Spoken Term Detection Based on a Syllable N-gram Index at the NTCIR-11 SpokenQuery&Doc Task

Nagisa Sakamoto
Toyohashi University of Technology
1-1 Hibarigaoka
Toyohashi-shi
Aichi,440-8580
sakamoto@slp.cs.tut.ac.jp

Kazumasa Yamamoto
Toyohashi University of Technology
1-1 Hibarigaoka
Toyohashi-shi
Aichi,440-8580
kyama@slp.cs.tut.ac.jp

Seiichi Nakagawa
Toyohashi University of Technology
1-1 Hibarigaoka
Toyohashi-shi
Aichi,440-8580
nakagawa@slp.cs.tut.ac.jp

ABSTRACT

For spoken term detection, it is crucial to consider out-of-vocabulary (OOV) and the mis-recognition of spoken words. Therefore, various sub-word unit based recognition and retrieval methods have been proposed. We also proposed a distant n-gram indexing/retrieval method for spoken queries, which is based on a syllable n-gram and incorporates a distance metric in a syllable lattice. The distance represents confidence score of the syllable n-gram assumed the recognition error such as substitution error, insertion error and deletion error. To address spoken queries, we propose a combination of candidates obtained through some ASR systems which are based on syllable or word units. We run some experiments on the NTCIR-11 SpokenQuery&Doc Task and report the evaluation results.

Team Name

NKGW

Subtasks

SQ-STD (Japanese)

Keywords

NTICIR-11, spoken term retrieval, syllable recognition, n-gram, Bhattacharyya distance

1. INTRODUCTION

The wide availability on the web of such multimedia data as audio continues to grow. Information can be found using an existing textual search engine if the target data are comprised of such textual information as transcriptions of broadcast news or newspapers. However, efficient robust spoken document retrieval (SDR) or spoken term detection (STD) methods have not yet to be established, since system designers face specific problems, such as recognition errors and out-of-vocabulary(OOV) terms that not appear in the word lattices generated by ASR systems. The SDR task, which seeks suitable documents or passages based on the query, is usually performed using STD results. The aim of this research is to develop a robust and efficient STD method for spoken queries.

For retrieving speech-based documents for text queries, some problems to be solved remain, such as OOV and recognition errors. In German, the retrieval method based on the weighted Levenshtein distance between syllables (words consist of only one syllable in a ratio of half)[1] has been proposed. In Chinese, syllable-unit (440 syllables in total) has often been used as a basic unit of recognition/retrieval[2]. 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[3]. Phoneme based n-gram has been proposed for various retrieval methods, usually with bag of words or partial exact matching[4, 5]. For document retrieval, Chen et al[6] used skipped (distant) bigrams such as s_1-s_3, s_2-s_4 for the syllable sequence of $s_1 s_2 s_3 s_4$. Phoneme recognition errors such as substitution errors have not been explicitly considered for OOV term retrieval[7, 8].

Typically, as with the dynamic time warping (DTW) method, a string is used to elastically match candidates for pruning. Katsurada et al. proposed a fast DTW matching method based on suffix array[9]. Kanda et al. [10] proposed a hierarchical DTW matching method between phoneme sequences, where a coarse matching process is followed by fine matching. However, their method still consumes a great deal of computation time and memory storage. Recently, Saito et al.[11] also proposed a coarse/fast retrieval method based on trigram matching results which were calculated in advance, and it is followed by the fine DTW matching. This method consumes huge computation in advance.

For Fast/Robust STD, we used the n-gram index with distance measure that accounts three kinds of recognition errors in the syllable recognition lattice[12, 13]. First, to handle substitution errors, we use a trigram array that considers the m-best and dummy syllables in the syllable lattice. Second, to tackle insertion errors, we create an n-gram array that permits a one-distant n-gram. Finally, to address deletion errors, we search for edited queries from which one syllable has been deleted.

For spoken queries, many works have been investigated directly matching with acoustic features obtained from documents and queries, which is referred to as query-by-example STD[14, 15]. Researchers focused on features that are frame-level phone posterior probabilities[16], or the HMM pattern configuration[17]. It is not necessary the speech data with annotated data in these approaches, so when the language of data is unknown, it seems like a very attractive[18]. We focus on Japanese speech data with labels, therefore, it is possible to utilize the most of them as making acoustic mod-

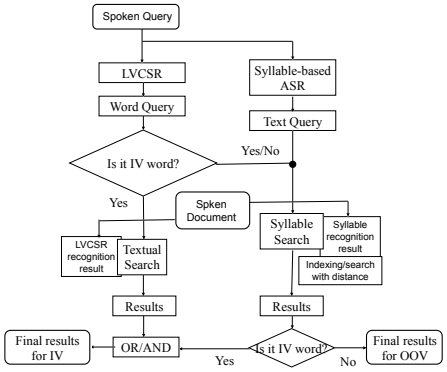


Figure 1: Flow chart of proposed technique

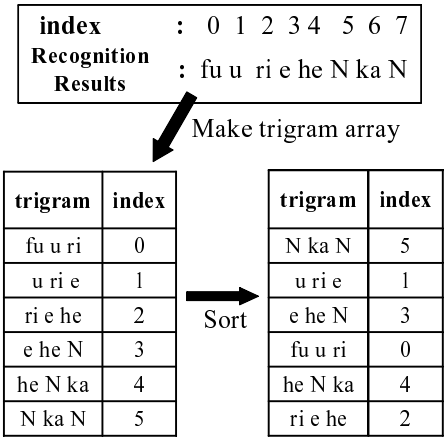


Figure 2: n-gram array indexing procedure (n=3) To simplify, the recognition result is represented by only the first candidate (1-best)

els. Makino et al.[19] presented a matching method using two-pass DTW for spoken queries. First they performed sub-word level matching and second they conducted more accurate matching on state-level.

Our system, which avoid matching between feature parameter sequences through sub-word based ASR systems, is novel in comparison with other studies in order to reduce the search time. In this paper, we extend this method to spoken queries from text queries. The remainder of this paper is organized as follows. In Section 2, we describe our retrieval system and, evaluation results are given in Section 3 and a conclusion in Section 4.

2. PROPOSED METHOD

In this section, we describe a method for spoken term detection to handle In-Vocabulary (IV) terms, Out-Of-Vocabulary (OOV) terms and mis-recognition. We obtain a word sequence through LVCSR system and n-gams from syllable based lattices through ASR systems, thus the system can detect IV and OOV words. To address mis-recognition errors, we construct syllable based N-grams assumed the recognition error. A query is represented by a sequence of words/syllables and a spoken query is recognized as a word sequence or a syllable sequence by using ASR. The detail of our method is described in below.

2.1 System overview

A flow chart of the search process is illustrated in Fig. 1[20]. First we transform spoken queries to text queries through ASR systems. Spoken documents and spoken queries are recognized by an LVCSR for IV words and by a continuous syllable recognition system for dealing with OOV words and mis-recognized words, and then the indexing is applied to the lattice. A search for OOV terms or mis-recognized words using the N-grams in the syllable lattice is described in below.

A query consisting of IV words is retrieved using a standard text search technique from the LVCSR results. To handle mis-recognition errors of the LVCSR, the system also searches spoken terms in IV using the same syllable-based method as OOV term detection and combine the results. "OR" operation in Fig. 1 increases *Recall* rate and "AND" operation increases *Precision* rate, respectively. Due to the mis-recognition of spoken queries, however, it may be difficult to correctly classify the IV terms and OOV terms.

Spoken term detection is executed through the two processes: 1) Indexing and 2) Search. In the indexing process, the N-gram information of syllables is maintained in a data structure called an N-gram array that consists of index and syllable distance information for each N-gram. Fig. 2 illustrates how a trigram array is arranged. First, the appearance positions of the syllables in a recognized syllable lattice for a spoken document are located. Then an n-gram of the syllable is constructed at every appearance position. Next, the n-gram is sorted in lexical order so that it can be searched for quickly using a binary search algorithm. In previous studies, we used only trigram array[12, 13]. We after proposed the extended method using trigram, bigram and unigram array[20].

The search process for an n-gram array includes three steps. First, a query is converted into a syllable sequence. Second, an n-gram of the query is constructed. Finally, the n-gram in a query is retrieved from the n-gram array. A query consisting of more than 4 syllables is retrieved using a combination of n-grams. A query consisting of less than 6 syllables but more than 4 syllables is separated into trigram 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. 3). For example, when a query consists of six syllables, "i mi ka i se ki" in Fig. 3, 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 threshold, 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. 3, 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

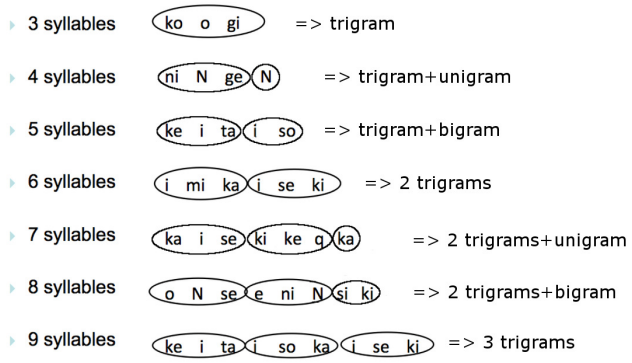


Figure 3: Example of query division into trigram

lower than a pre-determined threshold. Strictly speaking, the threshold depends on the query length.

$$\frac{\alpha \times \sum d_S + \beta \times \sum d_I + \gamma \times \sum d_D}{number\ of\ syllables} \quad (1)$$

,when d_S, d_I and d_D denotes the distances for substitution, insertion, and deletion errors, respectively.

2.2 Substitution error

To handle substitutions errors, we use an n-gram array constructed from the m-best of the syllable lattice[12]. 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 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[13] 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 are not contained in the lattice. Therefore, we introduce the dummy syllable symbol or "wild card"[20]. 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 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 $d_S(syllable\ of\ query, e) + \delta_*$ after finding the index, in other words, instead of a constant value as follows:

$$d_S(*,*) = \lambda \times d_S(\ syllable\ of\ query,\ best\ syllable\ for\ the\ dummy\ syllable\ ) + \eta \quad (2)$$

,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 $\lambda \times d_s(me, e) + \eta$.

2.3 Insertion error

To address the insertion errors, we make an n-gram array that permits a one-distant n-gram[12]. 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 as follows[20]:

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

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.

2.4 Deletion error

To handle the deletion errors, we search the query as above while allowing for the case where one syllable in the query is deleted[12].

Even if the recognition result is "fu u e he N ka N" having a deletion error, we can search 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 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 as follows[20]:

2.5 ASR of spoken query and retrieval

If a spoken query is received by the system after ASR processing, we can treat the query as a text query by considering a word sequence or a syllable-lattice generated from ASR systems. Although the lattice for spoken query may include insertion or deletion errors, we can attack them by the method described in the previous sections. However, this framework is strongly dependent on the performance of ASR, and then the performance of STD with spoken query is significantly lower than one with text query[21]. In the previous research[22], we proposed a combination of candidates obtained from multiple utterances or through different ASR systems to improve the recall score. We propose the following two approaches for spoken queries.

(a) IV/OOV classification:

For a spoken query, we cannot know whether it belongs to IV vocabulary or OOV vocabulary unlike a text query. Therefore, we should decide it, for example, by a matching distance between the syllable sequence of a recognized word by LVCSR and the syllable sequence by continious sylla-

Table 1: Syllable Recognition Rate (SRR) and Word Recognition Rate (WRR) of spoken document and spoken query

(a) JULIUS

meature	document		query	
	Corr	Acc	Corr	Acc
SRR	0.796	0.711	0.766	0.622
WRR	0.696	0.546	0.680	0.027

(b) SUPOJUS

meature	document		query	
	Corr	Acc	Corr	Acc
SRR	0.733	0.694	0.701	0.680
WRR	0.636	0.562	0.580	0.330

ble ASR. In order to calculate this distance, we performed the DTW method and datermined the IV/OOV word by comparing the distance with a pre-defined threshold. We must avoid as much as possible erroneous determination of OOV word because the OOV word may be retrieved as misrecognized IV word. After the classification, in only case of query classified to IV, we combine the word retrieval results and syllable-based n-gram retrieval results by using either "OR" operation or "AND" operation, where "OR" operation in Fig. 1 increases *Recall* rate and "AND" operation increases *Precision* rate. In other words, we obtained the candidates of retrieval from all hits of these retrieval or the hits voted both of them.

(b) Combination of ASR system outputs:

In Section 2.1, we described a combination of N-gram search and word-based search for IV words. For spoken queries, we further propose the following two combination methods of constructing N-grams, which utilizes multiple recognition systems. We regard all of the speech segments obtained from each system as hits, that is, "OR" operation. We search N-grams constructed by the 1-best syllable lattices for every ASR system, and combine of search results. In our experiments, we investigated two different ASR systems.

$$d_D(C_2V_2|C_1V_1\_C_3V_3)=\min\left\{\begin{array}{l} d_S(C_1V_1,C_2V_2) \\ d_S(V_1,C_2V_2) \\ d_S(C_2V_2,C_3V_3) \end{array}\right\}+\delta_D \qquad (4)$$

3. EXPERIMENTS

3.1 Evaluation Setup

We evaluate the our system on the SDPWSspeech data for search target document(♯lectures = 98) and the query set for the formal run in NTCIR11(♯query terms = 265, ♯IV terms = 252, ♯OOV terms = 13). Queries were uttered several times by different speakers(♯speakers = 37). So we searched each query and combined retrieval results. We utilized four transcripts of target document and spoken queries.

(1)"Word-based transcription by JULIUS" and (2)"Syllable-based transcription by JULIUS", which were recognized through "Match Models" provided from the organizers.

(3)"Word-based transcription by SPOJUS" and (4)"Syllable-based transcription by SPOJUS", which were recognized by SPOJUS++[23] developed in our laboratory.

In SPOJUS, the context-dependent syllable-based HMMs (928 models in total) were trained on 2707 lectures within the CSJ corpus. 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 syllable-based four grams and word-based trigrams as a language model, which was trained by the CSJ corpus excluding the core data.

The transcription of (2) is mapped SPOJUS-116 syllables from JULIUS-262 syllables, and after constructed the syllable-based n-grams. We evaluate our systems by using micro F-measure(max) and Mean Average Precision(MAP) as measures of search performance. The details of this task are shown in [24].

The syllable recognition rates and word recognition rate are summarized in Table 1. Word and syllable transcripts obtained by JURIUS is higher than ones by SPOJUS in both spoken documents and spoken queries.

Table 2: Retrieval results of text queries

(a) JULIUS

Method	F-measure	MAP
Syllable N-gram	0.276	0.222
Word	0.692	0.437
Syllable N-gram OR Word	0.692	0.498
Syllable N-gram AND Word	0.308	0.199

(b) SUPOJUS

Method	F-measure	MAP
Syllable N-gram	0.455	0.367
Word	0.626	0.373
Syllable N-gram OR Word	0.637	0.470
Syllable N-gram AND Word	0.520	0.335

Table 3: N-gram search results of spoken queries

transcript		document					
		J		S		J+S	
		F-measure	MAP	F-measure	MAP	F-measure	MAP
query	J	0.293	0.204	0.334	0.252	0.353	0.287
	S	0.200	0.134	0.365	0.284	0.378	0.290
	J + S	0.299	0.221	0.400	0.329	0.395	0.319

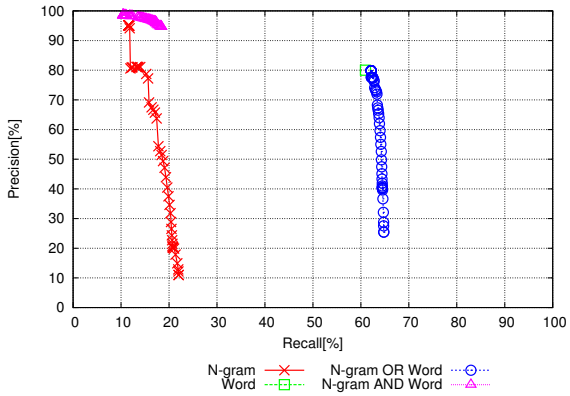
Table 4: Combination of n-gram search and word search for spoken queries

(a) Word transcript by JULIUS

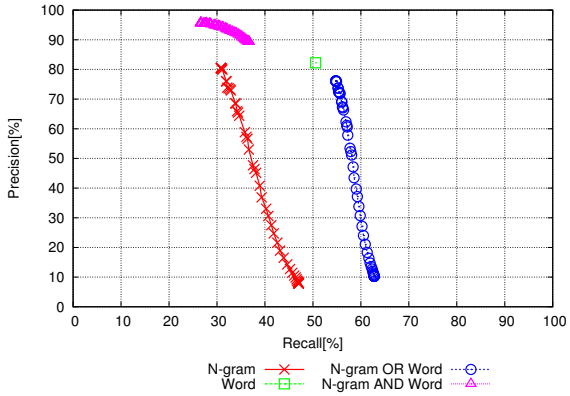
Method	F-measure	MAP
*Word	0.403	0.255
*Syllable N-gram OR Word	0.465	0.396
*Syllable N-gram AND Word	0.394	0.253
Word	0.410	0.255
Syllable N-gram OR Word	0.471	0.390
Syllable N-gram AND Word	0.395	0.244

(b) Word transcript by SPOJUS

Method	F-measure	MAP
*Word	0.513	0.304
*Syllable N-gram OR Word	0.514	0.395
*Syllable N-gram AND Word	0.501	0.302
Word	0.505	0.296
Syllable N-gram OR Word	0.519	0.392
Syllable N-gram AND Word	0.484	0.288

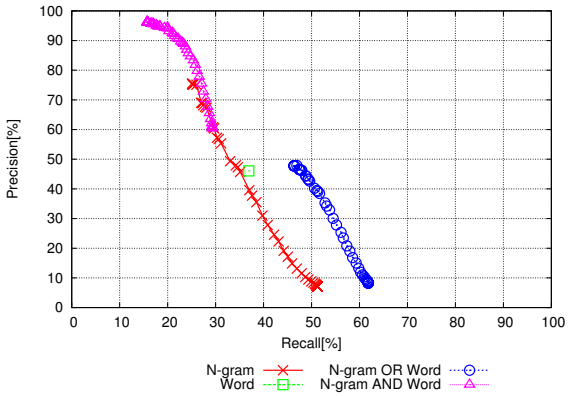


(a) JULIUS

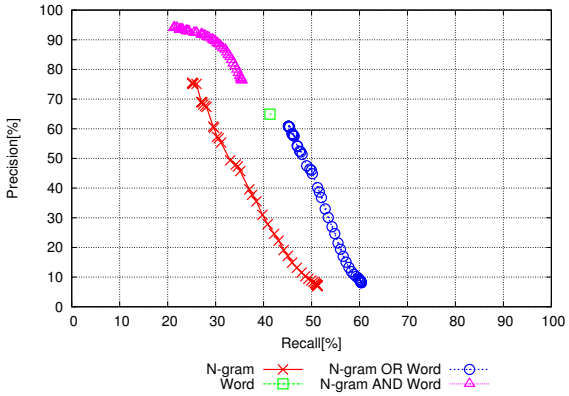


(b) SPOJUS

Figure 4: R-P Curve for text queries



(a) JULIUS



(b) SPOJUS

Figure 5: R-P Curve for spoken queries

3.2 Results

3.2.1 Text Queries

Retrieval results by a combination of word-based search and syllable based N-gram search are shown in Table 2 and Fig. 4 for text queries. In spite of the syllable recognition rate of SPOJUS is lower than the one of JULIUS, the performance of n-gram search is better. As we can see in these results, however, the only syllable N-gram system provides low performance of F-measure and MAP because recall is very low. Therefore, in the case of IV queries where we can classify IV/OOV correctly for text queries, we combine n-gram search and word search, and got the improvement on the performance in total from 0.455 of F-measure to 0.692.

3.2.2 Spoken Queries

For spoken queries, we performed the retrieval with each transcription by JULIUS or SPOJUS and both transcriptions by JULIUS and SPOJUS. Retrieval results by a combination of word search and syllable search are shown in Table 3 for spoken queries. In Table 3, “J” denotes the transcript by JULIUS and “S” denotes transcript by SPOJUS, respectively Note that the lower right in Table 3 shows the performance by combining retrieval results by the pairs of the same ASR system transcription of spoken documents and spoken queries. We got the good performance of n-gram search with the SPOJUS transcript of target document (F-measure=0.400, MAP=0.329).

Furthermore, we combined word search and syllable-based search using the document transcript by SPOJUS(the best search in Table 3), and the result is shown in Table 4 and Fig. 5. Word search was performed with the pair of the same ASR system transcription of spoken documents and spoken queries. When we integrated the IV/OOV classification approach described in Section 2.5 (a) with the combined method of word search and N-gram search, we got the best results (F=0.519, MAP=0.392) by using OR” combination. In this case, the IV classification error rate was 0.425 by JULIUS and 0.500 by SPOJUS (IV->OOV), and the OOV classification error rate was 0.280 by JULIUS and 0.475 by SPOJUS (OOV->IV), respectively. We should notice that the mis-classification of IV->OOV does not injure the syllable based N-gram search. We also show the performance of retrieval in the case of oracle (F=0.514, MAP=0.395) as shown in the column of mark ”*” in Table 4, that is, we assumed that IV/OOV classification was perfect. In surprisingly, the retrieval results with IV/OOV classification was better than the oracle case, because mis-recognizable IV words are assumed as OOV word by the classification.

4. CONCLUSION

In this paper, we described a Japanese spoken term detection method for spoken queries. We applied this method to an academic lecture presentation SDPWSSpeech database, and we achieved F-value of 0.692 and MAP of 0.498 for text-based IV queries by combining word search and syllable-

based N-gram search. For spoken queries, we proposed a combination of outputs of two ASR systems and we obtained F-value of 0.400. Finally, we achieved the best performance by a combination of word search and syllable-based search with IV/OOV classification. Finnaly, we got F-value of 0.519. In our future work, we will find a better method of IV/OOV decision and introduce contextual information into our syllable-based N-gram method to improve precision rate.

5. REFERENCES

[1] M. Larson and S. Eickeler, "Using syllable-based indexing features and language models to improve German spoken document retrieval," EuroSpeech, 2003, pp. 1217–1220.

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

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

[4] C. Allauzen, M. Mohri, and Saracla M, "General indexation of weighted automata - application to spoken utterance retrieval," Workshop on interdisciplinary approaches to speech indexing and retrieval, 2004, pp. 33–40.

[5] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," HLT/NAACL, 2004, pp. 129–136.

[6] B. Chen, H. Wang, and L. Lee, "Retrieval of broadcast news speech in Mandarin Chinese collected in Taiwan using syllable-level statistical characteristics," ICASSP, 2000, pp. 2985–2988.

[7] C. Ng, R. Wilkinson, and J. Zobel, "Experiments in spoken document retrieval using phoneme n-grams," Speech Communication, 2000, vol. 32, pp. 61–77.

[8] S. Dharanipragada and S. Roukos, "A multistage algorithm for spotting new words in speech," IEEE Transactions on Speech and Audio Processing, 2002, vol. 10, pp. 542–550.

[9] K. Katsurada, S. Teshima, and T. Nitta, "Fast keyword detection using suffix array," Interspeech, 2009, pp. 2147–2150.

[10] N. Kanda, H. Sagawa, T. Sumiyoshi, and Y. Obuchi, "Open-vocabulary keyword detection from super-large scale speech database," MMSP, 2008, pp. 939–944.

[11] H. Saito, Y.Itoh, K.Kojima, and M.Ishigame et al., "Examination of the index in method of the n-syllable sequences in advance," ASJ2013 Spring Meeting, 2013 (in Japanese).

[12] 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.

[13] K. Iwami, Y. Fujii, K. Yamamoto, and S. Nakagawa, "Efficient out-of-vocabulary term detection by n-gram array indices with distance from a syllable lattice," ICASSP 2011, 2011, pp. 5664–5667.

[14] Marijn Huijbregts, Mitchell McLaren, and David van Leeuwen, "Unspervised acoustic sub-word unit detection for query-by-example spoken term detection," ICASSP, 2011, pp. 4436–4439.

[15] Haipeng Wang, Cheung-Chi Leung, Tan Lee, Bin Ma, and Haizhou Li, "An acoustic segment modeling approach to query-by-example spoken term detection," ICASSP, 2012, pp. 5157–5160.

[16] Alberto Abad, Luis Javier Rodriguez-Fuentes, Mikel Penagarikano, Amporo Varona, and German Bordel, "On the calibration and fusion of heterogeneous spoken term detection systems," INTERSPEECH, 2013, pp. 20–24.

[17] Cheng-Tao Chung, Chun an Chan, and Lin shan Lee, "Unsuperised spoken term detection with spoken queries by multi-level acoustic patterms with varying model granularity," ICASSP, 2014, pp. 7864–7868.

[18] Chun-An Chan and Lin-Shan Lee, "Unspervised hidden markov modeling of spoken queries for spoken term detection without speech recognition," Interspeech, pp. 2141–2144.

[19] Mitsuaki Makino, Naoki Yamamoto, and Atsuhiko Kai, "Utilizing state-level distance vector representation for improved spoken term detection by text and spoken queries," Interspeech, pp. 1732–1736.

[20] S. Nakagawa, K. Imami, Y. Fujii, and K. Yamamoto, "A robust/fast spoken term detection method based on a syllable n-gram index with a distance metric," Speech Communication 2012, 2012.

[21] N. Sakamoto and S. Nakagawa, "Robust/fast out-of-vocabulary spoken term detection by n-gram index with exact distance through text/speech input," APSIPA, 2013.

[22] N. Sakamoto and S. Nakagawa, "Spoken term detection method by using multiple recognition results of spoken query," Spoken Document Processing Workshop, 2014.

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

[24] Tomoyosi Akiba, Hiromitsu Nishizaki, Hitoaki Nanjo, and Gareth J. F. Jones, "Overview of the ntcir-11 spokenquery&doc task," In Proceedings of the NTCIR-11 Conference, Tokyo, Japan, 2014.