	.1214	.0819
t	.1336	-----	.1975	.0250
u	.0084	.0208	.0010	.0790
v	-----	-----	.0240	-----
w	.0672	-----	.0482	.0136
x	-----	-----	-----	.0021
y	.0230	.0269	.0276	.0255
z	-----	-----	.0010	-----

point out only the higher probability. The diphthong which is represented by the transition of C-V-V (Note there are a few cases of the form of C-y-V-V (ex. kyou) and x-V (ex. taxi)) has many representations by the state transition for 2→0→0→2, 2→0→1→2, 2→0→2→2, 2→1→0→2, 2→1→1→2 or 2→1→2→2.

Thirdly, although HMMs for English don't have a clear structure except for the vowel transition. For the 2-state HMM, the vowel is observed when the state transits from the state-0 to the state-1. For the 3-state HMM, the vowel is observed when the state transits to the state-1 from the other states.

Additionally, the occurrence probabilities of Japanese syllables are calculated by the 3-state HMM and compared with ones which are obtained by counting the syllables in the Japanese text from the database. Table 2 summarizes the occurrence probability of the Japanese syllables. Only the representative transitions of syllables from the state 2 are taken into consideration on HMM. As seen in Table 2, almost all probabilities by HMMs are consistent with ones by statistics of texts each other.

Table 1 continued.

(b) 3-state HMM for Japanese

Stationary State Probabilities

state	0	1	2
probability	.2220	.4979	.2801

Transition Probabilities

from \ to	0	1	2
0	.0713	.0182	.9105
1	.0822	.0347	.8832
2	.5489	.3223	.1287

Output Probabilities

alphabet	transition								
	0→0	0→1	0→2	1→0	1→1	1→2	2→0	2→1	2→2
a	.0488	.4659	.2259	.0195	.8135	.5097	.0000	.0000	-----
b	-----	-----	-----	-----	-----	-----	.0400	.0004	-----
c	-----	-----	-----	-----	-----	-----	-----	-----	-----
d	-----	-----	-----	-----	-----	-----	.0012	.1411	-----
e	.0200	.0026	.0648	.3564	.0781	.2347	-----	.0000	.0544
f	-----	-----	-----	-----	-----	-----	.0005	-----	-----
g	-----	-----	-----	-----	-----	-----	.0157	.0746	-----
h	-----	-----	-----	.0052	-----	-----	.0514	.0806	-----
i	.0009	.4664	.2834	.0003	.1083	.0001	.0031	.0042	.3522
j	-----	-----	-----	-----	-----	-----	-----	-----	-----
k	-----	-----	-----	-----	-----	-----	.1792	.0532	-----
l	-----	-----	-----	-----	-----	-----	-----	-----	-----
m	-----	-----	-----	-----	-----	-----	.0602	.1200	-----
n	-----	-----	-----	-----	-----	-----	.1993	.0155	-----
o	.4218	-----	.1995	.6164	.0000	.2554	.0019	-----	.0039
p	-----	-----	-----	-----	-----	-----	.0044	.0024	-----
q	-----	-----	-----	-----	-----	-----	-----	-----	.2212
r	-----	-----	-----	-----	-----	-----	.1235	.0792	-----
s	-----	-----	-----	-----	-----	-----	.1634	-----	-----
t	-----	-----	-----	-----	-----	-----	.1047	.2741	-----
u	.1350	.0641	.2264	-----	-----	.0000	-----	.0012	.1444
v	-----	-----	-----	-----	-----	-----	-----	-----	-----
w	-----	-----	-----	-----	-----	-----	.0000	.1081	-----
x	-----	-----	-----	-----	-----	-----	-----	.0010	.2239
y	.3736	.0009	-----	.0022	-----	-----	.0123	.0444	-----
z	-----	-----	-----	-----	-----	-----	.0391	-----	-----

Of course, the occurrence probabilities are calculated very easily. They are obtained by the multiplication of the output probabilities and the transition probabilities. For example, for the syllable C-V, P(C-V|2→0→2) for the path 2→0→2 is $0.4972=0.9949\times0.5489\times1.0000\times0.9105$, P(C-V|2→1→2) for the path 2→1→2 is $0.2828=0.9936\times0.3223\times1.0000\times0.8832$, thus P(C-V) is given approximately by the sum of P(C-V|2→0→2) and P(C-V|2→1→2), that is, $0.4972+0.2828=0.7800$.

5 Entropy

For each HMM M which represents a language model, the entropy $H(M)$ is computed. When $P(y|i)$ is a conditional output probability where y is an observed symbol and i is a state, $H(Y|i)$ which is the entropy at the state i of this model is represented as:

Table 1 continued.

(d) 3-state HMM for English

Stationary State Probabilities

state	0	1	2
probability	.2220	.4979	.2801

Transition Probabilities

from\to	0	1	2
0	.0698	.7250	.2052
1	.3092	.3393	.3515
2	.3593	.4124	.2283

Output Probabilities

alphabet	transition								
	0→0	0→1	0→2	1→0	1→1	1→2	2→0	2→1	2→2
a	.0014	.2447	.0004	-----	.0321	.0000	.0023	.2961	.0003
b	.0635	.0034	.0007	.0415	.0071	.0011	.0513	.0138	.0065
c	.0111	-----	.0004	.0401	-----	.0501	-----	-----	.0389
d	.0001	.0113	.0344	.0800	.1115	.1169	-----	-----	-----
e	.3210	.4113	.4289	-----	.0000	-----	.0427	.1178	.0088
f	.0000	-----	.0166	.0490	.0282	.0422	.0002	-----	.0382
g	.0113	.0022	.0002	.0016	.0139	.0883	.0000	.0006	.0121
h	.0000	-----	-----	.0029	.0004	.0020	.7317	-----	.0368
i	.2132	.1476	-----	-----	.0697	-----	.0127	.1359	.0075
j	-----	-----	-----	.0004	-----	.0027	.0006	-----	.0009
k	.0450	.0017	.0016	.0228	.0046	-----	.0111	.0113	.0001
l	.1787	.0173	.0944	.1299	.0476	.0009	.0264	-----	.0008
m	.0438	-----	.0008	.1302	.0025	.0168	.0218	-----	.0013
n	.0003	-----	.0087	.1218	.2756	.0536	.0000	-----	.0000
o	.0000	.1328	.0339	-----	.0005	-----	.0000	.3610	.0983
p	.0015	.0000	.0192	.0189	.0281	.0128	.0000	.0002	.0637
q	-----	.0001	-----	-----	.0046	-----	-----	.0022	-----
r	.0100	-----	.0214	.1469	.1201	.0442	.0601	-----	.0560
s	.0733	.0000	.0071	.0858	.0880	.1542	.0019	-----	.0898
t	.0000	-----	.2395	.0021	.0176	.3434	-----	-----	.3050
u	.0000	.0130	-----	.0092	.1035	.0042	.0053	.0264	-----
v	-----	-----	-----	.0504	-----	-----	-----	-----	-----
w	.0247	-----	.0194	.0561	.0138	.0262	.0190	-----	.2023
x	-----	-----	-----	-----	.0028	-----	-----	-----	-----
y	.0009	.0145	.0725	.0082	.0275	.0406	.0128	.0349	.0327
z	-----	-----	-----	.0021	-----	-----	-----	-----	-----

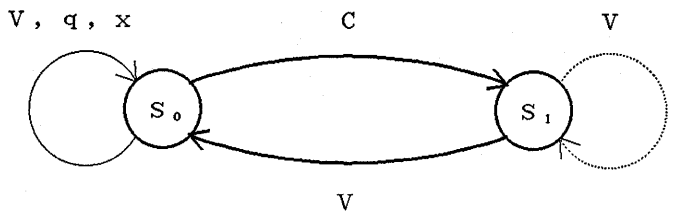

Fig. 2(a). Representative transitions in the 2-state HMM for Japanese
V and C denote a set of vowles and consonants, respectively.
⟶ corresponds to the transition with the probability
1.0 ~0.5
⟶ 0.5 ~0.25
⟶ 0.25 ~0.125
⟶ 0.125~0.000

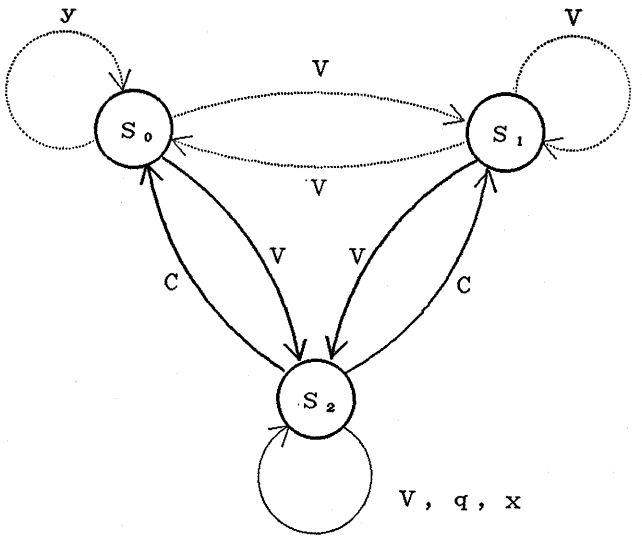

Fig. 2(b). Representative transitions in the 3-state HMM for Japanese.

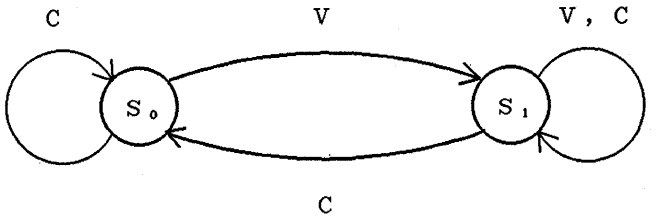

Fig. 2(c). Representative transitions in the 2-state HMM for English.

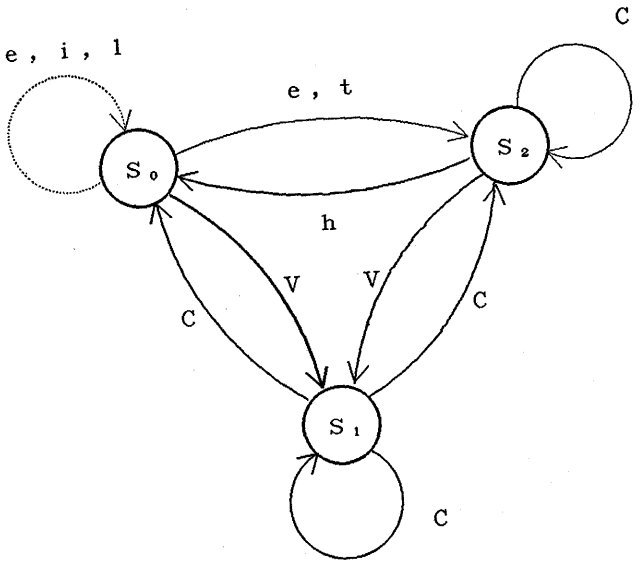

Fig. 2(d). Representative transitions in the 3-state HMM for English.

Table 2. The occurrence probabilities of Japanese syllables.

(a) Calculation by the 3-state HMM for Japanese

syllable	transition	multiple of probabilities	total
C-y-V	2-0-0-2	.9949×.5489 × .3736×.0713 × 1.0×.9105 = .01324	.0133
	2-0-1-2	.9949×.5489 × .0009×.0182 × 1.0×.8832 = .79e-5	
	2-1-0-2	.9936×.3223 × .0022×.0822 × 1.0×.9105 = .53e-4	
C-V-V	2-0-0-2	.9949×.5489 × .6265×.0713 × 1.0×.9105 = .02221	.1220
	2-0-1-2	.9949×.5489 × .9990×.0182 × 1.0×.8832 = .00877	
	2-0-2-2	.9949×.5489 × 1.0×.9105 × .5549×.1287 = .03551	
	2-1-0-2	.9936×.3223 × .9926×.0822 × 1.0×.9105 = .02379	
	2-1-1-2	.9936×.3223 × 1.0×.0347 × 1.0×.8832 = .00981	
	2-1-2-2	.9936×.3223 × 1.0×.8832 × .5549×.1287 = .02199	
C-V	2-0-2	.9949 × .5489 × 1.0 × .9105 = .4972	.7800
	2-1-2	.9936 × .3223 × 1.0 × .8832 = .2828	
q	2-2	.2212 × .1287 = .0285	.0285
x	2-2	.2239 × .1287 = .0288	.0288

(b) Summary of probabilites by the statistics and by the 3-state HMM

Note that for the diphthong is contained in C-V-V

syllable	statistics		HMM
	frequency	unigram	
C-y-V	229	0.0136	0.0133
C-V-V	2188	0.1302	0.1220
C-V	12904	0.7678	0.7800
C-y-V-V,x-V	591	0.0352	-----
q	446	0.0265	0.0285
x	448	0.0267	0.0288
total	16806	1.0000	0.9726

$$H(Y|i) = -\sum_{y} P(y|i) \log_2 P(y|i)$$

$$P(y|i) = \sum_{j} a_{ij} b_{ij}(y)$$

where a_{ij} is the transition probability that the state transits form the state i to the state j, $b_{ij}(y)$ is the output probability that the symbol y is obserbed through the state transition from i to j. $H(M)$ is the sum of $H(Y|i)$ multiplied by the stationary probability for each state, such as:

$$H(M) = \sum_{j} \pi(i) H(y|i)$$

where $\pi(i)$ is the stationary probability for the state i.

Besides, the entropy in a traditional Markov model which represents a given text is also computed. First, the entropy for unigram which assumes that the stationary probability $P(i)$ of each symbol i is independent each other is represented as:

$$F_1 = -\sum_{i} P(i) \log_2 P(i)$$

Then, the entropy for the first order Markov model (bigram) which assumes the independence of probability for each couple of symbols is represented as:

$$F_2 = -\sum_{i,j} P(i,j) \log_2 P(i|j)$$

Last, the entropy for the second order Markov model (trigram) which assumes the independence of probability for each trio of symbols is represented as:

$$F_3 = -\sum_{i,j,k} P(i,j,k) \log_2 P(i|j,k)$$

The entropy in HMM computed for each number of states $S = 2,3,5,7,10$ is summarized in Table 3. When the entropy is compared with traditional Markov models (see Table 4) by using the same text, it seems that HMM is better than the first order Markov model and almost the same as the second order Markov model.

Here, the trigram entropy for syllables in Table 4 was extremely small because the training data was too small to estimate many Markov model's parameters. From this, we think that HMM's parameters may be estimated even if the training data are not so many. Table 5 illustrates the number of parameters.

For modeling the alphabetical text, the number of free parameters of traditional Markov model has a N-th power of number of symbols, 676 as for bigram, 17576 as for trigram. When HMM is used, it is in proportion to about number of arcs times number of symbols, 1330 as for the 7-state HMM.

Therefore, in view of the capacity of information storages, HMM can sharply condense the information than traditional Markov models.

Table 3. Entropy in hidden Markov model.

Number of states		2	3	5	7	10
alphabet (phoneme)	English	3.63	3.48	3.23	3.02	—
	French	3.39	3.10	2.97	2.97	
	Germany	3.60	3.28	3.04	2.75	
	Italian	3.47	3.36	3.08	2.95	
	japanses	3.14	3.01	2.99	2.45	
	Spanish	3.37	3.24	2.97	2.65	
syllable	Japanses	5.31	5.06	4.81	4.60	4.65
word-categofy	English	4.05	3.61	3.34	3.11	2.97

Table 4. Entropy in Markov model.

Order of Markov model		F_1	F_2	F_3
alphabet (phoneme)	English	4.14	3.49	2.79
	French	3.99	3.40	2.91
	Germany	4.08	3.36	2.75
	Italian	4.01	3.37	2.94
	Japanese	3.93	3.06	2.77
	Spanish	4.03	3.38	2.94
syllable	Japanese	5.69	4.72	2.40
word-category	English	4.65	5.57	3.30

Table 5. Number of parameters.

	HMM			F_1	F_2	F_3
	5	7	10		bigram	trigram
alphabet	680	1330	2710	26	676	17576
syllable	2755	5397	11010	109	11881	1295029
word-category	2255	4417	9010	89	7921	704969

6 PREDICTION FOR PHONEME/SYLLABLE/WORD-CATEGORY

6.1 *Method*

The prediction is to predict the next symbol y_{t+1} when any length of symbol sequence $y_1, y_2, \cdots, y_t$ is given. This procedure is shown in the following:

1. For each symbol k which is used in a given model, calculate the probability $P(k|y_1, y_2, \cdots, y_t)$.
2. Sort these probabilities such that this order corresponds to the candidate order for y_{t+1}.

In this procedure, the probability $P(k|y_1, y_2, \cdots, y_t)$ is calculated by using the Forward path algorithm. We must notice that any state may become the initial state. First, $\alpha(i, t)$ is calculated by a given symbol sequence $y_1, y_2, \cdots, y_t$ using the Forward path algorithm, where $\alpha(i, t)$ is the probability when symbols were observed as $y_1, y_2, \cdots, y_t$ and the state was transited to i at the same time. Then normalize the $\alpha(i, t)$ to $\sum \alpha(i, t)=1$. Finally, calculate the $P(k|y_1, y_2, \cdots, y_t)$ as:

$$P(k:y_1, y_2, \cdots, y_t) = \sum_{i,j} \alpha(i, t)a_{ij}b_{ij}(k)$$

6.2 *Experimental result*

For testing data, a speech-unit (phoneme/syllable/word-cateogry) was predicted by using HMM, and the average correct prediction rate was calculated. The results of the tests are given in Tables 6(a), 7(a), 8(a).

To compare them, the predictions using traditional Markov models (bigram, trigram) were experimented in the same data as using HMMs. Results are given in Tables 6(b), 7(b), 8(b).

Here, the average correct prediction rate for a given test data in the second order Markov model (trigram) had very worse than one for a given training data (see Tables 7(b), 8(b)), because of lack of training data.

The results show that the 7-state HMM has the same performance as bigram and worse performance as trigram for the prediction of phonemes (alphabet). And it falls short of our expectations, because we think on the basis of model's entropies

Table 6. Average corre prediction rate for Japanese phoneme (alphabet) (%).

(a) HMM (7-state model)

	Number of candidates			
	1	3	5	10
training data	26.6	62.1	73.0	91.1
test data	26.0	56.0	73.9	88.2

(b) Markov model

	N-gram	Number of candidates			
		1	3	5	10
training data	2	23.5	56.0	76.8	93.2
	3	35.2	68.6	85.7	94.5
test data	2	26.3	54.9	73.0	88.3
	3	29.8	60.4	77.9	91.2

Table 7. Average correct prediction rate for Japanese syllable (%).

(a) HMM (10-state model)

	Number of candidates			
	1	3	5	10
training data	17.1	31.7	42.0	55.0
test data	12.9	27.4	35.4	50.4

(b) Markov model

	N-gram	Number of candidates			
		1	3	5	10
training data	2	12.4	29.8	40.4	64.0
	3	34.8	65.2	77.6	92.4
test data	2	14.7	29.0	40.7	54.4
	3	24.1	38.2	44.6	50.0

Table 8. Average correct prediction rate for English word category (%).

(a) HMM (10-state model)

	Number of candidates			
	1	3	5	10
training data	31.2	55.5	66.9	84.0
test data	31.2	52.5	62.1	78.0

(b) Markov model

	N-gram	Number of candidates			
		1	3	5	10
training data	2	31.6	54.4	69.6	87.1
	3	35.7	65.0	76.4	90.9
test data	2	29.9	52.9	64.9	80.2
	3	29.9	50.8	60.8	72.8

(Tables 2, 3) that the 7 or 10-state HMM has the better performance than bigram and the same performance as trigram.

7 Identification of Languages

7.1 *Method*

First, the given model is trained by the training data for each language. Next, the identification of language is given by the following procedure:

1. For each model, calculate the probability of a sequence of observation symbols as same as a training sequence, where every states may be both initial and final states.
2. Choose the model which has the highest probability.
3. A chosen model is regarded as the correct language model for the given text.

7.2 *Experimental result*

Many test sequences with a constant length which were taken from a testing text were identified, and the average identification rate was obtained. Spaces in text sequences were eliminated. For example, the following sequences show the examples of original text and the training or test set of 20 letters of length;

Table 9. Confusion of language identification matrix using 7-state HMM.

text	20 letters						50 letters					
	E	F	G	I	J	S	E	F	G	I	J	S
English	49	1	0	0	0	0	20	0	0	0	0	0
French	1	42	1	2	0	4	0	19	0	1	0	0
Germany	1	1	48	0	0	0	0	0	20	0	0	0
Italian	0	1	0	41	0	8	0	0	0	20	0	0
Japanese	0	0	0	0	50	0	0	0	0	0	20	0
Spanish	0	2	0	3	0	45	0	0	0	0	0	20

Table 10. Average identification rate using various HMMs (%).

Number of state	2	3	5	7
5 letters	54.2	56.1	57.8	58.8
10 letters	66.0	70.8	75.3	76.8
20 letters	84.3	86.7	89.0	91.7
30 letters	89.9	90.4	94.9	95.0
50 letters	97.5	96.7	97.5	99.2
100 letters	96.7	100.0	100.0	100.0

original: "For speech recognition, an accurate word recognition system needs..."
training or test set: "forspeechrecognition", "naccuratewordrecogni", etc.

Table 9 shows the confusion matrix using the 7-state HMM when the test sequence length has 20 letters or 50 letters. We are able to find the similarity between sister languages such as Italian and Spanish.

Table 10 shows the average identification rate using HMMs with various numbers of states S=2,3,5,7. In these results, it seems that the more number of letters in a test sequence has and the more number of states in HMM has, the more increase an average identification rate becomes. In addition, an average identification rate became about 100% when the 7-state HMM was used and the test sequence had 50 letters (about one sentence).

8 CONCLUSIONS

First, natural languages were modeled by HMMs and Markov models. From these trained HMMs, we found that each HMM extracted automatically the phonotactics of the given language. Then, each model's entropy was calculated. And the identification of language and the prediction for phoneme/syllable/word-category were experimented.

The results show that the HMM had a performance better than the first order Markov model (bigram) and worse than the second order Markov model (trigram). Of course, these results depend on the number of states in HMM.

From the results, we believe that HMM is useful to natural language processing as same as speech recognition, for example, the language identification of each word in the text which is interspersed foreign languages, or a stochastic language model.

Acknowledgements

Authors would like to thank to Dr. Shigeki Sagayama and colleague of ATR Interpreting Telephony Research Laboratories for giving us convenience of using computers and the speech database.

References

[1] Nai-Kuan Huang: "A Lerning Experiment on English Spelling Rules", ICNNN-87, II-351.
[2] R. Cave and L. Neuwirth: "Hidden Markov Models for English", Proc. Symbo. on the Application of Hidden Markov Models to Text and Speech, ed. J.D. Ferguson, Princeton, pp. 16–56 (1980).
[3] P. Henrich: "Language Identification for the Automatic Grapheme-to-Phoneme Conversion of Foreign Words in a German Text-to-Speech System", Speech-89, pp. 220–223 (1989).
[4] A. House and E. Neuburg: "Toward Automatic Identification of the Language of an Utterance. I. Preliminary Methodological Considerations", J. Acoust. Soc. Am., Vol. 62, No. 3, pp. 708–713 (1977).
[5] R. Cole, et al.: "Language Identification with Neural Networks: a Feasibility Study", Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp. 525–529 (1989).
[6] M. Nakamura and K. Shikano: "English Word Category Prediction Based on Neural Networks", Proc. ICASSP, pp. 731–734 (1989).