Robust/Fast Out-of-Vocabulary Spoken Term Detection By N-gram Index with Exact Distance Through Text/Speech Input

Nagisa Sakamoto* and Seiichi Nakagawa*
*Toyohashi University of Technology, Toyohashi, Aichi 441-8580, Japan
E-mail: {sakamoto,nakagawa}@slp.cs.tut.ac.jp

***Abstract*—For spoken term detection, it is very important to consider Out-of-Vocabulary (OOV). Therefore, sub-word unit based recognition and retrieval methods have been proposed. This paper describes a very fast Japanese spoken term detection system that is robust for considering OOV words. We used individual syllables as sub-word unit in continuous speech recognition and an n-gram index of syllables in a recognized syllable-based lattice. We proposed an n-gram indexing/retrieval method in the syllable lattice for attacking OOV and high speed retrieval. Specially, in this paper, we redefineded the distance of the n-gram and used trigram, bigram and unigram that instead of using only trigram to calculate the exact distance. In our experiments, where using text and speech query, we achieved to improve the retrieval performance.**

I. Introduction

Recently, with the growth of information and communication technology, multimedia data such as audio and video can be found on the Web. Required information can be found with an existing textual search engine if the target data consist of textual information such as transcribed broadcast news and newspapers; however, an efficient spoken document retrieval (SDR) or spoken term detection (STD) method is still not established, because spoken documents have specific problems such as recognition errors and out-of-vocabulary(OOV) terms. The aim of this research is to develop a robust and efficient STD method for OOV queries through text and speech input. A standard STD method is using textual search to LVCSR transcripts. However, OOV terms are not registered in a dictionary of speech recognizer. Therefore it is impossible to detect the OOV term with an existing text search engine because the word is not given as an output in the recognition result of an LVCSR. The advantage of using a sub-word unit based speech recognition system is that it can ignore grammatical/lexical constraints and recognize any OOV terms[1], [2], [3].

However, for retrieving speech-based documents, some problems to be solved remain, such as OOV and recognition errors. In Chinese, syllable-unit (440 syllables in total) has often been used as a basic unit of recognition/retrieval[4]. Japanese consists of only about 110 syllables, therefore the syllable unit is suitable for the spoken retrieval of OOV words. In addition, other retrieval methods based on elastic matching between two syllable sequences have been tried for considering recognition errors[5].

It is necessary to prune the many detection candidates. Typically, as with the dynamic time warping (DTW) method, a string is used to elastically match candidates for pruning. However, DTW processing is more time consuming than index base search processing. Instead of DTW, we used the n-gram array with distance measure that accounts recognition errors in the syllable recognition lattice[6], [7]. We showed a significant improvement of processing time using this method. In this paper, especially, we propose the robust/efficient attacking method for substitution, insertion and deletion errors in the syllable lattice, in other words, n-gram index with exact distance. We also propose to split the query into not only the trigram but also bigram or unigram and applied it to spoken term detection through text and speech input.

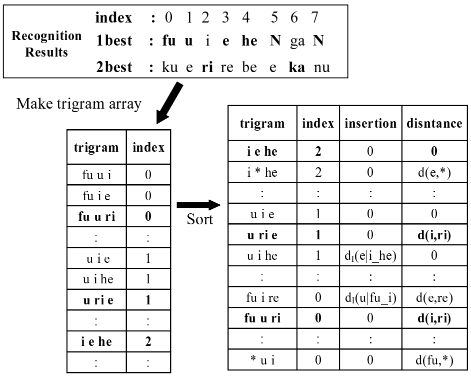
The remainder of this paper is organized as follows: In Section 2, we describe our retrieval system. In Section 3, we define the exact distance for the n-gram and, in Section 4, describe query dividing method. Evaluation results are given in Section 5 and a conclusion in Section 6.

II. High-speed OOV word retrieval method by n-gram with distance

In this study, we use an n-gram of syllables for Spoken Term Detection (STD), in particular for OOV terms. N-gram information of syllables is maintained by a data structure called an n-gram array that consists of index and syllable distance information within each n-gram. Fig. 1 illustrates how a trigram array is arranged. First, the appearance position of syllables in a spoken document is allocated. Then an n-gram of the syllable at every appearance position is constructed. Next, the n-gram is sorted in a lexical order so as to search it quickly using a binary search algorithm. The column of ”index” in Fig. 1 represents a position of trigram in a spoken document, ”insertion” represents newly defined measure, which means the distance of insertion errors in an indexed trigram. ”distance” denotes the substitution distance between the first candidate and an indexed candidate.

The search process on an n-gram array is divided into 3 steps. First, a query is converted into a syllable sequence. Second, an n-gram of the query is constructed. Finally, the query is retrieved from the n-gram array. A query consisting of more than n+1 syllables is retrieved by a combination of n-grams. How to divide the query is described in Section 4.

The purpose of considering the gap between the appearance positions is to deal with mis-recognitions. This is described in



	index	: 0 1 2 3 4 5 6 7
Recognition Results	**1best**	: **fu** **u** i **e** **he** **N** ga **N**
	2best	: ku e **ri** re be e **ka** nu

trigram	index
fu u i	0
fu i e	0
fu u ri	**0**
:	:
u i e	1
u i he	1
u ri e	**1**
:	:
i e he	**2**
:	:

trigram	index	insertion	disntance
i e he	**2**	0	**0**
i * he	2	0	d(e,*)
:	:	:	:
u i e	1	0	0
u ri e	**1**	0	**d(i,ri)**
u i he	1	d_I(e|i_he)	0
:	:	:	:
fu i re	0	d_I(u|fu_i)	d(e,re)
fu u ri	**0**	0	**d(i,ri)**
:	:	:	:
* u i	0	0	d(fu,*)

Fig. 1: Procedure for making trigram array

details in the next section.

III. Solving mis-recognized sub-word problem for OOV detection

A. Substitution error

To handle substitutions errors, we use an n-gram array constructed from the m-best of the syllable lattice[6]. An n-gram array is constructed by using the combination of syllables in the m-best syllable lattice. Thus, for one position in the lattice, there are m^n kinds of n-gram. For example, even if the recognition result of the 1-best is “fu u i e he N ga N” having recognition errors, we can search for the query “fu u ri e he N ka N (“Fourie Transform” in English)”, if a correct syllable is included in the m-best. We used HMM based Bhattacharrya distance between syllables [6] as the local distance between the 1st candidate and other candidate. The “fu u ri” distance is calculated as distance between “fu u ri” of target trigram and “fu u i” of the 1-best trigram, where the distance is $d_s(ri, i)$.

Even if we use the syllable lattices, some substitution errors may not be contained in the lattice. Therefore, we introduced the dummy syllable symbol or “wild card”[7]. A dummy syllable is represented by “*” . The dummy syllable can match with any syllable that is not contained in the m-best recognition results. For example, if the recognition result of the m-best does not include “C”, the original method can not search for the query “ABCD”. At this case, the query using the dummy syllable has n-gram as AB*, A*C and *BC, and we can retrieve the query “ABCD”. Therefore, the recall rate is increased. On the other hand, the method has the potentiality to decrease the precision rate. This problem is addressed by increasing the distance between “*” and any other syllable, where only one dummy syllable is allowed in a trigram. We should notice that this approach is different from a one distant bigram index method. We used the exact definition of $d_S(e, *)$ as $\lambda \times d_S(syllable\ of\ query, e) + \eta$ after finding the index, in other words, instead of a constant value[7] as follows:

$$d_S(*,*)=\lambda\times d_s(\ syllable\ of\ query,\ best\ syllable\ for\ the\ dummy\ syllable\ ) + \eta \qquad (1)$$

,where λ and η denotes an penalty for using the dummy syllable. For example, if “query” is “i me he”, the distance between “me” in the query and “*” in the lattice is defined as $\lambda \times d_s(me, e) + \eta$ in Fig. 1.

B. Insertion error

To address the insertion errors, we make an n-gram array that permits a one-distant n-gram. Considering the gap between appearance positions deals with the error. Even if the recognition result is ”fu ku u ri e he N ka N” having an insertion error ”ku”, we can search for the query ”fu u ri e he N ka N”, if the n-gram array that considers a one-distant n-gram is allowed. Therefore, it is possible to deal with one insertion error within every n-gram. The trigram of ”fu u ri” is constructed as a skipped trigram from ”fu ku u ri”, when ”ku” is regarded as an insertion error.

The insertion distance is defined, instead of a constant value[6], [7] as follows:

$$d_I(C_2V_2|C_1V_1\_C_3V_3)=\min\left\{\begin{array}{l} d_S(C_1V_1, C_2V_2) \\ d_S(V_1, C_2V_2) \\ d_S(C_2V_2, C_3V_3) \end{array}\right\}+\delta_I \qquad (2)$$

where C_2V_2 (C=consonant, V=vowel) denotes the insertion syllable, and C_1V_1 and C_3V_3 denote the left context and right context, respectively. ”δ_I” denotes an insertion penalty. ”$d_S(V_1, C_2V_2)$” means that ”a part of vowel V_1” is mis-separated into the vowel and an inserted syllable.

C. Deletion error

To handle the deletion errors, we search for the query as above while allowing for the case where one syllable in the query is deleted.

Even if the recognition result is ”fu u e he N ka N” having a deletion error, we can search for the query ”fu u ri e he N ka N”, if a syllable (’ri’) in the query is deleted.

When a query consisting of syllables more than 2n must consider deletions of two syllables, the errors for a long query can not be corrected simply by deleting one syllable. In such a case, the query is divided into two parts, and they are made to drop out by one syllable, and retrieved. For example, for the recognition result of ”fu ri e he N ka”, it is retrieved by considering one deletion of ”fu u ri e” and of ”he N ka N” in the case of $n = 3$, respectively.

The deletion distance in a query is defined instead of a constant value[6], [7] as follows:

$$d_D(C_2V_2|C_1V_1\_C_3V_3)=\min\left\{\begin{array}{l} d_S(C_1V_1, C_2V_2) \\ d_S(V_1, C_2V_2) \\ d_S(C_2V_2, C_3V_3) \end{array}\right\}+\delta_D \qquad (3)$$

IV. Dividing the query into n-gram

In our previous studies[6], [7], we used only trigram to split the query and make indexes. This was a problem in this way syllable overlap occures between divided trigrams. Therefore, we also introduced bigram and unigram. For example, a query consisting of less than 6 syllables but more than 4 syllables is

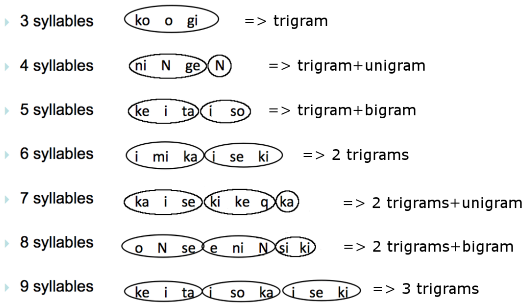


Fig. 2: Example of query division into trigram

separated into traigram and bigram or unigram for the first and second halves. Thus, the query is retrieved from the trigram array and bigram array or unigram array. The retrieved results are merged by considering whether the position at which the detection result occurred in the first and second halves is the same. Similarly, a query with less than 9 syllables but more than 7 syllables is retrieved by a sequence of syllables by dividing the query into three parts (Fig. 2). For example, when a query consists of six syllables, "i mi ka i se ki" in Fig. 2, the query's syllable sequence is divided into two trigrams; "i mi ka" and "i se ki." If the first term, "i mi ka," is detected at $s_1 \sim t_1$ with a distance less than a thereshold, that is, index position $= s_1$, and the second term, "i se ki," is detected at $t_1 + 1 \sim u_1$ with a distance less than a threshold, that is, index position $= t_1 + 1$, then "i mi ka i se ki" is detected at $s_1 \sim u_1$. For a query consisting of five syllables, "ke i ta i so" in Fig. 2, the query sequence is divided into a trigram and a bigram; "ke i ta" and "i so". If the first term "ke i ta" is detected at $s_2 \sim t_2$ and the second term "i so" is detected at $t_2 + 1 \sim u_2$, then "ke i ta i so" is detected at $s_2 \sim u_2$.

The query term is detected, if the following distance is lower than a pre-determined threshold. Strictly speaking, the threshold depends on the query length.

$$\frac{\alpha \times \sum d_S + \beta \times \sum d_I + \gamma \times \sum d_D}{number\ of\ syllables} \tag{4}$$

,when d_S, d_I and d_D denotes the distans for substitution, insertion, and delition errors, respectively.

V. Evaluation and Results

A. Experimental setup

For our experimental data, we used the 44 hours of core data in the CSJ (Corpus of Spontaneous Japanese) corpus as experimental data[8] and SPOJUS++[9] developed in our laboratory as the LVCSR. The context-dependent syllable-based HMMs were trained on 2707 lectures within the CSJ corpus excluding the core data. We used a left-to-right HMM, consisting of four states with self loops, and has four Gaussians with full covariance matrices per state. We used an OOV term set of 50 queries in the core data which have been reported by Ito et al[8]. Continuous syllable recognition was performed by context-dependent 928 syllable-based HMMs and syllable-based 4 grams as a language model.

TABLE I: Recognition results (%)

(a) Spoken documents

output	Del	Ins	Subs	Corr	Acc
Syllable (1best)	3.9	3.6	12.5	83.6	80.0
Syllable (3best)	3.9	2.2	6.9	89.1	86.9
Syllable (5best)	4.1	1.9	4.9	91.0	89.1

(b) Spoken queries

speaker	Del	Ins	Subs	Corr	Acc
1	5.3	4.4	24.4	70.4	66.0
2	1.3	11.1	19.2	79.6	68.5
3	2.2	5.4	20.5	77.4	72.0
4	7.9	3.1	29.2	62.8	59.7
5	2.7	6.1	22.0	75.4	68.8
6	2.1	5.6	18.5	78.3	72.8
Ave.	3.7	5.9	22.3	74.0	67.9

The F-value and MAP which are our measures are definded to be

$$F - value = \frac{2 \cdot Recall \cdot Precision}{Recall \cdot Precision} \tag{5}$$

$$MAP = \frac{1}{Q} \sum_{i=1}^{Q} AveP(query_i) \tag{6}$$

where the Q is the number of the query and the $AveP(query_i)$ is the average precision of the $query_i$.

The syllable recognition rates for spoken document are summarized in Table I(a). Considering 5-best candidates, the correct rate was about 91%

For spoken queries, six male speakers uttered the set of 50 queries two times. The recognition results of the six speakers condidering 1-best candidate are summarized in TableI(b). The rate for spoken queries was worse than that for spoken spoken documents.

B. text-based query for spoken documents

The 50 OOV queries and 238 occurrences in total were retrieved using syllable recognition results. Fig. 3 shows the retrieval results by using the baseline DTW-based method between syllable seaquences, original trigram index method[7], the method based on new distance definitions of Equations (1), (2) and (3), and n-gram (trigram, bigram, unigram) index method. The DTW-based method compares two syllable sequences between a query and spoken documents. For comparison, we used Bhattacharrya distance and edit distance as local distance measure between syllables for DTW approach, and found that the Bhattacharrya distance was superior to the edit distance. By introducing the exact distance for syllables of trigram defined as Equations (1)(2)(3), we improved the ritieval performance from $F{-}value = 0.563$ ($MAP = 0.561$) to $F - value = 0.611$ ($MAP = 0.635$). Furthermore, we got the best performance of $F - value = 0681$ ($MAP = 0.649$), but the index size is increases from 1.5GB to 1.7GB. These F-values outperformed the value of the baseline DTW-based approach.

C. spoken query for correctly transcribed documents

Next, we applied our proposed method to spoken query for correctly transcribed documents (text documents) and automatically transcribed spoken documents. Fig. 4 illustrates the retrieval results of OOV terms for correctly transcribed

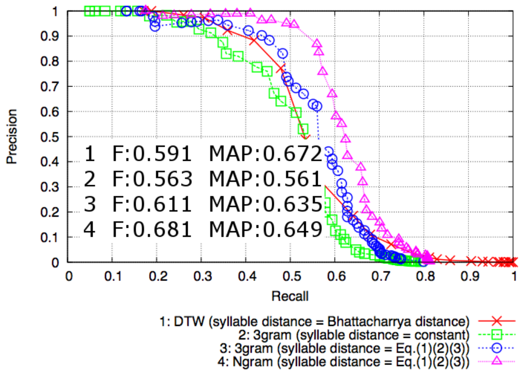


Fig. 3: Retrieval results of OOV for spoken document through text input

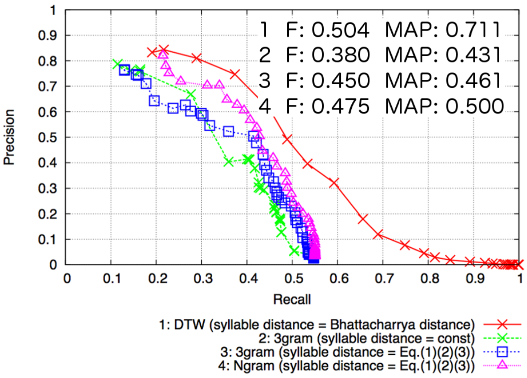


Fig. 4: Retrieval results of OOV for text document through apoken input

documents through spoken input. Of cource, the performance strongly depends on the syllable recognition rate for spoken queries. In this experiment, we did not use plural candidates for syllable recognition result, but used only the best candidate. Therefore, it is very difficult to retrieve OOV terms by using trigram syllable indices, because the rate including one or two correct syllables in a three syllable is very low. On the other hand, we can hit any syllables by using our baseline method (DTW). So we also improved the perforamnce by our proposed method in this experiment, but it was still worse than DTW method.

D. spoken query for spoken documents

Fig. 5 illustrates the retrieval results of OOV terms for spoken documents through spoken input. Furthermore, this task is more difficult than the above two tasks described in Sections V-B and V-C. For this task, our proposed method was also better than the conventional DTW method for the average of six speakers as shown in Fig. (5) on F-values.

VI. Conclusion

In this paper, we could retrieve OOV terms for spoken documents by typed queries at 0.681 of F-value. It was better than the conventional method based on DTW. The system can deal with all kinds of queries such as containing substitution error, insertion error and deletion error. Additionally, we could improve the recall rate by introducing a dummy syllable symbol or exact distances in the syllable lattice. To solve the problem of a large amount of index, we used the constraint of substitution actions when constructing the index. By applying these functions, we could implement a very robust/fast OOV term retrieval method. The retrieval time per query for 44 speech hours was about 80ms and it was 18 times faster than that of the DTW method. We also applied the proposed approach to spoken queries for text/spoken documents.

An important topic for future work is to improve the recall performance. One way to improve the recall rate is to relax the constraints of errors and to use only low confidence parts as OOV candidates from the results of the LVCSR[10]. Another way is to improve the syllable recognition rate by combining the results of several decoders[10], [11], [12]. For spoken query, we had better use plural candidates, but it consumes much computation time. Finally, we may use the syllable's likelihood obtained from the decoder, instead of the syllable distance to improve the retrieval accuracy.

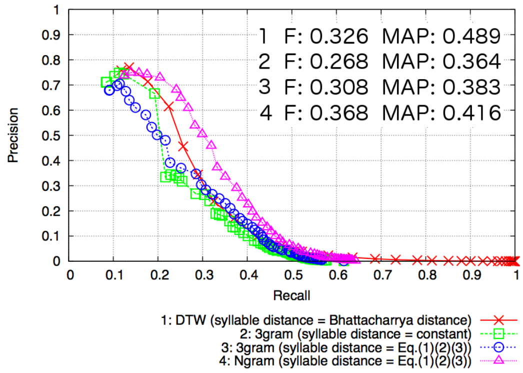


Fig. 5: Retrieval results of OOV for spoken document through spoken input

References

[1] K. Ng, "Towards robust methods for speech document retrieval," ICSLP, 1998, pp. 1088–1091.

[2] R. Rose, A. Norouziam, and et.al., "Sub-word based spoken term detection in audio course lectures," ICASSP, 2010, pp. 5282–5285.

[3] T. Mertens and D. Schneider, "Efficient subword lattice retrieval for german spoken term," ICASSP, 2009, pp. 4885–4888.

[4] H. Wang, "Experiments in syllable-based retrieval of broadcast news speech in Mandarin Chinese," Speech Communication, 2000, vol. 32, pp. 49–60.

[5] M. Wechsler, E. Munteanu, and P. Schauble, "New techniques for open-vocabulary spoken document retrieval," SIGIR, 2008, pp. 20–27.

[6] K. Iwami, Y. Fujii, K. Yamamoto, and S. Nakagawa, "Out-of-vocabulary term detection by n-gram array with distance from continuous syllable recognition results," SLT, 2010, pp. 200–205.

[7] S. Nakagawa, K. Iwami, and K. Yamamoto, "A robust/fast spoken term detection method based on a syllable n-gram index with a distance metric," Speech Communication, 2013, vol. 35, pp. 470–485.

[8] Y. Itoh, and et.al., "Constructing Japanese test collections for spoken term detection," INTERSPEECH, 2010, pp. 677–680.

[9] Y. Fujii, K. Yamamoto, and S. Nakagawa, "Large vocabulary speech recognition system: Spojus++," MUSP, 2011, pp. 110–118.

[10] H. Nishizaki and S. Nakagawa, "Japanese spoken document retrieval considering OOV keywords using OOV detection processing and word spotting," HLT, 2002, pp. 144–151.

[11] S. Natori, H. Nishizaki, and Y. Sekiguchi, "Japanese spoken term detection using syllable transition network derived from multiple speech recognizers outputs," INTERSPEECH, 2010, pp. 681–684.

[12] Y. Itoh, H. Nishizaki, and et.al., "Constructing japanese test collections for spoken term detection," INTERSPEECH, 2010, vol. 4, pp. 677–680.