ISCA Archive
http://www.isca-speech.org/archive

First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

PREDICTION FOR PHONEME/SYLLABLE/WORD-CATEGORY AND IDENTIFICATION OF LANGUAGE USING HMM

Yoshio Ueda and Seiichi Nakagawa

Toyohashi University of Technology
Tempaku-cho, Toyohashi, 441, Japan

ABSTRACT

Natural languages were modeled popularly by Markov models.

In this paper, natural languages were modeled by HMM (Hidden Markov Model). And the identification of language and the prediction for phoneme/syllable/word-category were performed using HMM.

The results show that the HMM had 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 not useful 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 by the decrease a search space. In other words, the higher correct prediction rate corresponds to the smaller search space.

We describe a language identification method and a prediction method of phoneme/syllable/word-category using HMM (Hidden Markov Model) to correct word 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 L.Neuwirth [2].

A language identification have 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 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 distribution of stop consonants from acoustic speech.

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 many number of states, the better result will be obtained.

2 TEXT DATABASE

For the identification of language and prediction for phonemes, texts of six languages (English, French, Germany, Itarian, 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 prediction of syllable, only Japanese text were 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 prediction of word-category, there were 1024 sentence (about 24 words per sentence) for training data and another 1024 sentence 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

HMM used in this study is a full structured (ergordic) model that any state can transit to all states (see Fig.1).

For each number of states S=2,3,5,7, HMM was trained for the model of a phoneme/syllable/ word-category sequence using many sentences, and experimented in the identification and a prediction about another 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 S-state consists of an S*S transition matrix, an (S*S)*(number of symbols) output probability matrix and an S*1 initial stationary matrix (vector) for the model parameters.

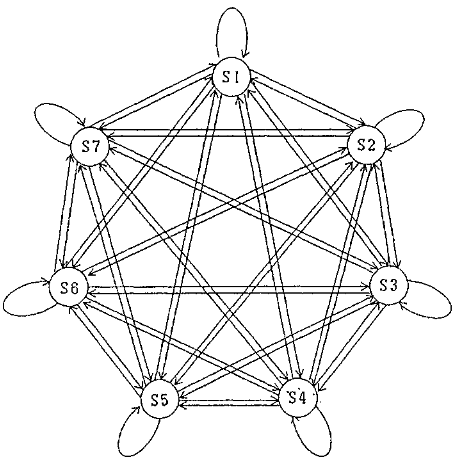

Fig.1 A seven state ergodic hidden Markov model

4 ENTROPY

For each HMM M which represented a language model, the entropy H(M) was 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:

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

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

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 output 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 represented for a given text was 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:

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

Table 1 Entropy in hidden Markov model

Number of states		2	3	5	7
alphabet (phoneme)	Eng.	3.63	3.48	3.23	3.02
	Fre.	3.39	3.10	2.97	2.97
	Ger.	3.60	3.28	3.04	2.75
	Ita.	3.47	3.36	3.08	2.95
	Jap.	3.14	3.01	2.99	2.45
	Spa.	3.37	3.24	2.97	2.65
syllable	Jap.	5.31	5.06	4.81	4.60
word-category	Eng.	4.05	3.61	3.34	3.11

Table 2 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	3.57	3.30

Table 3 Number of parameters

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

The entropy in HMM computed for each number of states S=2,3,5,7, is summarized in Table 1. When the entropy was compared with traditional Markov models (see Table 2) by using the same text. It seems that HMM is more 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 2 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 are able to estimate even if the training data are not so many. Table 3 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. And 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, HMM can sharply condense the information than traditional Markov models.

Table 3 Confusion matrix using 7-state HMM

	20 letters						50 letters					
text	E	F	G	I	J	S	E	F	G	I	J	S
Eng.	49	1	0	0	0	0	20	0	0	0	0	0
Fre.	1	42	1	2	0	4	0	19	0	1	0	0
Ger.	1	1	48	0	0	0	0	0	20	0	0	0
Ita.	0	1	0	41	0	8	0	0	0	20	0	0
Jap.	0	0	0	0	50	0	0	0	0	0	20	0
Spa.	0	2	0	3	0	45	0	0	0	0	0	20

Table 4 Average identification rate using various HMMs (%)

Number of states	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

5 IDENTIFICATION OF LANGUAGES

5.1 Theory

First, the given model is trained by the each training data. 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 are 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.

5.2 Experimental result

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

original: "For speech recognition, an accurate word recognition system needs ..."

training or test set: "forspeechrecognition", "anaccuratewordrecogn", etc.

Table 3 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 4 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).

6 PREDICTION FOR PHONEME/SYLLABLE/WORD-CATEGORY

6.1 Theory

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

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

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

Finally, calculate the $P(k|y_1, y_2, \ldots, y_t)$ as:

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

6.2 Experimental result

For testing data, a speech-unit (phoneme/syllable/word-category) was predicted by using HMM, and the average correct prediction rate was calculated. The results of the tests are given in Tables 5(a), 6(a), 7(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 5(b), 6(b), 7(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 6(b), 7(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. And it falls short of our expectations, because we think on the basis of model's entropies (Tables 1, 2) that the 7-state HMM has the performance better than bigram and the same performance as trigram.

Table 5 Average correct 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 6 Average correct prediction rate for Japanese syllable (%)

(a) HMM (7-state model)

	number of candidates			
	1	3	5	10
training data	14.9	25.5	41.0	57.1
test data	11.4	24.9	31.5	48.1

(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 7 Average correct prediction rate for English word category (%)

(a) HMM (7-state model)

	number of candidates			
	1	3	5	10
training data	30.4	54.8	66.9	84.8
test data	30.4	52.2	61.9	78.0

(b) Markov model

	N-gram	numer 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

7 CONCLUSIONS

First, natural languages were modeled by HMMs and Markov models. 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 first order Markov model (bigram) and worse than second order Markov model (trigram).

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 Experimnt on English Spelling Rules", ICNN-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", IEEE 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", ICASSP'89, S13.10, pp.731-734, Glasgow, (1989)