NEURAL NETWORK APPROACH TO WORD CATEGORY PREDICTION FOR ENGLISH TEXTS

Masami NAKAMURA, Katsuteru MARUYAMA[†], Takeshi KAWABATA[††], Kiyohiro SHIKANO[†††]

ATR Interpreting Telephony Research Laboratories
Seika-chou, Souraku-gun, Kyoto 619-02, JAPAN
e-mail masami@atr-la.atr.co.jp

Abstract

Word category prediction is used to implement an accurate word recognition system. Traditional statistical approaches require considerable training data to estimate the probabilities of word sequences, and many parameters to memorize probabilities. To solve this problem, NETgram, which is the neural network for word category prediction, is proposed. Training results show that the performance of the NETgram is comparable to that of the statistical model although the NETgram requires fewer parameters than the statistical model. Also the NETgram performs effectively for unknown data, i.e., the NETgram interpolates sparse training data. Results of analyzing the hidden layer show that the word categories are classified into linguistically significant groups. The results of applying the NETgram to HMM English word recognition show that the NETgram improves the word recognition rate from 81.0% to 86.9%.

1. Introduction

For the realization of an interpreting telephony system, an accurate word recognition system is necessary. Because it is difficult to recognize English words using only their acoustical characteristics, an accurate word recognition system needs certain linguistic information. Errors in word recognition results for sentences uttered in isolation include the following types of errors recoverable using linguistic information.

(a) Local syntax errors.

(b) Global syntax errors.

(c) Semantics and context errors.

Many errors arise with one-syllable words such as (I, by) and (the, be). More than half of these errors can be recovered by use of local syntax rules. The Trigram language model is an extremely rough approximation of a language, but it is a practical and useful model from the viewpoint of entropy. At the very least, the trigram model is useful as a preprocessor for a linguistic processor which will be able to deal with syntax, semantics and context.

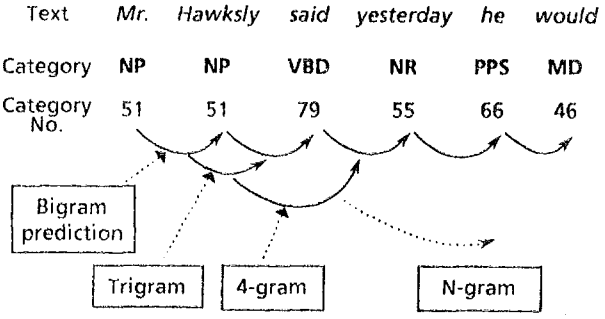

Fig.1 Word Category Prediction
Using Brown Corpus Text Data

The trigram model using the appearance probabilities of the following word was efficiently applied to improve word recognition results [1][2]. However, the traditional statistical approach requires considerable training samples to estimate the probabilities of word sequence and considerable memory capacity to process these probabilities. Additionally, it is difficult to predict unseen data which never appeared in the training data.

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 [3], which produces phonemes from English text, has been used successfully. Now a neural network is being applied to word category prediction [4].

This paper describes the NETgram, which is a neural network for word category prediction in text. The NETgram is constructed by a trained Bigram network with two hidden layers, so that each hidden layer can learn the coarse-coded features of the input or output word category. Also, the NETgram can easily be expanded from Bigram to N-gram network without exponentially increasing the number of parameters. The NETgram is tested by training experiments with the Brown Corpus English Text Database

† Research and Development Department, NITSUKO Corporation
†† NTT Basic Research Laboratories
††† NTT Human Interface Laboratories

[5]. The NETgram is applied to HMM English word recognition resulting in an improvement of its recognition performance.

2. Word Category Prediction Neural Net (NETgram)

The basic Bigram network in the NETgram is a 4-layer feed-forward network, as shown in Fig.2, which has 2 hidden layers. Because this network is trained for the next word category as the output for an input word category, hidden layers are expected to learn some linguistic structure from the relationship between one word category and the next in the text. The Trigram network in the NETgram has a structure such that, as the number of grams increases, every new input block produced is fully connected to the lower hidden layer of one basic Bigram network. The link weight is set at w1' as shown in Fig.3. When expanding from Trigram network to 4-gram network, one lower hidden layer block is added and the 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.

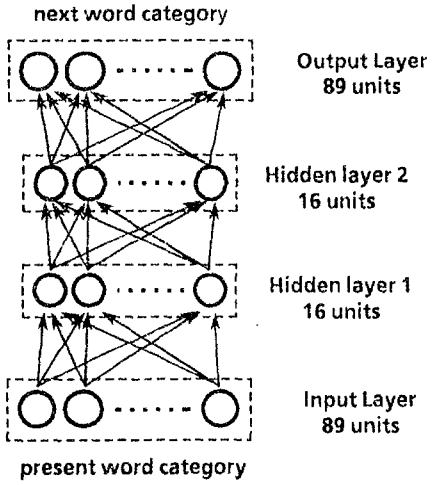

Fig.2 NETgram (Basic Bigram Network)

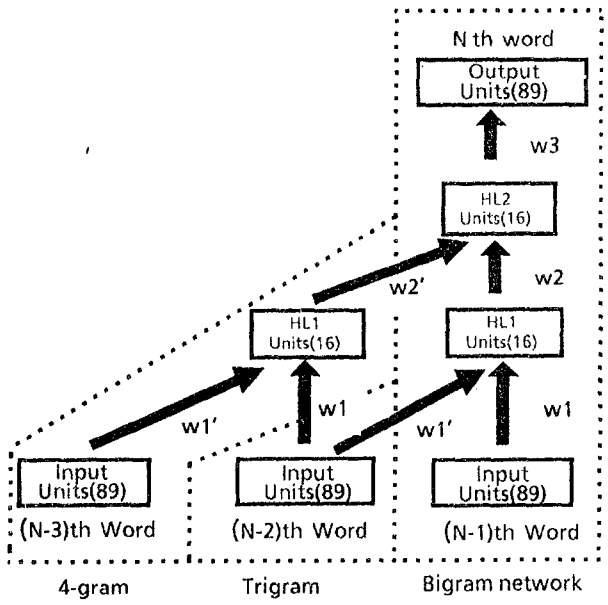

Fig.3 NETgram (Trigram, 4-gram Network)

3. How to Train NETgram

How to train a NETgram, e.g. a Trigram network, is shown in Fig.4. As input data, word categories in the Brown Corpus text[5] are given, in order, from the first word in the sentence to the last. In one input block, only one unit corresponding to the word category number is turned ON (1); The others are turned OFF (0). As output data, only one unit corresponding to the next word category number is trained by ON (1). The others are trained by OFF (0). The training algorithm is the Back-Propagation algorithm[6], which uses the gradient descent to change link-weights in order to reduce the difference between the network output vectors and the desired output vectors. First, the basic Bigram network is trained. Next, the Trigram networks are trained with the link weight values trained by the basic Bigram network as initial values.

This task is a many-to-many mapping problem. Thus, it is difficult to train because the updating direction of the link weight vector easily fluctuates. In a two-sentence

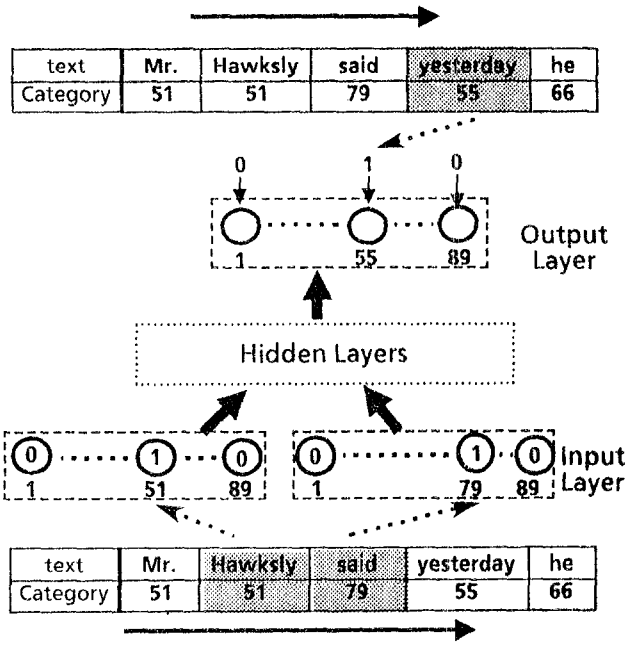

Fig.4 How to Train NETgram (Trigram Model)

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. However, for many training data, considerable time is required for training. 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.

4. Training Results

4.1. Basic Bigram Network

Word category prediction results show that NETgram (the basic Bigram network) is comparable to the statistical Bigram model.

Next, we consider whether the hidden layer has obtained some linguistic structure. We calculated the similarity of every two lower hidden layer (HL1) output vectors for 89 word categories and clustered them. Similarity S is calculated by

$$S(Ci,Cj) = \frac{(\mathbf{M}(Ci),\mathbf{M}(Cj))}{\| \mathbf{M}(Ci) \| \ \| \mathbf{M}(Cj) \|} \qquad (4.1)$$

where $\mathbf{M}(Ci)$ is the lower hidden layer (HL1) output vector of the input word category Ci. $(\mathbf{M}(Ci),\mathbf{M}(Cj))$ is the inner product of $\mathbf{M}(Ci)$ and $\mathbf{M}(Cj)$. $\| \mathbf{M}(Ci) \|$ is the norm of $\mathbf{M}(Ci)$. The clustering result is shown in Fig.5. 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 the NETgram can learn linguistic structure naturally.

4.2. Trigram Network

Word category prediction results are shown in Fig.6. The NETgram (Trigram network) is comparable to the statistical Trigram model for test data.

Furthermore, the NETgram performs effectively for unseen data which never appeared in the training data, although the statistical Trigram can not predict the next word category for unseen data. That is to say, NETgrams interpolate sparse training data in the same way deleted interpolation [7] does.

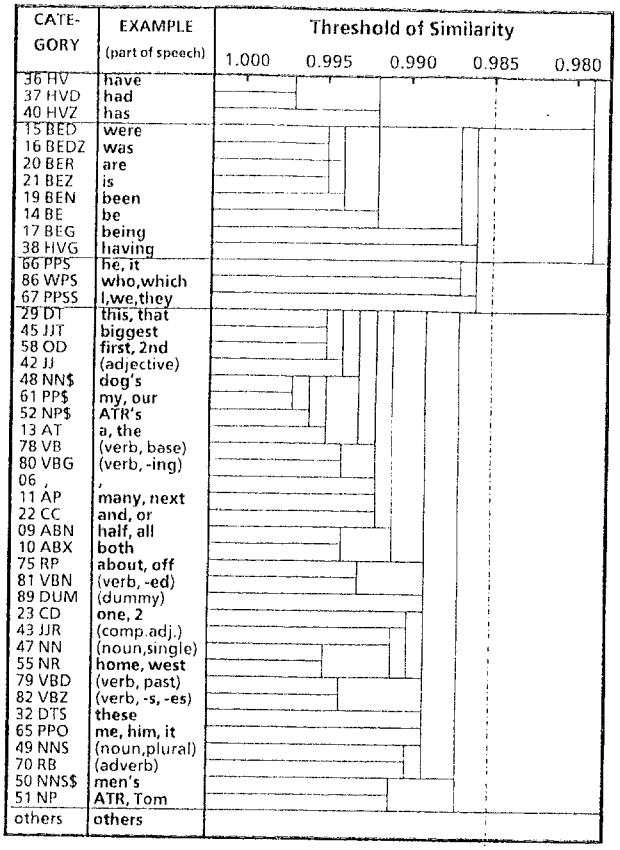

Fig.5 Clustering Result of HL1 Output Vectors of NETgram (Bigram)

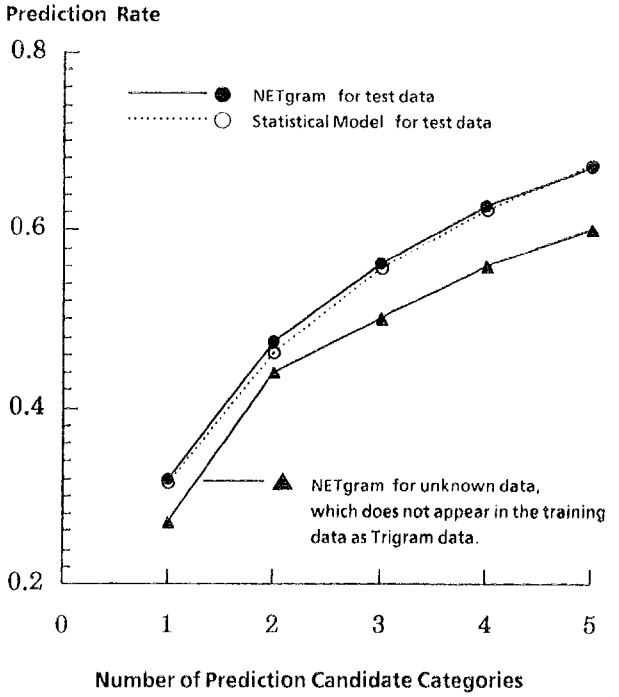

Fig.6 NETgram (Trigram) Prediction Rates

4.3. Differences between the statistical model and the NETgram

We discuss differences between two approaches, the conventional statistical model and the NETgram. The conventional statistical model is based on the table-lookup. In the case of the Trigram model, next appearance probabilities are computed from the histogram, counting the next word category for the two word categories in the training sentences. The probabilities are put in an 89*89*89 size table. Thus, the 89 appearance probabilities of the next word category are obtained from the 89*89*89 size table using the argument of 89*89 symbol permutation.

B^{89*89} ---> R^{89} B ; binary space
R ; real space

In order to get 89 prediction values for the next word category, the trained NETgram procedure is as follows :

First, encode from an 89*89 symbol permutation to a 16-dimensional analogue code. (from the input layer to the hidden layer 1)

Second, transform the 16-dimensional analogue code to a 16-dimensional analogue code of the next word's 89 prediction values. (from hidden layer 1 to hidden layer 2)

Finally, decode the 16-dimensional analogue code to 89 prediction values of the next word's 89 prediction values. (from hidden layer 2 to output layer)

B^{89*89} ---> R^{16} ---> R^{16} ---> R^{89}

The values of each space are output values of the NETgram units of each layer. These mappings are uniquely determined by link-weight values of the NETgram. That is to say, each layer unit value is computed by summing lower-connected unit values multiplied by link-weights and passing through the nonlinear function (sigmoid function). These two approaches need the following memory area (number of parameters).

·Statistical model $89\times89\times89 = 704{,}969$
(max number of table elements)

·NETgram $(89+89)\times16+16\times16+16\times89+121 = 5{,}193$
(number of link-weights)
(121 ; offset parameters)

Thus, the parameters of the statistical model are $89\times89\times89$ probabilities. In practice, there are many 0 values in $89\times89\times89$ probabilities and the size of the table can be reduced using a particular technique. However, this depends on the kind of task and the number of training data. On the other hand, the NETgram can produce $89\times89\times89$ prediction values using link-weight values memorized as parameters.

Next, concerning the data representation, the statistical model does not use input data structures because it is based on the table-lookup which get probabilities directly by symbol series input. On the other hand, the NETgram extracts a feature related to the distance between word categories from symbol series input into 16-dimensional analogue code. 16-dimensional analogue codes are described in 4.1 as the feature of the NETgram hidden layer in the Bigram model. Thus, the NETgram interpolates sparse training data in the process of bigram and trigram training. From the viewpoint of data coding, The NETgram compresses data from 89-dimensional binary space into 16-dimensional real space.

4.4. 4-gram Network

4-gram prediction rates of the NETgram trained by 2,048 are not much higher than the trigram prediction rates of that trained by 1,024 sentences. The statistical model experiment results show that more than 6,000 sentences are necessary as training data in order for the 4-gram prediction rates to equal the trigram prediction rates of the NETgram trained by 1,024 sentences. Futhermore, the trigram prediction rates of the statistical model increase as the training sentences increase, up to a max of 16,000 training sentences. The NETgram compensates for the sparse 4-gram data through the interpolation effect. However, it is clear that the 4-gram prediction NETgram needs far more than 16,000 training sentences in order to better the performance of the trigram prediction. Training for so many sentences was not possible because of the limited database and considerable computing required.

5. Applying the NETgram to Speech Recognition

The algorithm for applying the NETgram to speech recognition is shown in Fig.7. HMM refers to the Hidden Markov Model which is a technique for speech recognition[1][8][9].

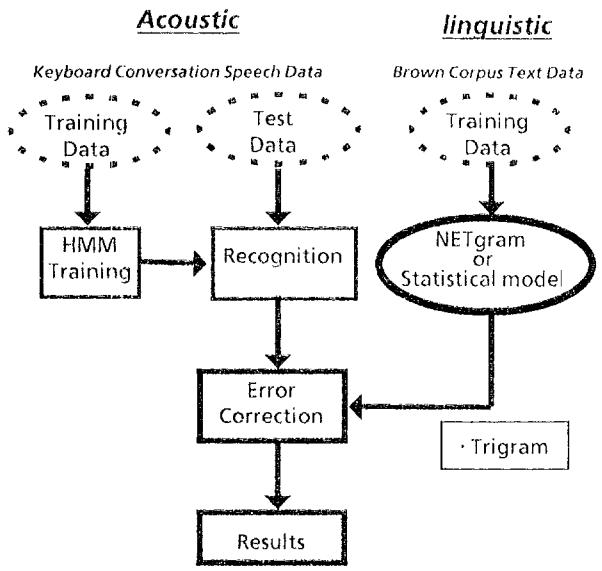

Fig. 7 Improvement of HMM English Word Recognition Using the NETgram

5.1. Formulation

Let w_i show a word just after w_{i-1} and just before w_{i+1}. Let C_i show one of the word categories to which the word w_i belongs. The same word belonging to a different category is regarded as a different word. The trigram probability of w_i is calculated using the following approximations.

$$\begin{aligned} & P(w_i / w_{i-2}\, w_{i-1}) \\ & \quad \simeq P(w_i / C_{i-2}\, C_{i-1}) \\ & \quad = P(C_i / C_{i-2}\, C_{i-1}) \\ & \qquad \times \{ P(w_i / C_{i-2}\, C_{i-1}) / P(C_i / C_{i-2}\, C_{i-1}) \} \\ & \quad \simeq P(C_i / C_{i-2}\, C_{i-1}) \{ P(w_i) / P(C_i) \} \end{aligned} \tag{5.1}$$

Word trigram probabilities are approximated using category trigram probabilities as follows :

$$P(w_i / w_{i-2}\, w_{i-1}) \simeq P(w_i / C_{i-2}\, C_{i-1}) \tag{5.2}$$

The probability of w_i is denoted by the preceding two-word sequence, w_{i-2}, w_{i-1}, and is approximated by their preceding two-category sequence.

$$P(w_i / C_{i-2}\, C_{i-1}) / P(C_i / C_{i-2}\, C_{i-1}) \simeq P(w_i) / P(C_i) \tag{5.3}$$

The probability ratio of wi and C_i given by C_{i-2} C_{i-1} is nearly equal to the total probability ratio of wi and C_i.

To calculate the above probability, the trigram probability of word category, $P(C_i \mid C_{i-2}\ C_{i-1})$, and word occurrence probability, $P(w_i)/P(C_i)$, are required. The word probability, $P(w_i) / P(C_i)$, is prestored in the dictionary of word w_i for each word category.

To avoid the multiplication of probabilities, the log-likelihood, ST_i, is defined as :

$$ST_i = \log P(C_i / C_{i-2}\, C_{i-1}) + \log(P(w_i) / P(C_i)) \tag{5.4}$$

The first term is retrieved from the trigram of word categories and the second term is retrieved from the word dictionary.

The maximum likelihood of a word, SW, is given by the sum of word likelihood values of a n-word sequence. The j-th word candidate in the i-th word of a sentence is denoted by $w_{i,j}$. The likelihood of $w_{i,j}$, $SW_{i,j}$, is defined as the sum of two types of likelihood which are the log-likelihood of the HMM output probability, $SH_{i,j}$, and the trigram likelihood, $ST_{i,j}$. Thus, the likelihood of $w_{i,j}$ is described as follows :

$$SW_{i,j} = (1-\omega) \cdot SH_{i,j} + \omega \cdot ST_{i,j} \tag{5.5}$$

where ω is the weighting parameter to adjust the scaling of two kinds of likelihood.

The maximum sentence likelihood values, G, are denoted by the following equations :

$$G_{0,j} = SW_{0,j} \qquad (i = 0) \tag{5.6}$$

$$G_{i,j} = \max_{k}(SW_{i,j} + G_{i-1,k}) \qquad (i \neq 0) \tag{5.7}$$

When the length of a sentence is N, the maximum value of $G_{N-1,j}$ is regarded as the maximum likelihood of the word sequence. The back-tracing of $w_{i,j}$ gives the optimal word sequence.

In this paper, the best-ten candidates in the HMM word recognition results are used. As the same word belonging to a different category is regarded as a different word, there are ten or more word candidates.

5.2. English Word Recognition Results

The experiment task is to translate keyboard conversations which include 377 English sentences (2,834 words) uttered word by word by one male native speaker. The sentences are composed of 542 different words. HMM phone models are trained using 190 sentences (1,487 words) without phone labels.

The trigram models, the NETgram and the statistical model, are trained using using 512 and 1,024 sentences of the Brown Corpus Text Database. One sentence is about 24 words long.

English word recognition results for 187 sentences (1,347 words) of keyboard conversations using HMM and the trigram models are shown in Table 1. The recognition rate in the experiment using only HMM is 81.0%. Using the

NETgram, the recognition rates have been improved about 5 or 6 %. The number of recognition errors decreases using NETgram.

Table 1 HMM English Word Recognition Rates using NETgram or Statistical model (%)

Model	Training Sentences	NETgram	Statistical Model
Trigram	512	86.3	85.5
	1,024	86.9	85.4

Comparing the NETgram and the statistical trigram model, the performance of the NETgram is higher than that of the statistical trigram in the case of training data consisting of 512 and 1,024 sentences. Furthermore, the statistical trigram model cannot learn word sequences which do not appear as a trigram in the training data. Thus, the prediction value of that word sequence is zero. The NETgram does not make such fatal mistakes.

Additional results for 4,096 and 30,000 sentences show that recognition rates are 86.9% and 87.2% using the NETgram, and 86.6% and 87.7% using the statistical model. It is confirmed that the NETgram performs better than the statistical trigram model when data is insufficient to estimate the correct probabilities Therefore, even if training data were insufficient to estimate accurate trigram probabilities, the NETgram performs effectively. That is to say, the NETgram interpolates sparse trigram training data using bigram training memory.

6. Conclusion

In this paper we have presented the NETgram, a neural network for N-gram word category prediction in text. The NETgram can easily be expanded from Bigram to N-gram network without exponentially increasing the number of parameters.

The training results showed that the Trigram word category prediction ability of the NETgram was comparable to that of the statistical Trigram model although the NETgram requires fewer parameters than the statistical model. We also confirmed that NETgrams performed effectively for unknown data which never appeared in the training data, that is to say, NETgrams interpolate sparse training data naturally.

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

Next, the NETgram was applied to HMM English word recognition, and it was shown that the NETgram can effectively correct word recognition errors in text. The word recognition rate using HMM is 81.0%. The NETgram trained by 1,024 sentences improves the word recognition rate to 86.9%. The NETgram performs better than the statistical trigram model when data is insufficient to estimate the correct probabilities of a word sequence.

Acknowledgement

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

References

[1] F.Jelinek, "Continuous Speech Recognition by Statistical Methods", Proceedings of the IEEE, Vol.64, No.4 (1976.4)

[2] K.Shikano, "Improvement of Word Recognition Result by Trigram Model", ICASSP 87, 29.2 (1987.4)

[3] T.J.Sejnowski, C.R.Rosenberg, "NETtalk, A Parallel Network that Learns to Read Aloud", Tech. Report, The Johns Hopkins University EESS-86-01 (1986)

[4] M.Nakamura, K.Shikano, "A Study of English Word Category Prediction Based on Neural Networks", ICASSP 89, S13.10(1989.5)

[5] Brown University, "Brown Corpus", Tech. Report, Brown University (1967)

[6] D.E.Rumelhart, G.E.Hinton, R.J.Williams, "Parallel Distributed Processing", M.I.T. Press (1986)

[7] F.Jelinek, R.Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data", Pattern Recognition in Practice, ed. E.S. Gelsema and L. N. Kanal, North Holland (1980)

[8] S.E.Levinson, L.R.Rabiner, M.M.Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", Bell Syst. Tech. J. 62(4), pp1035-1074 (1983)

[9] T.Hanazawa, G.Kawabata, K.Shikano, "Study of Separate Vector Quantization for HMM Phoneme Recognition", ASJ 2-P-8(1988.10)