Unsupervised Pattern Discovery from Thematic Speech Archives Based on Multilingual Bottleneck Features

Man-Ling Sung, Siyuan Feng and Tan Lee
Department of Electronic Engineering,
The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China
E-mail: {mlsung, syfeng, tanlee}@ee.cuhk.edu.hk

***Abstract*—The present study tackles the problem of automatically discovering spoken keywords from untranscribed audio archives without requiring word-by-word speech transcription by automatic speech recognition (ASR) technology. The problem is of practical significance in many applications of speech analytics, including those concerning low-resource languages, and large amount of multilingual and multi-genre data. We propose a two-stage approach, which comprises unsupervised acoustic modeling and decoding, followed by pattern mining in acoustic unit sequences. The whole process starts by deriving and modeling a set of subword-level speech units with untranscribed data. With the unsupervisedly trained acoustic models, a given audio archive is represented by a pseudo transcription, from which spoken keywords can be discovered by string mining algorithms. For unsupervised acoustic modeling, a deep neural network trained by multilingual speech corpora is used to generate speech segmentation and compute bottleneck features for segment clustering. Experimental results show that the proposed system is able to effectively extract topic-related words and phrases from the lecture recordings on MIT OpenCourseWare.**

***Index Terms*—Zero-resource speech technology, unsupervised speech modeling, acoustic segment model, string mining**

I. Introduction

In recent years, automatic speech recognition (ASR) technology is advancing to the level that is considered adequate for daily use in human-computer interaction [1], [2]. The high performance level is contributed largely by the effectiveness of deep learning algorithms with large amount of training data [3], [4]. For mainstream commercial systems, there is no much concern on the availability of training data and linguistic knowledge about the intended languages. With fine-grained or coarse transcription for part or all of the training data, supervised approaches could be applied to the training of a deep neural network for acoustic-phonetic mapping [3], [5], [6]. This is considered a rather straightforward process.

There are many application scenarios where transcribed training data are difficult or even impossible to acquire. One of the scenarios concerns those unpopular and less-studied languages or dialects, for examples, ethnic minority languages in China. In the ASR research community, low-resource languages (or zero-resource in the extreme case) refer to the cases that most of the key elements of linguistic knowledge required for ASR system development, e.g., definition of phonemes, pronunciation lexicon, orthographically represented data, etc., do not exist, although a certain amount of audio-form speech data may be available [7], [8]. There are also situations that the linguistic resources in existence do not adequately represent the spoken language in actual usage, for examples, regional variants of a major language, code-mixing speech, and technical language in a highly specialised area.

Another application scenario with practical significance is the unsupervised spoken term discovery from large-scale multi-genre audio archives. By multi-genre, we refer to a high diversity of audio content, including speech of different speaking styles and accents, and from varying acoustic channels, and all kinds of non-speech sounds. Examples are publicly available broadcast media content, live recordings of lectures/seminars/meetings, and personal digital recordings. While improving ASR performance for multi-genre speech transcription has attracted great research interest [9], unsupervised pattern discovery is considered a pragmatic approach to efficient indexing, search and organizing of complex audio content.

The present study tackles the problem of unsupervised keyword discovery from raw audio of topic-specific classroom lectures. The experimental dataset is adapted from unedited video recordings in the MIT OpenCourseWare, covering various courses on Mathematics, Engineering and Computer Programming [10]. In addition to the teacher's speech, the audio content contain students' speech (e.g., asking or responding questions), cough, laughter, chalk-writing sounds, furniture sounds, video sound, etc. The teachers may or may not be native speakers of English. Some of them carry very strong foreign accents. Our objective is to automatically find out a set of keywords that can represent the subject of a lecture or course, without performing explicit speech transcription.

To tackle this problem, we develop a two-stage approach, which comprises unsupervised acoustic modeling and decoding, followed by pattern mining in acoustic unit sequences. Subword-level acoustic models are trained from untranscribed audio recordings using an unsupervised approach. For the segmentation of audio frames and clustering of subword units, we propose to incorporate language-independent bottleneck features (BNFs) from a deep neural network (DNN) trained

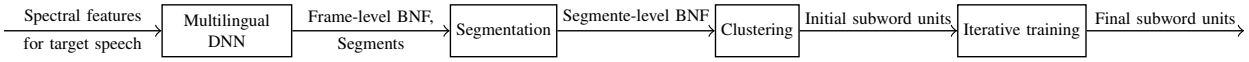

Fig. 1: Unsupervised acoustic modeling framework used to train acoustic subword models and generate subword unit sequences for target speech.

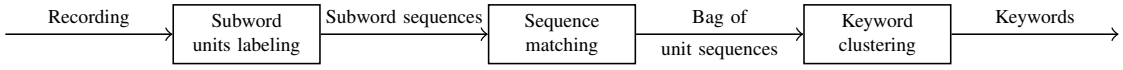

Fig. 2: Unsupervised pattern mining framework used to discover keywords in subword unit sequences of target speech.

by multilingual speech corpora [11]. The multilingual DNN is expected to provide a wide coverage and rich representation of acoustic and phonetic variations, so as to better characterize the unseen speech data. With unsupervisedly trained acoustic models, each lecture recording can be decoded into a pseudo transcription, which is in the form of subword unit sequence. A spoken keyword in the lecture is identified as a distinctive subsequence pattern that occurs multiple times in the pseudo transcription. The performance of the proposed system is evaluated by comparing and relating the discovered "keywords" with the ground-truth transcription provided at the MIT OpenCourseWare website.

II. Previous work on unsupervised acoustic modeling and pattern discovery

A. Unsupervised acoustic modeling

Unsupervised acoustic modeling aims at finding basic speech units, which are desirably equivalent to phonemes, from untranscribed speech data. Previously investigated approaches can be divided into two categories, namely *bottom-up modeling* and *top-down modeling*. In the bottom-up approach, speech is viewed as a sequence of low-level components, e.g., frames or segments, which can be grouped by clustering techniques to define higher-level structures [12]–[14]. The learned clusters are regarded as the basic units to represent the language concerned. Top-down modeling methods seek use of higher-level knowledge to provide constraints and guide the modeling of low-level speech components [15], [16]. The higher-level knowledge, e.g., word/phrase segment pairs, could be obtained by a process known as spoken term discovery (STD). In recent studies, DNNs has been shown effective in improving performance of both bottom-up and top-down modeling methods [17]–[20].

The commonly used bottom-up framework for unsupervised acoustic modeling consists of three steps, namely, speech segmentation, segment clustering, and iterative modeling. Speech segmentation aims to divide a speech utterance into variable-length segments based on, for instance, the spectral discontinuities. Segmentation could be data-driven, e.g. the bottom-up hierarchical clustering method presented in [21], or based on out-of-domain knowledge and/or resources, e.g., language-mismatched phone recognizers [22]. Segment clustering is to find and group speech segments that share similar acoustic properties. Various clustering algorithms such as Gaussian mixture model (GMM) [23], segmental GMM (SGMM) [24], vector quantization (VQ) [25] and spectral clustering [13], [22], have been investigated. With the cluster labels as initial tokenization, acoustic models are trained and refined by iteratively performing model parameter estimation and decoding, in a supervised manner. Following the terminology of [25], this framework is referred to as acoustic segment modeling (ASM) in this study.

B. Unsupervised pattern discovery

Unsupervised pattern discovery (also known as spoken term discovery (STD) [8], or unsupervised term discovery (UTD) [26]) refers to the task of automatically discovering words and linguistic entities from audio archives without supervision [27]. Unsupervised pattern discovery from audio signals could be done in two steps [27], [28]: (1) identifying similar audio segments from target audio archives, and (2) clustering the segments into groups of discovered patterns. In [27], Park and Glass proposed a segmental dynamic time wrapping (DTW) algorithm to discover similar audio patterns from pairs of utterances by comparing at acoustic level. They further applied graph-based clustering towards similar segments, in order to find the most common word-/phrase-like patterns. A number of extensions to [27] were made mainly to improve the feature representations, e.g., using Gaussian posteriorgram [28], language-mismatched phoneme posteriorgram and unsupervised subword posteriorgram [29]. Another direction of improvement was to improve the computation efficiency [30].

Recently, there are works focusing on the topic of *lexicon discovery* from untranscribed speech [31], [32]. It is closely related to unsupervised pattern discovery, but emphasizing on full-coverage segmentation of speech into word-like units.

III. Unsupervised acoustic modeling and decoding

In the present study, the ASM framework is adopted for unsupervised acoustic modeling and decoding. The novelty of our work lies in that a multilingual DNN trained with resource-rich language resources is involved in both speech segmentation and segment representation. The proposed system design is shown as in Fig. 1.

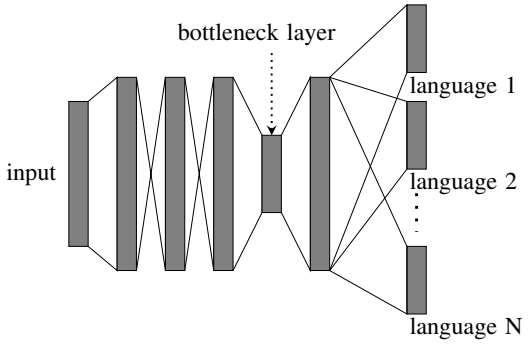

Fig. 3: Multilingual bottleneck network used to obtain language independent acoustic information.

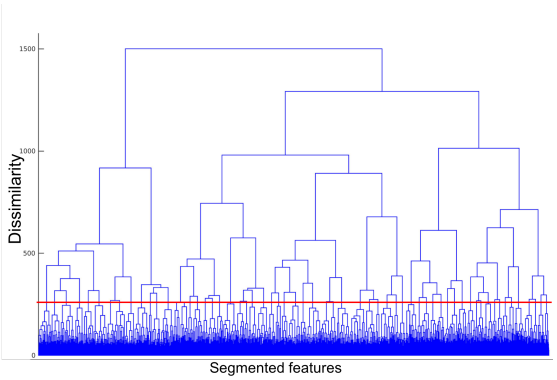

Fig. 4: Dendrogram showing hierarchical clustering of 100,000 segments.

A. Segmentation and BNF extraction

The input utterance is divided into variable-length segments, from which fixed-dimension feature representations are extracted. In the proposed system, a multilingual DNN is trained for this purpose. The architecture of a multilingual DNN is illustrated as in Figure 3. It contains a set of language-specific output layers and a number of shared hidden layers, including a low-dimensional bottleneck (BN) layer. Training of the DNN follows stochastic gradient descent (SGD) algorithm, to minimize language-weighted average cross-entropy loss function [33].

Given an input utterance, the trained DNN would produce multiple sets of phoneme-level time alignments at the block-softmax layers. These hypothesized alignments are combined to give a single segmentation as in described in Feng et al. [22], i.e., hypothesized segment boundaries within an internal of 20 ms are merged. Frame-level BNF of an input utterance are extracted from the bottleneck layer. Segment-level BNFs are obtained by averaging the frame-level features.

B. Segment clustering

The BNFs obtained from all segments have the same dimension. By a BNF clustering process, segments with similar acoustic properties are grouped together. A practical issue for segment clustering is that the number of clusters is unknown, as in most cases linguistic knowledge on the concerned language is completely unavailable. The hierarchical agglomerative clustering (HAC) [34] approach is applied to obtain a general picture on the similarity among segments, and allow the number of clusters to be determined manually. In this study, the Ward's minimum variance [35] is used as the linkage criterion for HAC. An example dendrogram describing the result of HAC on a lecture recording from MITOpenCourseWare (refer to Section V for details) is illustrated as in Fig. 4.

After determining the number of clusters R, the segments from the entire dataset are clustered by the k-means algorithm. Each of the resulted cluster is assigned a specific label that is used to represent all its members. The clusters are expected to correspond to a set of linguistically relevant speech units are subword level. Given an input utterance, the sequence of segment labels is regarded as a kind of *pseudo transcriptions*, which can be used for further analysis and comparison.

C. Iterative training and decoding

The availability of pseudo transcription makes supervised training of acoustic models possible. Specifically, DNN-HMM acoustic models [3] can be trained to represent the unsupervisedly learned subword units, and these models are refined in conjunction with the pseudo transcriptions of all training data in an iterative manner, as elaborate below:

1) Train an initial set of DNN-HMM acoustic models with the pseudo transcriptions resulted from segment clustering.
2) Decode the training data with the current models and obtain new pseudo transcriptions.
3) Train the acoustic models with the new pseudo transcriptions.
4) Repeat Step 2) and 3) until convergence.

Iterative training is carried out with the domain-specific dataset, i.e., all lectures from the same course. In this way, acoustic models and pseudo transcriptions are jointly optimized for the target application.

After iterative training, the final version of pseudo transcriptions, in the form of subword unit sequences, are used for the subsequent process of keyword discovery.

IV. Pattern Discovery from Subword Unit Sequence

With the unsupervisedly trained acoustic models, the audio content of a lecture can be represented by a pseudo transcription (subword unit sequence). A keyword spoken frequently during the lecture is identified by a distinctive pattern that repeatedly occurs in the transcription. The process of searching for such candidate patterns consists of two parts:

- Sequence matching between each pair of pseudo transcriptions and generating "bags of unit sequences";
- Clustering the "bags of unit sequences" into groups of similar sequence patterns, which correspond to keywords.

A. Generating "bags of unit sequences"

Identification of similar or closely related subsequences in long symbol sequences is an important problem in bioinformatics. In [36], an algorithm of inexact matching between a pair of short symbol sequences was described. Based on this algorithm, the "bags of unit sequences" containing only the matching subsequences are generated. This algorithm is referred to as *local sequence alignment* in this paper. Pseudo code for this algorithm is listed in Algorithm 1.

In our application, symbol sequences are pseudo transcriptions, i.e. subword unit sequences, acquired from unsupervised acoustic modeling, as discussed in Section III. The motivation for allowing inexact matching of "bags of unit sequences" is to cope with the possible decoding errors in pseudo transcriptions and pronunciation variations. Suppose there are N utterances for the target untranscribed speech. Each time we pick up 2 symbol sequences A and B, corresponding to 2 utterances, and perform Algorithm 1 towards A and B to generate "bags of unit sequences". After $\frac{N(N-1)}{2}$ times, all possible pairs of sequences are used to perform Algorithm 1. Finally, all the discovered "bags of unit sequences" are stored, and will be used for subsequent clustering process. We are only interested in the "bags of unit sequences" with 4 symbols or longer, to make sure they mainly cover content-related words.

Algorithm 1 Local sequence alignment

1: **procedure** LOCALALIGN($A = a_1a_2...a_n, B = b_1b_2...b_m$)
2: $s(a_i, b_j) = \begin{cases} +1 & \text{, if } a_i, b_j \text{ match} \\ -1 & \text{, if } a_i, b_j \text{ mismatch} \end{cases}$ ▷ Similarity score between sequence elements a_i and b_j
3: Compute an $(n+1) \times (m+1)$ matrix $\mathbf{P}$, where the element $p_{i,j}$ is,

$$p_{i,j} = \begin{cases} 0 & \text{, } i \text{ or } j = 0 \\ \max \begin{cases} p_{i-1,j-1} + s(a_i, b_j) \\ p_{i-1,j} \\ p_{i,j-1} \\ 0 \end{cases} & \text{, elsewhere} \end{cases}$$

4: Traceback from p_{n,j^*} ending with an element of $\mathbf{P}$ equal to 0, where p_{n,j^*} are local maxima of $\{p_{n,j} | 0 \leq j \leq m\}$, to obtain common subsequence pairs in A and B.
5: Store all obtained subsequences with reasonable length into the "bag of words".
6: **end procedure**

B. Sequence clustering

The "bag of unit sequences" created as in Section IV-A contain a large number of unit sequences of different lengths. These sequences are clustered into groups using the leader clustering algorithm [37], as depicted in Algorithm 2. The radius of each cluster is given as T. To avoid clusters from overlapping to much, we set the minimum distance between centroids into $a*T$, where a is larger than 0. Leader clustering is sensitive to the initialization of centroids. To avoid having the centroid that is not the representative of the cluster due to poor initialization (e.g. outliers), the centroid is updated with the most representative sequence, i.e., the sequence having the least total distance with all cluster members, as is shown in Line 10 of Algorithm 2. The clustering process iterates until the number of clusters converges (remain unchanged).

The normalized Levenshtein distance $||L(x,y)||$ used in Algorithm 2 is defined as,

$$||L(x,y)|| = b\frac{L(x,y)}{\sqrt{|x|^2 + |y|^2}}, \quad (1)$$

where $L(x,y)$ is the Levenshtein distance [38] between sequences x and y. b is an adjustable scalar. The normalization allows clusters to be formed with consideration of sequence length: short sequences to have more strict match (e.g. 1 difference out of 5), and longer sequence to have looser match (2-3 differences out of 10).

Algorithm 2 Leader clustering

1: **procedure** LEADER(bag of sequences S)
2: Initial a point i to centroid
3: **for** each point p in S **do**
4: **if** $||L(i,p)|| > a * T$ for all i in centroid, $a > 0$: **then**
5: Add p to centroid
6: **end if**
7: **end for**
8: Assign each point p in S to its closest cluster i with $||L(i,p)|| < T$.
9: **for** each group **do**
10: Update centroid with the representative of the cluster (measured by smallest total distance with same group members).
11: **end for**
12: Repeat steps 2-11 until the number of clusters converge.
13: **end procedure**

V. EXPERIMENTAL SETUP

A. Training data

The proposed system is evaluated on the task of unsupervised keyword discovery from audio recordings of academic lectures. The audio data are extracted from three courses that are publicly available at the MIT OpenCourseWare website [10]. The courses are "Mathematics for Computer Science" (MATH), "Principles of Digital Communication II" (COMM) and "Introduction to Computer Science and Programming in Python" (PYTH). The recording conditions of the three courses were similar. Each course consists of $12-25$ lectures. The duration of lecture is in the range of $45-70$ minutes. The course teacher of MATH spoke with French accent. The speaking rate in PYTH was relatively fast and that in COMM was slow. The long recordings are divided into short segments (roughly $5-10$ seconds) for further processing.

B. System set-up

Unsupervised acoustic modeling is carried out for the three courses separately. That is, a set of subword units are learned from all lecture recordings of each course, leading to a set of acoustic models tailored for the course.

A multilingual DNN with BN layer is trained with 5 speech corpora of 4 resource-rich languages: TIMIT (English) [39], WSJ (English) [40], CUSENT (Cantonese) [41], 863 (Mandarin) [42] and a distant-speech database of German [43], using Kaldi [44]. The DNN has 5 hidden layers with dimensions of 1500, 1500, 1500, 40 (BN layer) and 1500 respectively. There are 5 block-softmax output layers, corresponding to the 5 speech corpora. The state-level labels required for supervised training of the multilingual DNN are obtained from the context-dependent GMM-HMM (CD-GMM-HMM) trained separately for the 5 corpora. 23-dimensional Mel-scale filter-bank features (fbanks) are extracted for training both CD-GMM-HMM and DNN.

The trained multilingual DNN is used to obtain subword-level segmentation and generate frame-level BNFs for the lecture recordings (see Section III-A). The segmentation is derived from the 5 block-softmax layers. Segment-level BNFs are obtained by averaging the frame-level features. By applying k-means clustering to segment-level features, 55 clusters are obtained, each denoting a subword-level unit. The number of clusters is determined by HAC, as discussed in Section III-B.

For iterative training, acoustic models are trained in a course-specific manner, also using Kaldi [44][1]. In each iteration, GMM-HMM acoustic models with speaker adaptive training (GMM-HMM-SAT) are trained beforehand to generate feature-space maximum likelihood linear regression (fMLLR) features, with the latest version of pseudo transcriptions and multilingual BNFs as input features. The fMLLRs and HMM state-level alignments generated by GMM-HMM-SAT are used to train the DNN-HMM acoustic models, followed by decoding target speech to obtain an updated version of pseudo transcriptions, i.e., the 1-best path of decoding lattices. The language model needed for decoding is trained with exact the same version of transcriptions used for GMM-HMM-SAT training in this iteration. During iterative training of course-specific acoustic models and decoding target data into pseudo transcriptions, the frame-level unit labels before and after each iteration are compared. If the percentage difference is below 0.1%, convergence of training is assumed. In our experiments, it is observed that 5 to 6 iterations are needed for convergence.

As a comparison, iterative training with the same configuration except for replacing multilingual BNFs with fbanks (as inputs to GMM-HMM-SAT training) is also tested. The system with multilingual BNFs converges faster than that with the fbanks[2].

After iterative training, the recordings are represented by pseudo transcriptions. Bag of subword sequences, generated by performing Algorithm 1, are clustered by Algorithm 2 in order to obtain similar sequence patterns, i.e. the discovered keywords. The parameter b in Equation (1) is set to 4. We experimented various parameters from $T : 0.7 - 1.6$ (with interval 0.1), $a : 1 - 2$ (with interval 0.1), and found out that $T = 1.4$, $a = 1.8$ could lead to relatively good performance in terms of cluster number, purity and cluster size.

VI. Results and Analysis

Given a set of lecture recordings, the proposed system is able to generate a certain number of clusters of subword unit sequences. Unit sequences in the same cluster should be similar (based on the normalized Levenshtein distance) and are expected to represent the same word or phrase being spoken in the lectures. For the intended task of keywords discovery, it is desired that a significant portion of the content-related words could be covered by the unsupervisedly generated clusters. In this section, we analyze the automatically discovered keyword clusters with respect to the frequently occurred words and phrases in the ground-truth transcriptions (available at the MIT OpenCourseWare website[3]).

A. Quality of discovered "word" clusters

We examine the clustering results for 2 selected lectures in the course MATH. For Lecture 4 ("Number Theory I", 80 min. long), the system generates 95 keyword clusters from 34,313 candidate unit sequences. For Lecture 8 ("Graph Theory II: Minimum Spanning Trees", 83 min. long), there are 119 clusters from 25,899.

Tables I and II list the words corresponding to the 10 longest sequence clusters in the two lectures, respectively. It is observed that most of these clusters correspond to words that are related to the lecture topic. Clusters of sequences with 12 to 16 subword units generally have high purity. A cluster with high purity provides a valid representation of a specific word or phrase. As the sequence length decreases, the cluster's purity tends to decrease. Sequences with less than 5 units typically correspond to parts of different words that have similar pronunciations, e.g., "so", "(al)so", and "so(lve)"; "in" and "in(teger)".

The same word may be represented by more than one clusters. For example, clusters #116 and #109 for Lecture 8 (Table II) both correspond to "vertices". It is also observed that some of discovered words can be composed by other clusters of shorter sequences. Some of the clusters represent non-speech sounds, e.g., cluster #77 in Table I for "chalk-writing sounds", which are very common in live recordings of lectures.

B. Coverage of discovered words

In this section, the coverage of automatically discovered "words" is examined with respect to the ground-truth transcriptions. For the three courses that we are experimenting with, word-by-word speech transcriptions are available at the

[1] kaldi/egs/timit/s5/run.sh

[2] Due to limited space, experimental results on comparing BNFs and fbanks are not presented here.

[3] ocw.mit.edu/index.htm

TABLE I: Lecture 4: Number Theory I

Cluster #	Corresponding words	Cluster size	Purity
58	divide any result	2	100%
63	the zero steps	2	100%
70	Multiple of	2	100%
56	linear combination	17	100%
40	the/a number theory	5	100%
70	a and b	14	100%
28	use the lemma again	2	100%
43	Greatest common, The greatest common	18	100%
77	*chalk-writing sounds*	8	100%
76	gallon jug	15	100%

TABLE II: Lecture 8: Graph Theory II: Minimum Spanning Trees

Cluster #	Corresponding words	Cluster size	Purity
70	b equal to	2	100%
66	this particular edge, this particular err	4	75%
57	connected subgraph, the subgraph	7	100%
89	A subgraph,The smaller part	4	50%, 50%
87	still connected, both connected	4	100%
83	Double star is	2	100%
80	So we know that	2	100%
116	Vertices, vertices that, -ices have	7	85.7%
45	the spanning tree, a spanning tree, spanning tree	25	100%
109	vertices	4	100%

the course website. Word-level trigrams, bigrams and unigrams are computed from the transcription for each lecture session or all lectures in a course, with the function words like "is", "a", "the" being discarded.

For a specific lecture session, the most frequent N-grams are examined one by one, to determine whether the corresponding word(s) can be matched with any of the discovered "word" clusters. There are cases that a cluster may partially match a trigram or bigram. If the unmatched part is a function word, e.g. "linear combination" versus "linear combination of", it is regarded as a case of match. If the unmatched part is a content word, e.g., "divisor" versus "common divisor", it is regarded as mismatch.

For Lecture 4 ("Number Theory I") of the course MATH and Lecture 8 ("Object-Oriented Programming") of the course PYTH, we analyze the top 10 trigrams, top 20 bigrams and 30 unigrams and match them with the 10 longest sequence clusters and other highly-populated clusters. The matching rates are found to be 73.3% and 51.6% respectively. The details of matching results for the lecture of "Object-Oriented Programming" are given as in Table III. It is noted that the uncovered unigrams are mostly words with a small number of phones, e.g. "add", "car", "code", while the covered words are mostly polysyllabic words, e.g., "python", "coordinate".

TABLE III: Matching results for the lecture of "Object-Oriented Programming"

Trigram	Count	Match ?
a coordinate object	14	yes
can interact with	13	yes
a fraction object	9	partly
an object of	7	yes
of the class	7	yes
of type coordinate	7	partly
the exact same	6	
you can create	6	
is equal to	6	
going to define	6	yes

(a) Trigram

Bigram	Count	Match ?
an object	25	yes
the class	22	yes
coordinate object	20	partly
interact with	19	yes
a list	17	
a coordinate	17	yes
the object	14	yes
fraction object	14	partly
object that	14	yes
the x	13	
this method	13	yes
can interact	13	yes
create a	13	
a fraction	12	yes
data attributes	12	yes
of type	11	yes
for example	11	yes
to define	10	
the list	10	
underscore underscore	10	

(b) Bigrams

Unigram	Count	Match ?
object	118	yes
coordinate	70	yes
class	60	yes
type	47	yes
method	46	yes
data	45	yes
objects	44	yes
right	38	
i	35	
list	32	
python	30	yes
x	29	
create	28	
self	28	
print	27	
one	25	yes
c	24	yes
dot	24	
attributes	22	yes
fraction	21	yes
underscore	20	
value	20	yes
interact	19	yes
y	18	
define	18	
init	17	
add	17	
lists	16	
car	16	
code	15	

(c) Unigrams

For the lecture of "Number Theory I", the matching rate for unigrams is higher, due to more complicated phonetic structure of the words.

The same analysis has also been done for the course COMM, which contains 25 lectures of 70 minutes long. The 100 most frequent trigrams, bigrams and unigrams are examined by comparing with clusters generated from all lectures. A high matching rate of 85.8% is recorded.

The proposed system needs improvement in its ability of identifying keywords of relatively short length. In fact, short candidate sequences are not included when generating the candidate sequences for clustering (see Section IV-A). There is also an issue related to clusters that represent parts of a word.

VII. Conclusion

We propose a two-stage approach to unsupervised keyword discovery. Pseudo transcription are generated by decoding results of unsupervised subword models. Keywords are tokenized by searching matching subword sequences in pseudo transcription using local sequence alignment, and clustered into groups with leader clustering.

We experimented the model on 3 academic courses, concluding that the model has the ability in extracting topic related information, with coverage of $51.6\% - 85.8\%$ of the most frequent keywords. The performance of keywords matching is related to the phone size of the word, and is easier to match words with more phonemes with high purity in the clusters.

It is expected that the model has the capability in modeling other languages (e.g. zero-resource languages) with the use of language transfer property in multilingual bottleneck network. Future application on topic classification of recordings can also be considered.

Acknowledgment

This research is partially supported by a GRF project grant (Ref: CUHK 14227216) from the Hong Kong Research Grants Council.

References

[1] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *Proc. INTERSPEECH*, 2017, pp. 132–136.

[2] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," in *Proc. INTERSPEECH*, 2017, pp. 949–953.

[3] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *TASLP*, vol. 20, no. 1, pp. 30–42, 2012.

[4] X. Chen, A. Ragni, X. Liu, and M. J. Gales, "Investigating bidirectional recurrent neural network language models for speech recognition," in *Proc. INTERSPEECH*, 2017, pp. 269–273.

[5] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. INTERSPEECH*, 2014.

[6] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. ICASSP*, 2016, pp. 4945–4949.

[7] M. Versteegh, R. Thiolliere, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," in *Proc. INTERSPEECH*, 2015.

[8] E. Dunbar, X. Cao, J. Benjumea, J. Karadayi, M. Bernard, L. Besacier, X. Anguera, and E. Dupoux, "The zero resource speech challenge 2017," in *Proc. ASRU*, 2017.

[9] A. Ali, S. Vogel, and S. Renals, "Speech recognition challenge in the wild: Arabic MGB-3," in *Proc. ASRU*, 2017.

[10] J. Durphy, "Massachusetts Institute of Technology: MIT OpenCourseWare," online: "http://www.ocw.mit.edu", License: Creative Commons BY-NC-SA.

[11] K. Veselỳ, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proc. SLT*, 2012, pp. 336–341.

[12] Y. Zhang and J. R. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *Proc. ICASSP*, 2010, pp. 4366–4369.

[13] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *TASLP*, vol. 23, no. 2, pp. 264–277, 2015.

[14] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Parallel inference of Dirichlet process Gaussian mixture models for unsupervised acoustic modeling: A feasibility study," in *Proc. INTERSPEECH*, 2015, pp. 3189–3193.

[15] A. Jansen and K. Church, "Towards unsupervised training of speaker independent acoustic models," in *Proc. INTERSPEECH*, 2011, pp. 1693–1696.

[16] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training." in *Proc. ICASSP*, 2013, pp. 8091–8095.

[17] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. ICASSP*, 2014, pp. 7634–7638.

[18] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. ICASSP*, 2015, pp. 5818–5822.

[19] D. Renshaw, H. Kamper, A. Jansen, and S. Goldwater, "A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge," in *Proc. INTERSPEECH*, 2015, pp. 3199–3203.

[20] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Multilingual bottleneck feature learning from untranscribed speech," in *Proc. ASRU*, 2017, pp. 727–733.

[21] Y. Qiao, N. Shimomura, and N. Minematsu, "Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons," in *Proc. ICASSP*, 2008, pp. 3989–3992.

[22] S. Feng, T. Lee, and H. Wang, "Exploiting language-mismatched phoneme recognizers for unsupervised acoustic modeling," in *Proc. ISCSLP*, 2016, pp. 1–5.

[23] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. ICASSP*, 2012, pp. 5157–5160.

[24] H. Gish and K. Ng, "A segmental speech model with applications to word spotting," in *Proc. ICASSP*, vol. 2, 1993, pp. 447–450.

[25] C.-H. Lee, F. K. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Proc. ICASSP*, 1988, pp. 501–504.

[26] V. Lyzinski, G. Sell, and A. Jansen, "An evaluation of graph clustering methods for unsupervised term discovery," in *Proc INTERSPEECH*, 2015.

[27] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *TASLP*, vol. 16, no. 1, pp. 186–197, 2008.

[28] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU*, 2009, pp. 398–403.

[29] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. ICASSP*, 2012, pp. 5157–5160.

[30] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. ASRU*, 2011, pp. 401–406.

[31] C.-y. Lee, T. J. O'Donnell, and J. Glass, "Unsupervised lexicon discovery from acoustic input," *Trans. ACL*, vol. 3, pp. 389–403, 2015.

[32] H. Kamper, A. Jansen, and S. Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *CSL*, vol. 46, pp. 154–174, 2017.

[33] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*, 2013, pp. 7304–7308.

[34] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.
[35] F. Murtagh and P. Legendre, "Ward's hierarchical clustering method: clustering criterion and agglomerative algorithm," *arXiv preprint arXiv:1111.6285*, 2011.
[36] T. Smith and M. Waterman, "Identification of common molecular subsequences." *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.
[37] J. A. Hartigan, "Clustering algorithms," 1975.
[38] V. Pieterse and P. Black, "Levenshtein distance," *Dictionary of Algorithms & Data Structures*, 2015.
[39] L. F. L. John S. Garofolo *et al.*, "TIMIT acoustic-phonetic continuous speech corpus," 1993, Philadelphia: Linguistic Data Consortium.
[40] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Speech and Natural Language*, 1992, pp. 357–362.
[41] T. Lee, W. K. Lo, P. Ching, and H. Meng, "Spoken language resources for cantonese speech processing," *Speech Communication*, vol. 36, no. 3-4, pp. 327–342, 2002.
[42] Y. Qian, Y. Liu, H. Liu, and Q. Liu, "An introduction to corpora resources of 863 program for chinese language processing and human-machine interaction," *Proc. ALR2004, affiliated to IJCNLP*, 2004.
[43] S. Radeck-Arneth, B. Milde, A. Lange, E. Gouvêa, S. Radomski, M. Mühlhäuser, and C. Biemann, "Open source german distant speech recognition: Corpus and acoustic model," in *International Conference on Text, Speech, and Dialogue*. Springer, 2015, pp. 480–488.
[44] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.