AN AUTOMATIC ACQUISITION METHOD OF STATISTIC FINITE-STATE AUTOMATON FOR SENTENCES

Motoyuki SUZUKI, Shozo MAKINO

Computer Center
TOHOKU University

Hirotomo ASO

Graduate School of Engineering
TOHOKU University

ABSTRACT

Statistic language models obtained from a large number of training samples play an important role in speech recognition. In order to obtain higher recognition performance, we should introduce long distance correlations between words. However, traditional statistic language models such as word n-grams and ergodic HMMs are insufficient for expressing long distance correlations between words. In this paper, we propose an acquisition method for a language model based on HMnet taking into consideration long distance correlations and word location.

1. INTRODUCTION

It is essential to utilize a language model in speech recognition systems to overcome phoneme recognition errors and to increase recognition accuracy. The language models are classified into two types: one is manually constructed by an expert based on his grammatical knowledge. The other is automatically constructed using a large amount of training sentences. The former type of language model usually gives a higher recognition performance, however needs very long time to construct and is difficult to apply to different tasks. On the other hand, the latter type can be automatically constructed from a large number of sentences. Recently, statistic language models belonging to the latter type such as word n-grams [1–3] and ergodic HMMs [4] models were frequently used in speech recognition. They can represent correlations between two words located in the neighborhood. However, they are insufficient for expressing long distance correlations between words and cannot consider variation of occurrence probability dependent on word-location. In this paper, we propose a new automatic construction algorithm for a statistic finite-state automaton from a large number of training sentences. The algorithm is based on Hidden Markov Network(HMnet) [5] taking into consideration long distance correlations and word location.

Figure 1 shows an example of HMnet. Each state corresponds to a word location and has a discrete distribution each element of which expresses output probability of a dictionary word-item at the location. The number of element is that of word-items in the dictionary used in the task. As can be seen from this figure, HMnet can express long distance correlations between words and also can do information concerning to word-location.

The next problem is how to construct and how to define structure of HMnet. In order to solve these problems, we propose a new construction algorithm based on the Successive State Splitting algorithm [5]. The original algorithm is for constructing an acoustic models. It starts from a single state, and splits the state with the maximum distribution to two states. All possibilities are examined based on the pre-defined factors such as phoneme class, phoneme contexts, and so on. However, when applying this method to construct a language model, it is very difficult to define factors for splitting. In this paper, we propose a new construction algorithm requiring no factors for splitting. The new method splits a state to two states based on distance between samples. Furthermore, it is a problem how to define the distance between samples. We introduce a word distance by taking into account occurrence frequencies of the previous and the following words.

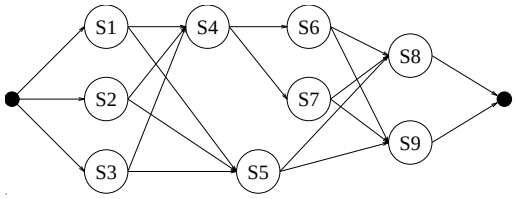

Figure 1: Example of HMnet

2. THE DISCRETE-TYPE HMNET CONSTRUCTION ALGORITHM

2.1. Outline of HMnet and new construction algorithm

An example of HMnet is shown in Fig. 1. Each path of HMnet from the initial state to the terminal state corresponds to a single HMM. Each state of HMnet has output probability distribution and transition probability, and these are estimated from a large number of training samples.

The structure of HMnet is constructed by the new algorithm as follows. The algorithm starts from a single state. Next it defines the state with the maximum distribution size in the HMnet and then splits the state into two states. The distribution size is assumed to be bigger if the number of the context at the state is larger. That is, the bigger distribution size means that the state represents the word set with different contexts. On the other hand, the smaller distribution size means that the state represents the word set with similar contexts.

2.2. The definition of distance between words

In order to calculate the distribution size of a state, we should define the distance between two words in the following [6].

From the training samples, $P_l(k|w_i)$(occurrence probability of a preceding word k) and $P_r(k|w_i)$(occurrence probability of a successive word k) are estimated for each word w_i. The distance between word w_i and w_j is defined as

$$\begin{aligned} d(w_i, w_j) &= D(P_l(k|w_i), P_l(k|w_j)) + \\ & \quad D(P_r(k|w_i), P_r(k|w_j)), \end{aligned} \tag{1}$$

where $D(p, q)$ is Kullback Divergence computed by:

$$D(p,q) = \sum_i p(i) \log \frac{p(i)}{q(i)} + q(i) \log \frac{q(i)}{p(i)}. \tag{2}$$

This value reflects the difference of the kind of contexts.

2.3. New construction algorithm

Step 1 Training of an initial model
An HMnet consisting of one state with a discrete distribution is trained using all training samples.

Step 2 Calculation of the distribution size
The size of distribution V_i of i-th state $S(i)$ is calculated by

$$V_i = n_i \times \min_j \sum_k^N d(w_j, w_k) P_i(w_k), \tag{3}$$

where N denotes the number of words; $P_i(w_k)$ denotes the output probability of word w_k at state i; and n_i denotes the number of training samples for state i.

Step 3 State splitting
State $S(m)$ with a maximum distribution size is split into two new states $S'(m)$ and $S(M)$. The following two possibilities of the state splitting are carried out.

a) Split on the temporal domain
The new HMnet concatenating two new states in series is constructed, it is retrained using all training samples.

b) Split on the contextual domain
The new HMnet concatenating two new states in parallel is constructed. Each of the training samples accepted by state $S(m)$ is assigned to new states using the following algorithm:

1. The word-sequences assigned to state $S(m)$ are picked up using the Viterbi algorithm.
2. The picked-up sequences are split into two clusters using the Furthest-Neighbor clustering algorithm. The distance between the picked-up sequences is calculated using dynamic programming.
3. The clusters are assigned to new states.

The new HMnet is retrained using all training samples.

Step 4 Choice of HMnet
The new HMnet with a higher likelihood for all training samples is selected.
Steps 2 to 4 are repeated until the number of states reaches the pre-defined number.

3. NL-HMNET CONSTRUCTION ALGORITHM

The discrete-type HMnet has self-loop transitions. However, we cannot find such a structure in natural language. So, we propose a new construction algorithm for HMnet without a self-loop (No Loop HMnet: NL-HMnet).

The NL-HMnet construction algorithm is similar to the discrete-type HMnet construction algorithm. The following steps, step 1 and step 3, are modified for the new algorithm.

Step 1' Training of an initial model
Training of an initial model consists of the following two steps.

1-1 Construction of an initial model
An HMnet consisting of n states with a discrete distribution is constructed using all training samples. n is set to the maximum length of training samples, and each state in the initial model has transitions to all of the following states (Fig. 2).

1-2 Assignment of training samples to the path
Each training sample is assigned to the path with a maximum likelihood using the Viterbi algorithm.

Step 3' State splitting
State splitting is only carried out on the contextual domain. Words are split into two clusters using the minimum distortion clustering algorithm because the length of the picked-up sequences is one. The center words of the two clusters are defined as w_{c1} and w_{c2}. They are determined by

$$(w_{c1}, w_{c2}) = \underset{i,j}{\operatorname{argmin}} \sum_{k}^{N} \min(d(w_i, w_k), d(w_j, w_k)) \times P_m(k), \quad (4)$$

where $P_m(k)$ denotes the output probability of word k at state m.

4. EXPERIMENTS

4.1. Application to artificial language

A source language model expressed by a finite state automaton represents airport control commands, where the number of states is 64 and the vocabulary size is 59 words. Transition probabilities are assumed to be equal. Training and test samples are randomly generated from the model. The number of training samples is 5,000, the number of test samples is 20,000.

Figure 3 shows the test set perplexities. The mark '$\diamond$' denotes that the n-gram has the same number of parameters as an HMnet of that point. Both HMnets show lower perplexities than word n-grams at an optimum number of states. However, all perplexities with a large number of states are not reliable because the number of training samples is much smaller than the freedom of the models. On the other hand, ergodic HMMs show lower perplexities than discrete HMnets, and about equal as NL-HMnet at a large number of states. However, the training of the large ergodic HMM needs much computing power.

From the comparison with perplexities of both HMnets, HMnet using as a language model doesn't need self-loop transitions.

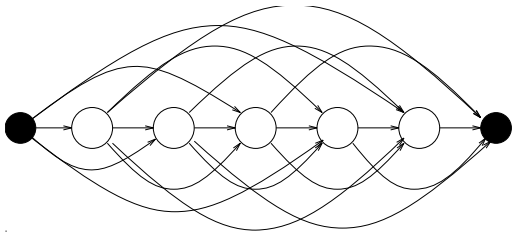

Figure 2: Example of an initial model for NL-HMnet

4.2. Application to natural language

Autopsy document was used as training and test samples. 72 documents were used as training samples, and other 19 documents were used as test samples. Each document consists of 28 sections, and the average number of training samples is 382 sentences par one section. All documents were analyzed into morphology using morphological analysis system named "ChaSen" [7].

We constructed NL-HMnet and trigram model with deleted interpolation method [8] for each section, and test set perplexity calculated using NL-HMnet was compared with that using trigram. In the test documents, 4 sections which had few sentence accepted by the NL-HMnet (coverage was less than 30 %) were ignored. To avoid the influence from flooring value when a sentence is not accepted by the NL-HMnet, test sentences accepted by the NL-HMnet is only used as test samples. The total coverage of all section was 69 %.

Figure 4 shows perplexities for each section, and Fig. 5 shows the average number of words in one sentence. From the Fig. 4, NL-HMnet totally showed lower perplexities than trigram. Especially, we can find

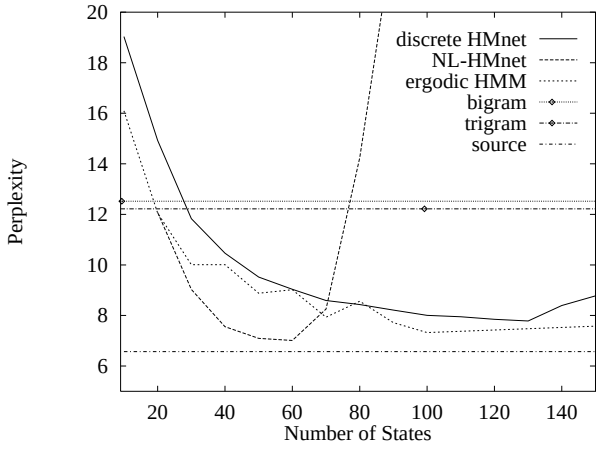

Figure 3: Perplexity for airport control commands task

big differences was shown between perplexities in the section 2, 4, 6. Figure 5 shows that average numbers of words were relatively large in these sections. From these results, we can confirm that NL-HMnet can express long distance correlations between words, but the trigram cannot. On the other hand, the trigram showed lower perplexities than NL-HMnet in the section 12, 18–21 and 24, because average numbers of words were small in these sections.

5. CONCLUSION

We have proposed a new construction algorithm for the discrete-type HMnet and NL-HMnet, and applied it to building language models. From the experimental results, NL-HMnet with the optimum number of states showed lower perplexities than traditional models such as word n-grams and ergodic HMMs. From the comparison with perplexities of both HMnets, NL-HMnet is very good for representing natural language.

To confirm the advantage of NL-HMnet in applying to natural language, we constructed NL-HMnet expressed autopsy documents. From the experimental results, NL-HMnet showed lower perplexities than trigram.

6. REFERENCES

[1] F. Jelinek: "Self-organized language modeling for speech recognition", IBM T. J. Watson Research Center, Unpublished (1985).

[2] S. Deligne and F. Bimbot: "Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams", Proc. ICASSP'95, pp. 169–172 (1995).

[3] G. Bordel, I. Torres and E. Vidal: "Qwi: A method for improved smoothing in language modeling", Proc. ICASSP'95, pp. 185–188 (1995).

[4] T. Kuhn, H. Niemann and E. G. Schukat-Talamazzini: "Ergodic hidden markov models and polygrams for language modeling", Proc. ICASSP'94, pp. 357–360 (1994).

[5] J. Takami and S. Sagayama: "A successive state splitting algorithm for effecient allophone modeling ", Proc. ICASSP'92 Vol. I, pp. 573–576 (1992).

[6] M. K. McCandless and J. R. Glass: "Empirical acquisition of language models for speech recognition", Proc. ICSLP'94, pp. 835–838 (1994).

[7] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, O. Imaichi and T. Imamura: "Japanese morphological analysis system chasen manual", NAIST Technical Report NAIST-IS-TR97007 (1997).

[8] F. Jelinek and R. L. Mercer: "Interpolated Estimation of Markov Source Parameters from Sparse Data", Pattern Recognition in Practice, E. S. Gelsema and L. N. Kanal, North-Holland, pp. 381–397 (1980).

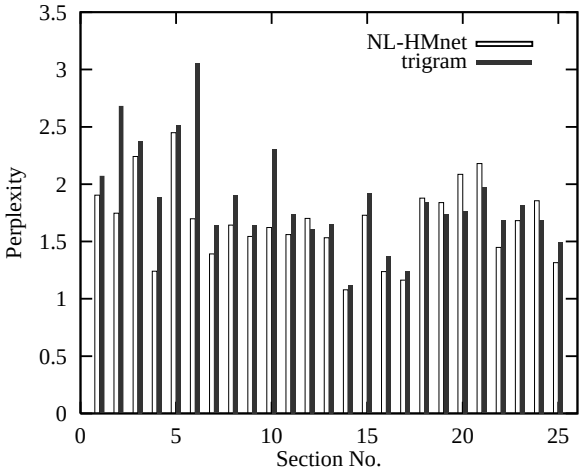

Figure 4: Perplexity for test set document.

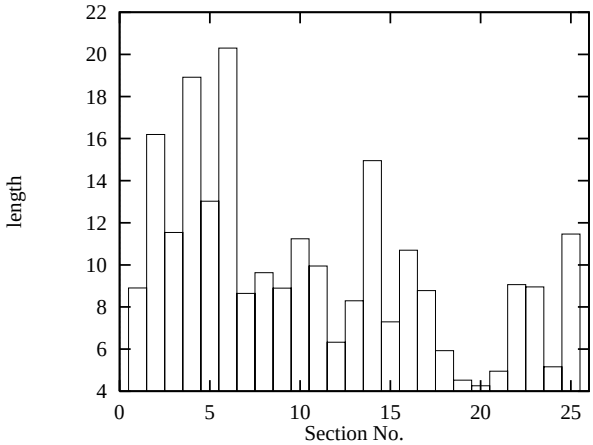

Figure 5: Mean number of words in a sentence.