Internal Use Only (非公開)

TR-I-0052

A Study of English Word Category Prediction Based on Neural Networks

ニューラルネットによる英文単語列予測モデルの検討

M. Nakamura, K. Shikano
中村雅己、鹿野清宏

1988.11

Abstract

Using traditional statistical approaches, it is difficult to make an N-gram word prediction model to construct an accurate word recognition system because of the increased demand for sample data and parameters to memorize probabilities.To solve this problem, NETgrams, which are neural networks for N-gram word category prediction in text, are proposed. NETgrams are constructed by a trained Bigram network with two hidden layers. Each hidden layer learns the coarse-coded Micro Features (MF1 or MF2) of the input or output word category. NETgrams can easily be expanded from Bigram to N-gram networks without explosively increasing the number of free parameters.

NETgrams are tested by training experiments with a Brown Corpus English Text Database . The training method is the Back-Propagation algorithm. After training, the Trigram word category prediction rates for test data show that the NETgrams are comparable to the statistical model and compress information more than 130 times. Results of analyzing the hidden layer (Micro Features) show that the word categories are classified into some linguistically significant groups. We are now training the 4-gram networks and obtaining good results.

In addition, this paper proposes a new method to speed up the Back-Propagation algorithm, which dynamically controls the training parameters, updating step size and momentum. This new method can automatically determine better parameters and achieve a shorter training time.

ATR Interpreting Telephony Research Laboratories
ATR 自動翻訳電話研究所

Contents

List of Figures

List of Tables

Acknowledgement

The authors would like to express their gratitude to Dr. Akira Kurematsu, president of ATR Interpreting Telephony Research Laboratories, for his encouragement and support, which made this research possible. We are also indebted to the members of the Speech Processing Department at ATR, for their help in the various stages of this research.

1. Introduction

There is a method of predicting word categories using the appearance probabilities of the next word to correct word recognition errors in text[1]. The point of improving prediction ability is to use as much past word information as possible. However, by using such a statistical approach, it is difficult to make an N-gram word prediction model because of the increased demand for sample data and parameters to memorize probabilities.

Neural networks are interesting devices which can learn general characteristics or rules from limited sample data. Neural networks are particularly useful in pattern recognition. In symbol processing, NETtalk[2], which produces phonemes from English text, has been successful. Now we are trying to apply neural networks to word category prediction in English text. It is expected that this task will be very difficult to train because this task is a many-to-many mapping problem with many exception output data and a symbol-to-symbol mapping problem rather than a pattern-to-symbol mapping problem like pattern recognition. We are interested in learning to what degree a neural network can be applied to symbol processing.

This paper describes NETgrams, which are neural networks for N-gram word category prediction in text. In the following section, NETgram requirements are described. In section 3, two NETgrams are proposed. Each model is constructed by a trained basic Bigram network with two hidden layers. Each hidden layer learns the coarse-coded Micro Features (MF1 or MF2) of input or output word category. NETgrams can easily be expanded from Bigram to N-gram networks without exponentially increasing the number of free parameters. In section 4, NETgram training is described. We use Back-Propagation[3] as a training algorithm and Brown Corpus English Text Database[4] as training data. The training results are reported in section 5

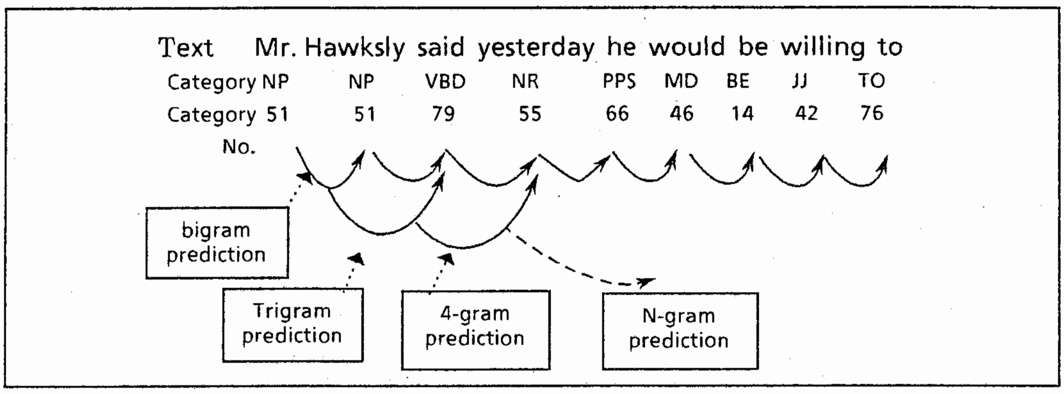

Text Mr. Hawksly said yesterday he would be willing to
Category NP NP VBD NR PPS MD BE JJ TO
Category No. 51 51 79 55 66 46 14 42 76
bigram prediction
Trigram prediction
4-gram prediction
N-gram prediction

Fig. 1-1 Word Category Prediction Using Brown Corpus Text Data

of this paper. The Trigram word category prediction ability of NETgrams are comparable to that of the statistical Trigram model. This means that NETgrams, like Trigram networks, compress information. Results of analyzing the hidden layer (Micro Features) show that the word categories are classified into some linguistically significant groups. In addition, this paper proposes a new method to speed up the Back-Propagation algorithm, which Dynamically Controls the training Parameters (DCP), updating step size and momentum. Considerable time is required to train NETgrams because of the many-to-many mapping problem. In section 6, we describe the DCP method and show that it can automatically determine better parameters and attain a shorter training time.

2. NETgram Requirements

To design neural networks for word category prediction in text, we make the following requirements :

a. Training data is the categories put on the word in Brown Corpus English Text Database[4]. Categories are 88 tags, corresponding to parts of speech in Brown Corpus, and one sentence head blank.

b. The network input layer has several blocks corresponding to the number of input words. For example, a Bigram Network has one input block and a Trigram Network has two. Each block has 89 units and local representation for an input word category. Therefore, in one input block, only one unit corresponding to word category No. is turned ON; The others are turned OFF.

c. Outputs are prediction values for the next possible word categories. Therefore ,an output layer has 89 units.

d. Training algorithm is Back-Propagation[5].

In addition, we consider the following requirements :

e. After learning, hidden layers obtain the coarse-coded Micro Features of the input and output word categories.

f. NETgrams can easily be expanded from Bigram to N-gram networks without exponentially increasing units and connections between them.

3. NETgram (NETwork for N-gram word category prediction)

Two NETgrams are proposed considering the above requirements. Each NETgram is expanded from one basic Bigram network.

3.1. Basic Bigram Network

Basic Bigram network is a 4-layer feed-forward network, as shown in Fig.3-1, which has 2 hidden layers so that each hidden layer obtains coarse-coded MF (Micro Features) of the input or output word category. Because this network is trained the next word category as the output for an input word category, hidden layers are expected to learn some linguistic structure between a word category and the next one in text.

3.2. Expand to N-gram Network

We propose two models to expand to N-gram networks.

3.2.1. Model 1

Model 1 consists of basic Bigram networks put side by side as shown in Fig.3-2. An upper hidden layer (MF2) of a basic Bigram network is fully connected to that of the next basic Bigram network with the link weight set w4. Each basic network's link weight set (w1, w2 or w3) has the same values.

3.2.2. Model 2

Model 1 can learn the Micro Features (MF1) for each input word category independently and has the possibility of expansion to a recurrent network. On the other hand, it must be difficult to train because the number of layers from the first input block to the output layer increase as the gram increases. In order to hold the number of layers from all input blocks to the output layer at four, Model 2 is proposed as shown in Fig.3-3. Model 2 has a structure such that every new input block produced as the gram increases is fully connected to the lower hidden layer of one basic Bigram network with the link weight set at w1'. Initial values of link weight set w1' are all zero. Therefore, the training starts at the output values equal to the trained output values of the basic Bigram network. However, as all input word category information is compressed to one lower hidden layer (MF1), input word category information must be lost to some degree as input blocks increase. Therefore, when expanding from Trigram network to 4-gram network, one lower hidden layer block is added and first and second input blocks are fully connected to one lower hidden layer block, and the second and third input blocks are fully connected to the other lower hidden layer block.

4. How to train NETgram

As input data, word categories in the Brown Corpus text are given in order from the first word in the sentence to the last word. In one input block, only one unit corresponding to the word category No. is turned ON; The others are turned OFF. As output data, only one unit corresponding to the next word category No. is trained by 1 ; The others are trained by 0.

The training algorithm is a new method to speed up the Back-Propagation algorithm, which proposed in section 6.

How to train a NETgram, e.g. Trigram network, is shown in Fig.4-1. First, the basic Bigram network is trained, and next, the Trigram networks are trained with the link weight values trained by the basic Bigram network as initial values. 4-gram networks are trained in the same way.

This task is a many-to-many mapping problem. Thus it is difficult to train because the updating direction of the link weights vector easily fluctuates. In a two-sentence, training experiment of about 50 words, we have confirmed that the output values of the basic Bigram network converge on the next occurrence probability distribution. But for many training data, considerable time is required to train. Therefore in order to increase training speed, we use the next word category occurrence probability distribution calculated for 1,024 sentences (about 24,000 words) as output training data in the basic Bigram network. Of course, in Trigram and 4-gram training, we use the next one-word category as output training data.

5. Training results

5.1. Basic Bigram network

Word category prediction results for training data are shown in Fig.5-1. NETgram (the basic Bigram network) is comparable to the statistical Bigram model.

We calculated the similarity of every two lower hidden layer (MF1) output vectors for 89 word categories and clustered them. Similarity S is calculated by

$$S(c1,c2)=\frac{(M(c1),M(c2))}{\|M(c1)\|\,\|M(c2)\|} \qquad (1)$$

where M(ci) is the lower hidden layer (MF1) output vector of the input word category ci, $(\cdot,\cdot)$ is the inner product function and $\|\cdot\|$ is the norm function. A clustering result is shown in Fig.5-2. Clustering by the threshold of similarity, 0.985, the word categories are classified into linguistically significant groups, which are the HAVE verb group, BE verb group, subjective pronoun group, group whose categories should be before a noun, and others. Therefore NETgrams learn linguistically significant structure naturally.

5.2. Trigram network

Word category prediction results are shown in Fig.5-3. Two NETgrams (Trigram networks) are comparable to the statistical Trigram model for test data in spite of slight inferiority for training data.

Next, we discuss the free parameters of the NETgram. The number of free parameters of the statistic model is the power of 89 in this task, e.g. $89^3=704{,}969$ in the Trigram model, and increases exponentially as the number of grams increases. On the other hand, the number of free parameters of NETgram is the number of link-weights, e.g. 5,193 in the Trigram network Model 1, and increases linearly though the number of grams increase. Therefore, the NETgram in Trigram prediction compresses information more than 130 times as shown in Table 5-1.

Table 5-1 Number of Free Parameters
(ratio; NETgram/statistical model)

	Statistical Model	NETgram Model 1	NETgram Model 2
Bigram	7,921 $=89^2$	3,225 (1/2.5)	3,225 (1/2.5)
Trigram	704,969 $=89^3$	5,193 (1/136)	4,649 (1/152)

6. A new method to speed up the Back-Propagation algorithm

Considerable time is required to train NETgrams (word category prediction networks). It is also very difficult to converge the global minimum because of the many-to-many mapping problem. In this section, a new method to speed up the Back-Propagation algorithm, called DCP (Dynamic Control training Parameters), is proposed.

A basic theory of Back-Propagation [3] is the gradient descent. The rule for changing link weights is given by

$$\Delta w_{ij(k)} = \eta \cdot (-\partial Ep/\partial w_{ij}) + \alpha \cdot \Delta w_{ij(k-1)} \qquad (2)$$

where Ep is the error between the output values and the training desired values and is a function of the link weights. wij is the link weight from the ith unit to the jth unit. The first term is the direction of the gradient descent and the second term is the memory of the last updating step size. This provides a kind of momentum in weight space. Each term has a parameter, η,α, which decides the current real updating step size. The optimal values of these parameters depend on the shape of the weight space, determined by the type of task and the size of the training data, and depend on the degree of training. The DCP method dynamically changes the training parameters (η,α) every N training iterations so that Ep is at a minimum as in the following equation.

$$\begin{aligned} &Ep(w_{ij(k)} + \Delta w_{ij(k)(\eta(k),\alpha(k))}) \qquad (3) \\ &= \underset{l,m}{\mathrm{Min}}\ Ep(w_{ij(k)} + \Delta w_{ij(k)(\eta l,\alpha m)}) \end{aligned}$$

where one of the combinations of N η_l and α_m is chosen.

We performed some for NETgram task experiments. The conditions are as follows :

a. task Basic Bigram network

b. training data 23 words (1 sentence)

In these experiments, we use the next one word category, ON or OFF,

rather than probability distribution for the output training data.

c. parameters

·η	○fixed	(0.1 or 0.4)
	○DCP	(1/2, 1, 2) × $\eta_{(k-1)}$ (choice from 1/2, 1,or 2 times the last η)
·α	○fixed	(0 or 0.9)
	○DCP	(0, 0.9) (choice from 0 or 0.9)

d. threshold value of output error

$Ep < 0.4$

The results are shown in Table 6-1 .As a result of the DCP method, a shorter training time is attained (4.3 times faster in this task) and unsuitable local minima is avoided.

Table 6-1 Training Results of Experiments with Dynamic Control and Fixed Parameters (Basic Bigram Network, 1 Sentence Training Set Size)

	CASE 1 (DCP)	CASE 2 (fixed)	CASE 3 (fixed)	CASE 4 (fixed)	CASE 5 (fixed)	CASE 6
η (step size)	(1/2, 1, 2) ×$\eta_{(k-1)}$	0.1	0.4	0.4	0.1	(1/2, 1, 2) ×$\eta_{(k-1)}$
α (momentum)	(0.0 , 0.9)	0.9	0.9	0	0	0
Iteration	**35**	153	more than 200	178	more than 200	more than 200

7. Conclusion

In this paper we have presented two NETgrams, neural networks for N-gram word category prediction in the text. Each model is constructed by a trained basic Bigram network with two hidden layers. NETgrams can easily be expanded from Bigram to N-gram networks without exponentially increasing the number of free parameters.

The training results showed that the Trigram word category prediction ability of NETgrams was comparable to that of the statistical Trigram model and compressed information more than 130 times.

The results of analyzing the hidden layer (Micro Features) after training showed that the word categories were classified into some linguistically significant groups, that is to say a NETgram learns a linguistically significant structure naturally.

In addition, this paper proposed a new method to speed up the Back-Propagation algorithm, which Dynamically Controls the training Parameters (DCP), updating step size and momentum. Considerable time is required to train NETgrams because of the many-to-many mapping problem. The experiment results showed an ability to automatically determine better parameters and achieve a shorter training time.

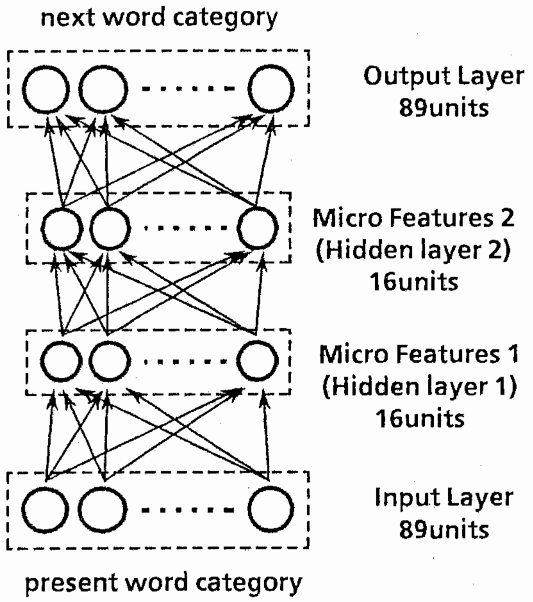


Fig.3-1 Basic Bigram Network for Word Category Prediction

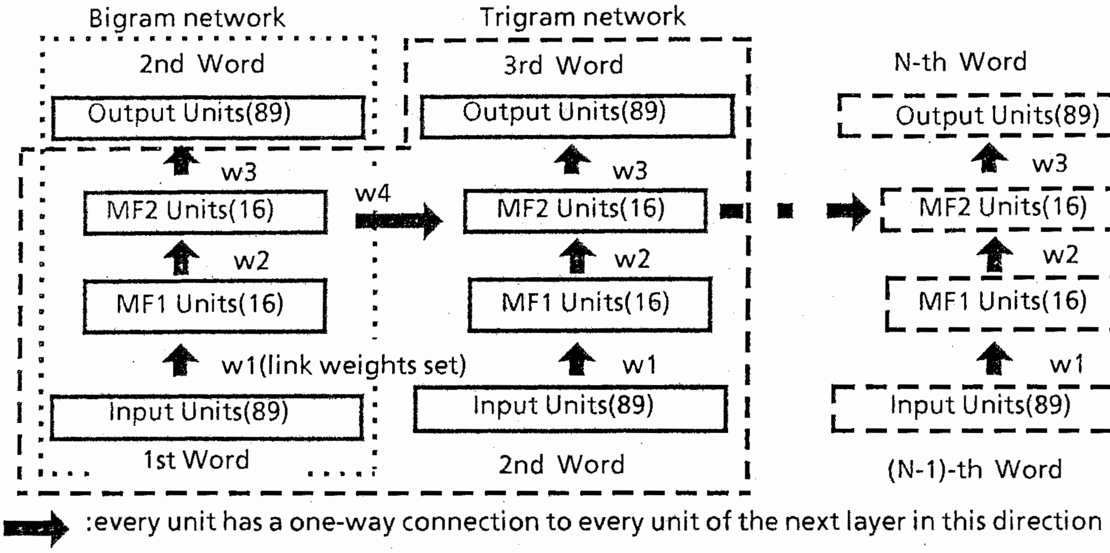


Fig.3-2 NETgram Model 1 for Word Category Prediction

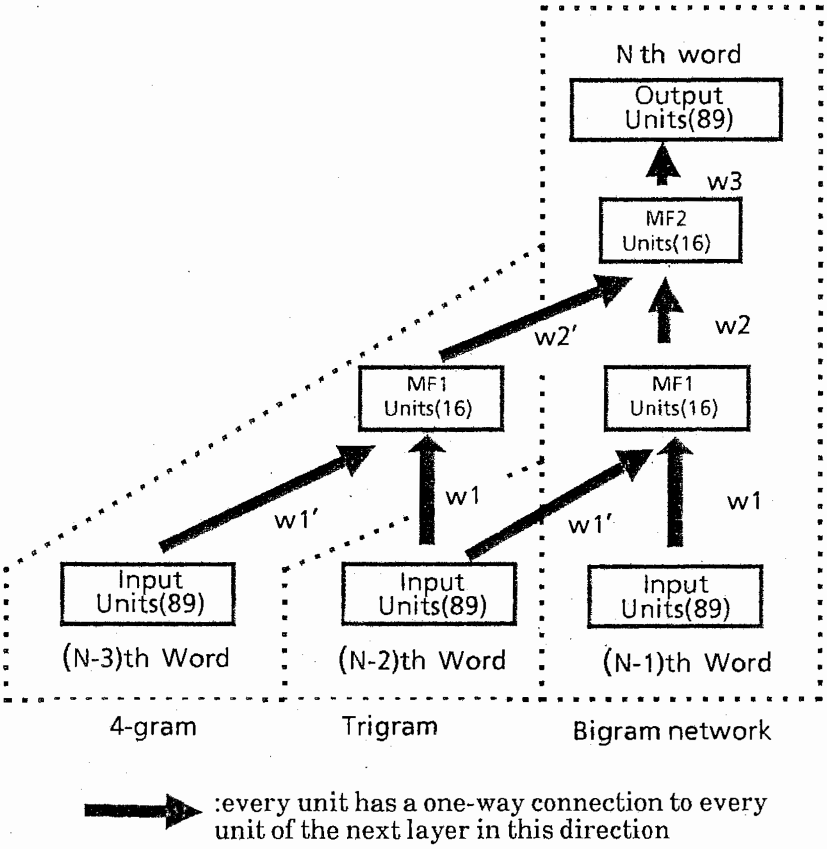
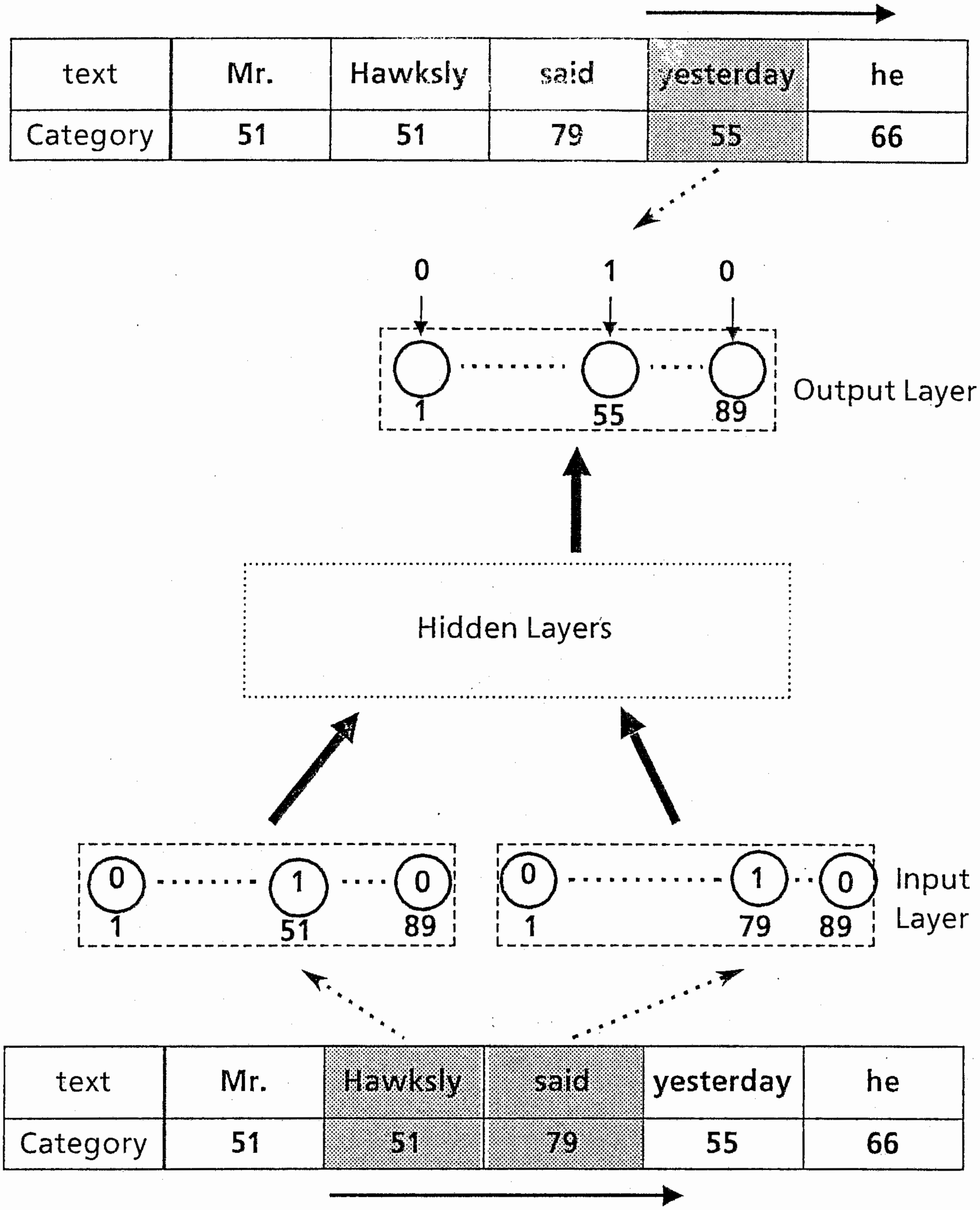


Fig.3-3 NETgram Model 2 for Word Category Prediction



text	Mr.	Hawksly	said	yesterday	he
Category	51	51	79	55	66

text	Mr.	Hawksly	said	yesterday	he
Category	51	51	79	55	66

Fig.4-1 How to Train NETgram(Trigram Model)

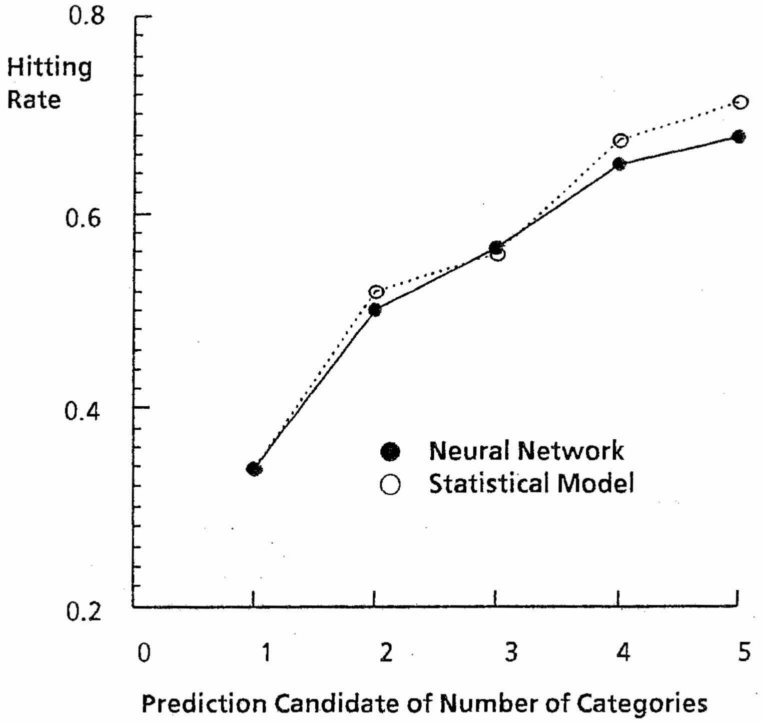
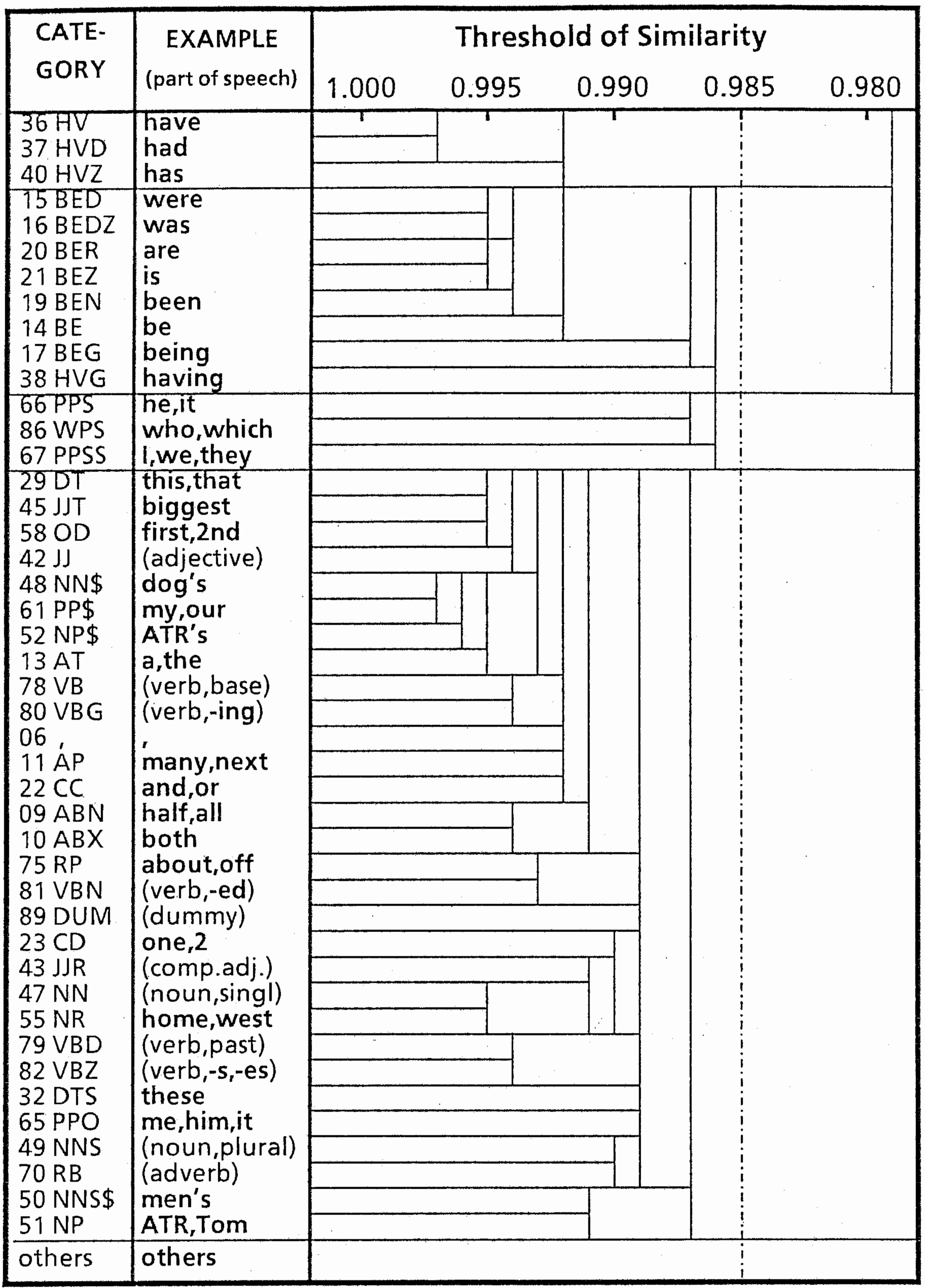
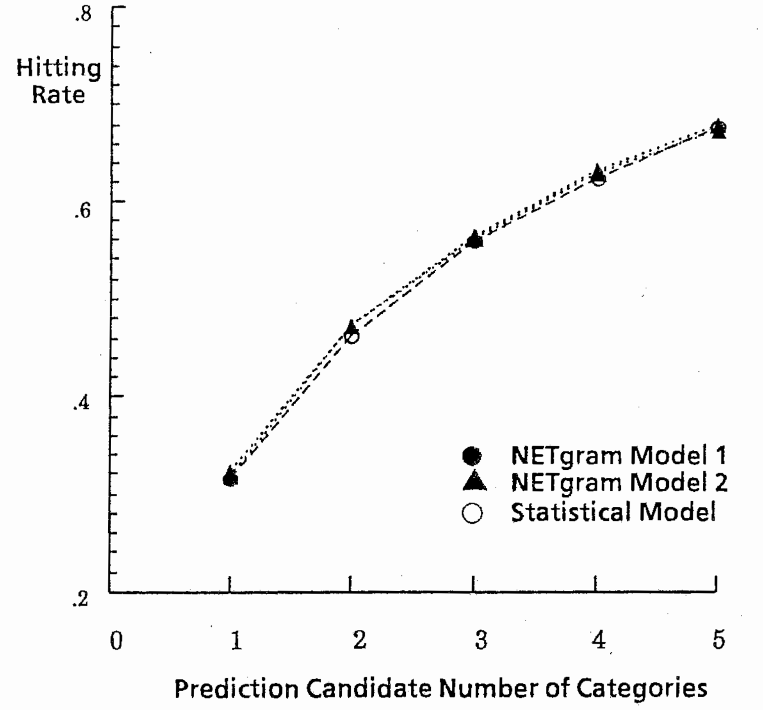


Fig.5-1 Basic Bigram Network Prediction Hitting Rate (Training Data)

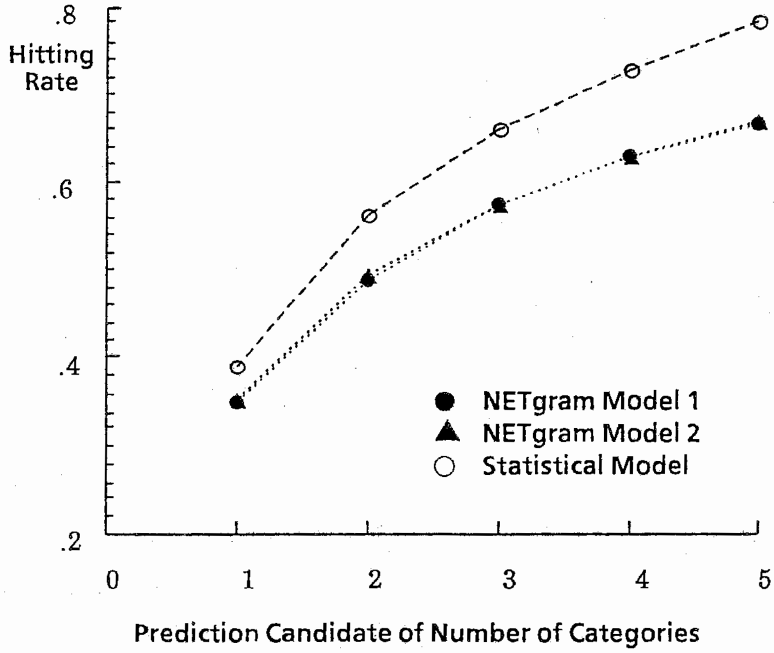


CATE-GORY	EXAMPLE (part of speech)
36 HV	have
37 HVD	had
40 HVZ	has
15 BED	were
16 BEDZ	was
20 BER	are
21 BEZ	is
19 BEN	been
14 BE	be
17 BEG	being
38 HVG	having
66 PPS	he,it
86 WPS	who,which
67 PPSS	I,we,they
29 DT	this,that
45 JJT	biggest
58 OD	first,2nd
42 JJ	(adjective)
48 NN$	dog's
61 PP$	my,our
52 NP$	ATR's
13 AT	a,the
78 VB	(verb,base)
80 VBG	(verb,-ing)
06 ,	,
11 AP	many,next
22 CC	and,or
09 ABN	half,all
10 ABX	both
75 RP	about,off
81 VBN	(verb,-ed)
89 DUM	(dummy)
23 CD	one,2
43 JJR	(comp.adj.)
47 NN	(noun,singl)
55 NR	home,west
79 VBD	(verb,past)
82 VBZ	(verb,-s,-es)
32 DTS	these
65 PPO	me,him,it
49 NNS	(noun,plural)
70 RB	(adverb)
50 NNS$	men's
51 NP	ATR,Tom
others	others

Fig. 5-2 A Clustering Result of MF1 Output Values of Basic Bigram Network



(a) Test Data of 512 Sentences



(b) Training Data of 512 Sentences

Fig.5-3 NETgram (Trigram) Prediction Hitting Rates

References

[1] K.Shikano: Improvement of Word Recognition Results by Trigram Models, ICASSP86, 29.2 (1987.4)

[2] T.J.Sejnowski.et.al.:NETtalk, A Parallel Network that Learns to Read Aloud, Tech. Report, The Johns Hopkins University EESS (1986)

[3] D.E.Rumelhart et al.: Parallel Distributed Processing, M.I.T. Press (1986)

[4] Brown University: Brown Corpus, Tech. Report, Brown University (1967)