Automatic Construction of a FSA Language Model and Speech Recognition on it with Dynamic Alternative Path Search

Tsuyoshi MORIMOTO, Shin-ya TAKAHASHI

Abstract—**For a small- or middle-size (around 1,000 words) vocabulary speech recognition, a Finite State Automaton (FSA) language model is widely used. However, defining a FSA model with sufficient coverage and consistency requires much human effort. We already proposed a method to automatically construct a FSA language model from learning corpus by use of FSA DP matching algorithm. Experiment results show that this model attains quite high recognition correct rate for closed data, but only low rate for open data. This is mainly because a necessary path does not appear in a generated FSA. To cope with this problem, we propose a new search algorithm that allows to jump dynamically to an alternative path when speech recognition of some words seems to fail. Experiment results shows the effectiveness of the algorithm.**

Index Terms—**FSA language model, automatic construction of a language model, dynamic path search**

I. Introduction

For a small- or middle-size (around 1,000 words) vocabulary speech recognition, a Finite State Automaton (FSA) language model is widely used. However, defining a FSA model by hands requires much human effort. In some systems, FSA models are automatically generated by converting from regular-grammar type grammar-rules (for example, see HParse[7]), but it is still a very much time- and effort-consuming task to prepare a grammar with sufficient coverage and consistency. When a large amount of learning texts (corpus) is available, on the other hand, a statistical language model such as bi-gram or tri-gram generated from a corpus is quite powerful for speech recognition. For example, most of current dictation systems employ bi-gram or tri-gram language models. However, if the size of corpus is not large enough, reliability of statistical information calculated from the corpus decreases and then so the efficiency of the generated statistical language model (*a sparseness problem*). Furthermore, preparing sufficient amount of spoken texts is generally very much expensive.

On generating a FSA language model from learning data, several methods have been proposed. Note that constructing an acyclic FSA from given data is a simple problem; one can construct a TRIE tree in which common prefix is shared, and then minimize the tree by merging equivalent states. This method, however, is computationally quite expensive. Several method have been proposed to improve the efficiency[2][3], but they are still at a basic research stage and have not applied yet to practical applications. Meanwhile, other kind of approaches have been proposed[4]-[6] aiming at applying to actual speech recognition language modeling. However, since common key technique used in them for improving efficiency is to use stochastic features, *a sparseness problem* mentioned above arises again when corpus size is not large enough.

We already proposed a new method to generate a FSA language model by use of DP (Dynamic Programming) matching method[1]. Experiment results shows that this model can attain high recognition correct rate for closed data, but only low rate for open data. This is mainly because a necessary path does not appear in a generated FSA for an open data. To cope with this problem, we propose, in this paper, a new search algorithm that allows to jump dynamically to an alternative path when speech recognition of some words seems to fail. Experiment results show the effectiveness of the algorithm.

II. Overview of Constructing a FSA Language Model

As mentioned above, we already proposed a method to generate a FSA language model by use of DP matching. Here, we briefly describe an overview of the method and also about speech recognition experiment results conducted on generated FSA models.

A. *Constructing a FSA Language Model*

(a) Before executing DP matching, sentences in a corpus are divided into some number of groups (clusters) based on distances between them. This is because to avoid unnecessary execution of DP matching between dissimilar sentences. There are several ways to calculate a distance between two sentences, but we adopted a following equation for simplicity.

$$d(S_x, S_y) = 1 - \frac{Num(w \mid w \in S_x \cap S_y)}{Num(w \mid w \in S_x \cup S_y)} \qquad (1)$$

where

$d(S_x, S_y)$: distance between sentence S_x *and* S_y

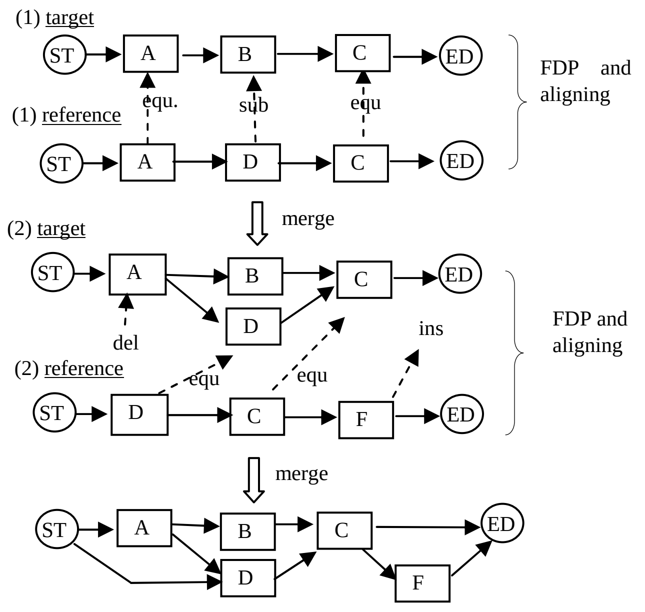
$Num(w)$: number of words such as w

Manuscript accepted January 7, 2009.

T. Morimoto is with the Electronics and Computer Science Department, Fukuoka University, 8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, JAPAN (e-mail: morimoto@ fukuoka-u.ac.jp).

S. Takahashi is with the Electronics and Computer Science Department, Fukuoka University, 8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, JAPAN (e-mail: takahasi@tlsun.tl. fukuoka-u.ac.jp)

(b) For each cluster, one randomly chosen sentence (a target) is converted to a simple one path FSA[1] and is put on a y-axis. Next, another sentence (a reference) is picked up, converted to a one path FSA as well, and is put on a x-axis. Next, two FSA are aligned by DP matching (hereafter, we call this DP matching method as a FSA-DP matching: *FDP*). As a result of *FDP*, relationships between words such as '*equal*', '*substitution*', '*deletion*', or '*insertion*' are obtained. According to these relationships, nodes of a reference FSA are merged to a target FSA (see Fig. 1).



ST: start node ED: end node

Fig. 1. FDP and merging

A target FSA constructed in this way is used as a target FSA in the next *FDP*, but it is no longer one-path FSA. Therefore, nodes in the target FSA are sorted topologically by use of a distance from a start node, and put on y-axis. Next *FDP* is executed according to paths defined by a target FSA as shown in Fig. 2.

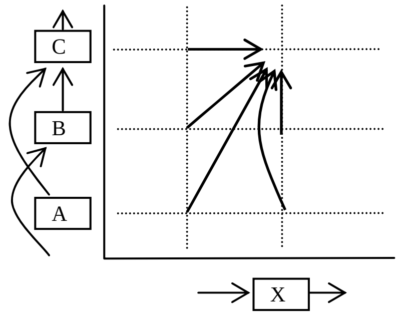


Fig. 2. FSA DP matching

1) A FSA is defined as a DAG (directed-acyclic-graph). A word is represented as a node, and a connection between words is represented as a directed-arc. A FSA has a start node (ST) and an end (ED) node.

A global distance at a point (x, y) is calculated according to equation (2)

$$gd(x,y) = \min\begin{pmatrix} gd(x-1,\,y) \\ gd(x-1,\,\Delta y) \\ gd(x,\,\Delta y) \end{pmatrix} + ld(x,y) \qquad (2)$$

where

$gd(x,y)$: a global distance at a point (x, y)

$ld(x,y)$: a local distance between word w_x and word w_y, and calculated as follows.

$$ld(x,y) = \begin{cases} 0.0 & (w_x = w_y) \\ 1.0 & (w_x \neq w_y) \end{cases}$$

Δy : a location on a y-axis of the previous word.

An each sentence in a cluster is taken out as a reference in turn and the above procedure is applied to repeatedly, and thus the target FSA is incrementally extended.

(c) When the procedure has finished for all clusters, the same number of FSA models with the number of clusters are obtained. Finally, these FSA models are combined into one FSA so that they share a common ST (start) node and an ED (end) node.

B. Basic Performance of a Generated FSA

1) Features of Corpus and Speech Recognizer

We conducted several speech recognition experiments on generated FSA language models. As a corpus, Japanese travel conversation sentences have been collected from four Japanese-English travel conversation textbooks. Here, sentences being too colloquial or fragmental were removed. As number of collected sentences is a slightly small (about 950 sentences), then several new sentences (about 50 sentences) modified little from the originally collected ones were added so that number of sentences becomes 1,000. Features of the corpus are listed in Table 1, and some example sentences appearing in the corpus is shown in Table 2.

Table 1. Corpus

Vocabulary	1,254 words
Sentences	1,000 sentences
Average words per sentence	8.87 words

Table 2. Example sentences

1	*Bizyutsu-kan meguri-no tsua-wa ari-mase-n-ka?* (Is there a tour of art museum?)
2	*Okurimono-you-ni housou-site morae-masu-ka?* (Can I gift wrapped?)
3	*Koko-de tabako-wo sutte-mo ii-desu-ka?* (May I smoke here?)

In speech recognition experiments, HVite[7] is used as a decorder, and a speaker-independent tri-phone HMM of Julius[8] is used as a Japanese acoustic model. As a test data for speech recognition experiments, 60 utterances spoken by three male speaker are used.

2) *Experiment for closed data*

All 1,000 sentences were used as a learning data, and several FSA language models were constructed for different number of clusters. Experiment results are shown in Table 3. In the table, an experiment result using bi-gram language model is also shown for a reference. From this table, we can see that FSA language models constructed by our method have very small branching factors and then attain high speech recognition rate. Especially, a sentence recognition rate is 20 points higher than that of a bi-gram model. As number of clusters increases, a branching factor decreases and then speech recognition rate slightly increases.

Table 3. Speech recognition result (for closed data)

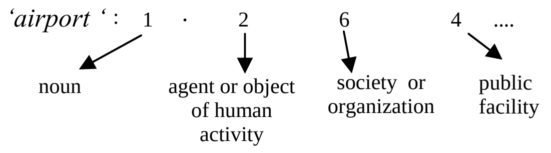
Num. of clusters	Ave. branching factor	Word %correct	Sentence %correct
30	1.45	98.7	90.0
50	1.42	98.7	90.0
70	1.40	98.9	91.7
bi-gram	5.88 (perplexity)	93.0	70.0

3) *Experiment for open data*

We took out 60 sentences as test data from the corpus, and constructed FSA models from the rest 940 sentences. Here, the amount of leaning sentences are absolutely insufficient, simple elimination of test data will cause deletion of several words from a learning corpus. To avoid this problem, noun words appeared in 940 sentences were replaced by appropriate some semantic classes, and then FSA models were constructed by *FDP*. In *FDP*, equation of calculating local distance is modified as next equation.

$$ld(x, y) = \begin{cases} 0.0 \; (w_x = w_y) \\ 0.5 \; (w_x \neq w_y, sem_x = sem_y) \\ 1.0 \; ((w_x \neq w_y, (sem_x \neq sem_y \; or \; w_x \neq noun)) \end{cases} \quad (3)$$

When $ld(x, y)$ is 0.5, a node denoting that semantic class is generated in a *FDP* stage. For converting from a word w_x to a semantic class, we used *Bunrui-Goi-Hyo* (*BGH*) [9] as a thesaurus. In *BGH*, each word is categorized in five levels. We use two decimal places as a semantic code.


'airport ' : 1 · 2 6 4 ....
noun
agent or object of human activity
society or organization
public facility

After completion of *FDP*, a FSA containing some semantic classes is obtained. In the next step, these nodes are expanded to the all of nouns which belong to that semantic class, and thus the final FSA (a word level FSA) is generated.

Experiment result is shown in Table 4. Compared with the result in Table 3, we see that speech recognition correct rates, especially sentence recognition rates, drop very much, though they are still better than that of bi-gram. This is because some words disappeared in the final FSA model an/or even some paths did so because the size of training data was fairly small. We investigated the reasons for recognition errors, and the result is shown in Table 5. From this table, we see that many of sentence recognition errors are due to deletion of corresponding paths.

Table 4. Speech recognition result (for open data)

Num. of clusters	Ave. branching factor	Word %correct	Sentence %correct
30	1.70	78.8	26.7
50	1.67	79.9	21.7
70	1.68	79.4	28.3
bi-gram[2)]	6.36 (perplexity)	58.4	3.3

2) When constructing a bi-gram language model, 60 sentences are not eliminated but back-off (Witten-Bell) smoothing method is applied.

Table 5. Details of sentence recognition error

Due to	Num. of sentences
Deletion of words	4 (23.5%)
Deletion of paths	13 (76.5%)
Total	17 (100%)

III. Dynamic Alternative Path Search

A. Dynamic Alternative Path Search

As described above, main reason of big fall of sentence recognition rate is due to deletion of necessary paths. To cope with this problem, the following dynamic path search algorithm was incorporated into the speech recognizer's decorder (see Fig. 3).

- When recognition of each word is finished, the recognition score of the word is examined whether the recognition result seems to be correct or incorrect.
- If seemed to be incorrect, words that are the same with the one prior word on the different paths are searched for.
- If found, recognition information (recognition score and path information) of that prior word was copied and attached to those words, and searches are restarted from these words.
- Search on the current path is also continued to avoid misjudgment on incorrectness of recognition result.

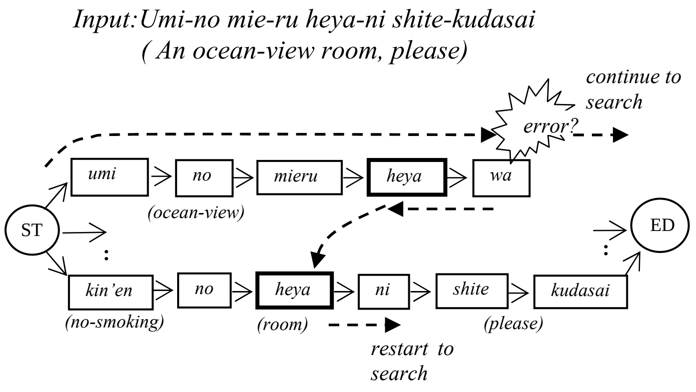


Fig. 3. Dynamic alternative path search

Here, a worrisome problem is that how we should detect a recognition error on a certain word, because an acoustic score is not always reliable. For the moment, we use the following criteria for simplicity.

- For each recognition path, a score per frame (*spf*) of that path is maintained.
- When recognition for a certain word is finished, a *spf* of the word is calculated.
- When the following inequality holds, the recognition of that word is regarded to be wrong, otherwise to be correct. Here, β is decided experimentally.

$$(\text{ a } spf \text{ of a word}) \leqq \beta \times (\text{a } spf \text{ of a path}) \qquad (4)$$

Here, log-likelihood is used as a score, in the same way in HVite.

B. Experiment Result

We conducted speech recognition experiment to see the effectiveness of the new search algorithm. The conditions of the experiment are the same as those for open data experiment described in Ⅱ-B-3). The experiment was done only for 70 clusters, and the result is shown in Table 6.

Table 6. Speech recognition result with dynamic path search

Num. of clusters	Word %correct	Sentence %correct
70	82.6	40.0

Compared with the result in Table 4, we can see the following points;

(a) word %correct rises up slightly (3.2 points up)

(b) sentence %correct rises up remarkably (11.7 points up).

The reason for (b) might be that common sentence final expressions are frequently used in spoken Japanese. If recognition failure occurs at some word in later place of a utterance, an alternative path which has an appropriate sentence final expression for the utterance might exist.

IV. Conclusion

We developed a new method to construct a FSA language model automatically. Experiment results shows that this model attains quite high recognition correct rate for closed data, but only low rate for open data. This is mainly because an necessary path does not appear in a generated FSA. To cope with this problem, we propose a new search algorithm that allows to jump dynamically to an alternative path when speech recognition of some words seems to fail. Experiment for open data, sentence %correct is improved remarkably (11.7 points up). This shows the effectiveness of this algorithm.

An FSA language model is quite convenient and effective for a middle-sized speech recognition. However, it has a big fault that speech recognition fails even when a recognition path slightly stray from a path defined in a model. The proposed method could be thought to be a kind of a back-off method for a FSA language model to cope with this problem. From this point of view, there will be also other methods as following.

(a) When forward recognition fails, backward recognition is carried out, and two recognized fragments are tried to be glued.

(b) When recognition fails somewhere after a pause, all possible paths that follow a pause will be tried to be recognized.

In future, we will study these methods as well.

References

[1] T. Morimoto and S. Takahashi, "Automatic Construction of FSA Language Model for Speech Recognition by FSA DP-Matching", Trends in Intelligent Systems and Computer Engineering (Edtd. by O. Castillo et al.), Springer, 515-524, 2008

[2] K. J. Lang, B. A. Pearlmutter, and R. Price, "Results of the Abbadingo One DFA Learning Competition and a New Evidence Driven State Merging Algorithm", Proc. of Int. Colloquium Grammatical Inference, 1-12, 1998

[3] S. M. Lucas and T. J. Reynolds, "Learning Deterministic Finite Automata with a Smart State Labeling Evolutionary Algorithm", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.27, No.7, July 2006

[4] C. Kermorvant, C. de la Hinguera, and P. Dupont, "Learning Typed Automata from Automatically Labeled Data", Journal Électronique d'Intelligence Artificielle, Vol. 6, No.45, 2004

[5] J. Hu, W. Turin, and M. K. Brown, "Language Modeling with Stochastic Automata", Proc. of ICSLP-1996, 1996

[6] G. Riccardi, R. Pieraccini, and E. Boccieri, "Stochastic Automata for Language Modeling", Computer Speech and Language, Vol.10, No. 4, 265-293, 1996

[7] S. Young et al., "The HTK Book (for Ver. 3.0)", 1999 (http://htk.eng.cam.ac.uk/)

[8] T. Kawahara, A. Lee, K. Takeda, K. Itou, and K. Shikano, "Recent Progress of Open-Source LVCSR Engine Julius and Japanese Model Repository -- Software of Continuous Speech Recognition Consortium -- ", Proc. of ICSLP-2004, 2004 (http://julius.sourceforge.jp/en/julius.html)

[9] National Language Research Institute, "Bunrui-Goi-Hyo (Word List by Semantic Principles)", Syuei-Shuppan, 1994 (in Japanese)