SEGMENTAL AUDIO WORD2VEC: REPRESENTING UTTERANCES AS SEQUENCES OF VECTORS WITH APPLICATIONS IN SPOKEN TERM DETECTION

Yu-Hsuan Wang, Hung-yi Lee, Lin-shan Lee

College of Electrical Engineering and Computer Science
National Taiwan University

{r04922167, hungyilee}@ntu.edu.tw, lslee@gate.sinica.edu.tw

ABSTRACT

While Word2Vec represents words (in text) as vectors carrying semantic information, audio Word2Vec was shown to be able to represent signal segments of spoken words as vectors carrying phonetic structure information. Audio Word2Vec can be trained in an unsupervised way from an unlabeled corpus, except the word boundaries are needed. In this paper, we extend audio Word2Vec from word-level to utterance-level by proposing a new segmental audio Word2Vec, in which unsupervised spoken word boundary segmentation and audio Word2Vec are jointly learned and mutually enhanced, so an utterance can be directly represented as a sequence of vectors carrying phonetic structure information. This is achieved by a segmental sequence-to-sequence autoencoder (SSAE), in which a segmentation gate trained with reinforcement learning is inserted in the encoder. Experiments on English, Czech, French and German show very good performance in both unsupervised spoken word segmentation and spoken term detection applications (significantly better than frame-based DTW).

Index Terms— recurrent neural network, autoencoder, reinforcement learning, policy gradient

1. INTRODUCTION

In natural language processing, it is well known that Word2Vec transforming words (in text) into vectors of fixed dimensionality is very useful in various applications, because those vectors carry semantic information[1][2]. In speech signal processing, it has been shown that audio Word2Vec transforming spoken words into vectors of fixed dimensionality[3][4] is also useful for example in spoken term detection or data augmentation[5][6], because those vectors carry phonetic structure for the spoken words. It has been shown that this audio Word2Vec can be trained in a completely unsupervised way from an unlabeled dataset, except the spoken word boundaries are needed. The need for spoken word boundaries is a major limitation for audio Word2Vec, because word boundaries are usually not available for given speech utterances or corpora[7][8].

Although it is possible to use some automatic processes to estimate word boundaries followed by the audio Word2Vec[9][10][11][12][13], it is highly desired that the signal segmentation and audio Word2Vec may be integrated and jointly learned, because in that way they may enhance each other. This means the machine learns to segment the utterances into a sequence of spoken words, and transform these spoken words into a sequence of vectors at the same time. This is the segmental audio Word2Vec proposed here: representing each utterance as a sequence of fixed-dimensional vectors, each of which hopefully carries the phonetic structure information for a spoken word. This actually extends the audio Word2Vec from word-level up to utterance-level. Such segmental audio Word2Vec can have plenty of potential applications in the future, for example, speech information summarization, speech-to-speech translation or voice conversion[14]. Here we show the very attractive first application in spoken term detection.

The segmental audio Word2Vec proposed in this paper is based on a *segmental sequence-to-sequence autoencoder* (SSAE) for learning a segmentation gate and a sequence-to-sequence autoencoder jointly. The former determines the word boundaries in the utterance, and the latter represents each audio segment with an embedding vector. These two processes can be jointly learned from an unlabeled corpus in a completely unsupervised way. During training, the model learns to convert the utterances into sequences of embeddings, and then reconstructs the utterances with these sequences of embeddings. A guideline for the proper number of vectors (or words) within an utterance of a given length is needed, in order to prevent the machine from segmenting the utterances into more segments (or words) than needed. Since the number of embeddings is a discrete variable and not differentiable, the standard back-propagation is not applicable[15][16]. The policy gradient for reinforcement learning[17] is therefore used. How these generated word vector sequences carry the phonetic structure information of the original utterances was evaluated with the real application task of query-by-example spoken term detection on four languages: English (on TIMIT), Czech, French, German (on GlobalPhone corpora)[18].

2. PROPOSED APPROACH

2.1. Segmental Sequence-to-Sequence Autoencoder (SSAE)

The proposed structure for SSAE is depicted in Fig. 1, in which the *segmentation gate* is inserted into the recurrent autoencoder. For an input utterance $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$, where $\mathbf{x}_t$ represents the t-th acoustic feature like MFCC and T is the length of the utterance, the model learns to determine the word boundaries and produce the embeddings for the N generated audio segments, $\mathbf{Y} = \{ \mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N\}$, where $\mathbf{e}_n$ is the n-th embedding and $N \leq T$.

The proposed SSAE consists of an encoder RNN (ER) and a decoder RNN (DR) just like the conventional autoencoder. But the encoder includes an extra segmentation gate, controlled by another RNN (shown as a sequence of blocks **S** in Fig. 1). The segmentation problem is formulated as a reinforcement learning problem. At each time t, the segmentation gate agent performs an action a_t, ”segment” or ”pass”, according to a given state $\mathbf{s}_t$. $\mathbf{x}_t$ is taken as a word boundary if a_t is ”segment”.

For the segmentation gate, the state at time t, $\mathbf{s}_t$, is defined as the concatenation of the input $\mathbf{x}_t$, the gate activation signal (GAS) $\mathbf{g}_t$ extracted from the gates of the GRU in another pre-trained RNN autoencoder [10], and the previous action a_{t-1} taken [19],

$$\mathbf{s}_t = \left[\mathbf{x}_t || \mathbf{g}_t || a_{t-1}\right]. \tag{1}$$

The output $\mathbf{h}_t$ of layers of the segmentation gate RNN (blocks S in Fig. 1) followed by a linear transform (W^{π},$\mathbf{b}^{\pi}$) and a softmax nonlinearity models the policy π_t at time t,

$$\mathbf{h}_t = \mathit{RNN}(\mathbf{s}_1, \mathbf{s}_2, ...\mathbf{s}_t), \tag{2}$$

$$\pi_t = \mathit{softmax}(W^{\pi}\mathbf{h}_t + \mathbf{b}^{\pi}). \tag{3}$$

This π_t gives two probabilities respectively for ”segment” and ”pass”. An action a_t is then sampled from this distribution during training to encourage exploration. During testing a_t is ”segment” whenever its probability is higher.

When a_t is ”segment”, the time t is viewed as a word boundary, and the segmentation gate passes the output of encoder RNN as an embedding. The state of the encoder RNN is also reset to its initial value. So the embedding $\mathbf{e}_n$ is generated based on the acoustic features of the audio segment only, independent of the previous input in spite of the recurrent structure,

$$\mathbf{e}_n = Encoder(\mathbf{x}_{t_1}, \mathbf{x}_{t_1+1}, ..., \mathbf{x}_{t_2}), \tag{4}$$

where t_1, t_2 refers to the beginning and ending time for the n-th audio segment.

The input utterance $\mathbf{X}$ should be reconstructed with the embedding sequence $\mathbf{Y} = \{\ \mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N\}$. Because the decoder RNN (DR) is backward in order as shown in Fig. 1 [20], for the embedding $\mathbf{e}_n$ for the input segment from t_1 to t_2 in Eq.(4) above, the reconstructed feature vector is,

$$\hat{\mathbf{x}}_t = Decoder(\hat{\mathbf{x}}_{t_2}, \hat{\mathbf{x}}_{t_2-1}, ...\hat{\mathbf{x}}_{t+1}, \mathbf{e}_n). \tag{5}$$

The decoder RNN is also reset when beginning decoding each audio segment to remove the information flow from the following segment.

2.2. Encoder and Decoder Training

The loss function $\mathcal{L}$ for training the encoder and decoder is simply the averaged squared ℓ-2 norm for the reconstruction error of all input $\mathbf{x}_t$:

$$\mathcal{L} = \sum_l^L \sum_t^{T_l} \frac{1}{d} \left\| \hat{\mathbf{x}}_t^{(l)} - \mathbf{x}_t^{(l)} \right\|^2, \tag{6}$$

where the superscript (l) indicates the l-th training utterance with length T_l, and L is the number of utterances used in training. d is the dimensionality of $\mathbf{x}_t^{(l)}$.

2.3. Segmentation Gate Training

2.3.1. Reinforcement Learning

The segmentation gate is trained with the reinforcement learning. After the segmentation gate performs the segmentation for each utterance, it receives a reward r and a reward baseline r_b for the utterance for updating the parameters. r and r_b will be defined in the next subsection. We can write the expected reward for the gate under policy π as $J(\theta) = \mathbf{E}_{\pi}[r]$, where θ is the parameter set. The updates of the segmentation gate are simply given by:

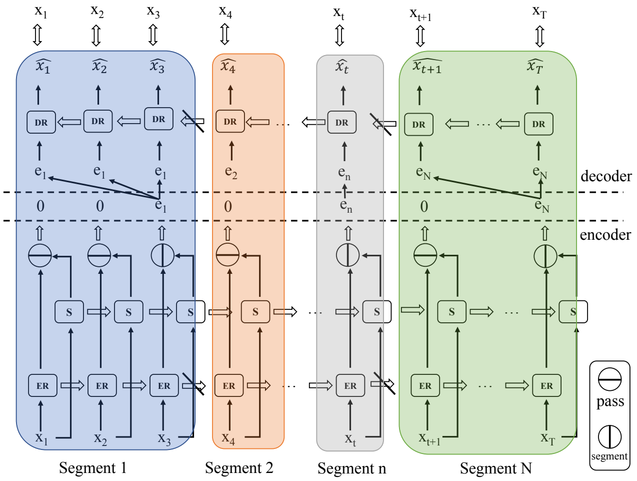


Fig. 1. The segmental sequence-to-sequence autoencoder (SSAE). In addition to the encoder RNN (ER) and decoder RNN (DR), a segmentation gate (blocks **S**) is included in the encoder for estimating the word boundaries. During transitions across the segment boundaries, encoder RNN and decoder RNN are reset (illustrated with a slash in front of an arrow) so there is no information flow across segment boundaries. Each segment (shown in different colors) can be viewed as performing sequence-to-sequence training individually.

$$\nabla_{\theta} J(\theta) = \mathbf{E}_{a \sim \pi}[\nabla_{\theta} \sum_{t=1}^{T} log\pi_t^{(\theta)}(a_t)(r - r_b)], \tag{7}$$

where $\pi_t^{(\theta)}(a_t)$ is the probability for the action a_t taken as in Eq.(3).

2.3.2. Rewards

The reconstruction error is certainly a good indicator to see whether the segmentation boundaries are good, since the embeddings are generated based on the segmentation. We hypothesize that good boundaries, for example those close to word boundaries, would result in smaller reconstruction errors, because the audio segments for words would appear more frequently in the corpus and thus the embeddings would be trained better giving lower reconstruction errors. So the smaller the reconstruction errors the higher the reward:

$$r_{\mathit{MSE}} = -\sum_t^T \frac{1}{d} \left\| \hat{\mathbf{x}}_t - \mathbf{x}_t \right\|^2. \tag{8}$$

This is very similar to Eq.(6) except for a specific utterance here.

On the other hand, a guideline for the proper number of segments (words) N in an utterance of a given length T is important, otherwise for minimizing the reconstruction error as many segments as possible will be generated. So the smaller number of segments N normalized by the utterance length T, the higher the reward:

$$r_{\mathit{N/T}} = -\frac{N}{T}, \tag{9}$$

where N and T are respectively the numbers of segments and frames for the utterance as in Fig. 1.

The total rewards r is obtained by choosing the minimum between r_{MSE} and $r_{N/T}$:

$$r = min(r_{MSE}, \lambda r_{N/T}) \tag{10}$$

where λ is a hyperparameter to be tuned for a reasonable guideline for estimating the proper number of segments for an utterance of length T. In our experiments, this minimum function gave better results than linear interpolation.

We further use utterance-wise reward baseline to remove the bias between utterances. For each utterance, M different sets of segment boundaries are sampled by the segmentation gate, each used to evaluate a reward r_m with Eq.(10). The reward baseline r_b for the utterance is then the average of them:

$$r_b = \frac{1}{M}\sum_{m=1}^{M} r_m. \tag{11}$$

2.4. Iterative Training Process

Although all the models described in sections 2.2 and 2.3 can be trained simultaneously, we actually trained our model with an iterative process consisting of two phases. The first phase is to train the encoder and decoder with Eq.(6) while fixing the parameters of the segmentation gate. The second phase is to update the parameters of the segmentation gate with rewards provided by the encoder and decoder while fixing their parameters. The two phases are performed iteratively. In phase one, the encoder and decoder should be learned from random initialized parameters each time, instead of taking the parameters learned in the previous iteration as the initialization, which was found to offer better training stability.

3. EXAMPLE APPLICATION: UNSUPERVISED QUERY-BY-EXAMPLE SPOKEN TERM DETECTION

This approach can be used in many potential applications. Here we consider the unsupervised query-by-example spoken term detection (QbE STD) as the first example application. The task of unsupervised QbE STD is to locate the occurrence regions of the input spoken query in a large spoken archive without performing speech recognition. With the SSAE proposed here, this can be achieved as illustrated in Fig. 2. Given frame sequences of a spoken query and a spoken document, SSAE can represent these sequences as embeddings, $\mathbf{q}$ = { $\mathbf{q}_1$, $\mathbf{q}_2$, ..., $\mathbf{q}_{N_q}$ } for the query and $\mathbf{d}$ = { $\mathbf{d}_1$, $\mathbf{d}_2$, ..., $\mathbf{d}_{N_d}$ } for the document. With the embeddings, simply subsequence matching can be used to evaluate the relevance score $S(\mathbf{q}, \mathbf{d})$ between $\mathbf{q}$ and $\mathbf{d}$:

$$S(\mathbf{q}, \mathbf{d}) = max(S_1, S_2, ..., S_n, ..., S_{N_d - N_q + 1}), \tag{12}$$

$$S_n = \prod_{m=1}^{N_q} Sim(\mathbf{q}_m, \mathbf{d}_{m+n-1}). \tag{13}$$

Cosine similarity can be used in the similarity measure in Eq.(13). As is clear in the right part of Fig. 2, $S_1 = sim(\mathbf{q}_1, \mathbf{d}_1) \cdot sim(\mathbf{q}_2, \mathbf{d}_2)$, $S_2 = sim(\mathbf{q}_1, \mathbf{d}_2) \cdot sim(\mathbf{q}_2, \mathbf{d}_3)$ and so on. The relevance score $S(\mathbf{q}, \mathbf{d})$ in Eq.(12) between the query and document is then the maximum out of all S_n's obtained in Eq.(13). In this way, the frame-based template matching such as DTW can be replaced by segment-based subsequence matching with much less on-line computation requirements.

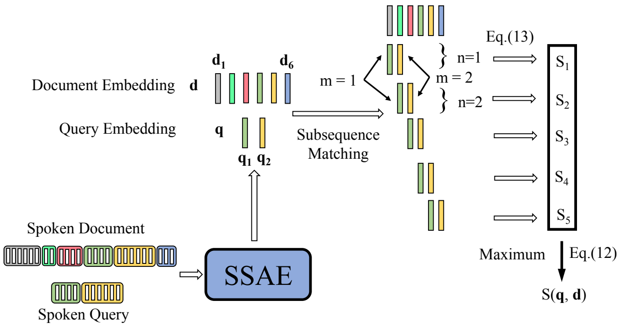


Fig. 2. An illustration of unsupervised spoken term detection performed with segment-based subsequence matching using SSAE.

4. EXPERIMENTS

4.1. Experimental Setup

We performed the experiments on four different languages: English, Czech, French, German. The English corpus was TIMIT and the corpus for the other languages was the GlobalPhone[18]. The ground truth word boundaries for English were provided by TIMIT, while for the other three languages we used the forced aligned word boundaries. Both the encoder and decoder RNNs of the SSAE consisted of one hidden layer of 100 LSTM units[21]. The segmentation gate consisted of 2 layers of LSTM of size 256. All parameters were trained with Adam[22]. $M = 5$ in Eq.(11) in estimating the reward baseline for each utterance. The proximal policy optimization algorithm was used to train the reinforcement learning model[23]. The tolerance window for word segmentation evaluation was taken as 40 ms. The acoustic features used were 39-dim MFCCs with utterance-wise cepstral mean and variance normalization (CMVN) applied. In our experiments $\lambda = 5$ in Eq.(10), which was obtained empirically and obviously had to do with the average duration of the segmented spoken words. In spoken term detection, 5 words for each language containing a variety of phonemes were randomly selected to be the query words as listed in Table 1, and several occurrences for each of them in training set were used as the spoken queries. The testing set utterances were used as spoken documents [8]. The numbers of spoken queries used for evaluation on English, Czech, French and German were 29, 21, 25 and 23 respectively.

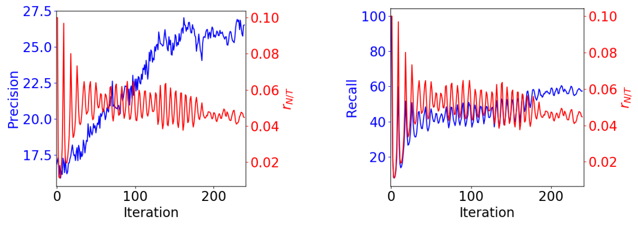
Language	Query Words
English	fail, simple, military, increases, problems
Czech	pracují, použití, textu, demokracie, abych
French	soldats, organisme, travaillant, soulève, sportifs
German	vergeblich, gutem, sozial, großes, ernennung

Table 1. List of the randomly selected query words containing a variety of phonemes for English, Czech, French and German used in the experiments.

4.2. Spoken Word Segmentation Evaluation

Fig. 3 shows the learning curves for SSAE on the Czech validation set. From the figure, we can see that SSAE gradually learned to segment utterances into spoken words because both the precision and recall (blue curves in Fig. 3(a)(b) respectively) got higher when the reward $r_{N/T}$ in Eq.(9) (red curves) converged to a reasonable number. Similar trends were found in the other three languages.

We evaluated the spoken word segmentation performance of the proposed SSAE by comparison with the random segmentation baseline and two segmentation methods, one using Gate Activation Signals (GAS)[10] and the other using the hierarchical agglomerative clustering (HAC)[12][24], and the results are shown in Table 2 in terms of F1 score. Precision (P) and recall (R) were also provided for English. We see that the proposed SSAE performed significantly better than the other two methods on all languages except comparable to GAS for German. Also, for English recall about 50% was achieved while the precision was significantly lower, which implies many of the word boundaries were actually identified, but many spoken words were in fact segmented into subword units. Similar trends were found for other languages.



(a) Precision

(b) Recall

Fig. 3. The learning curves of SSAE on the Czech validation set. The red curves are for $r_{N/T}$ in Eq.(9). The blue curves are (a) precision and (b) recall.

Lang.	English			CZ	FR	GE
Method	P	R	F1	F1		
Random	24.60	41.08	30.77	22.56	32.66	25.41
HAC	26.84	46.21	33.96	30.84	33.75	27.09
GAS	33.22	52.39	40.66	29.53	31.11	32.89
SSAE	37.06	51.55	43.12	37.78	48.14	31.69

Table 2. Spoken word segmentation evaluation compared to different methods across different languages. Random segmentation was also provided as a baseline.

4.3. Spoken Term Detection (STD) Evaluation

We evaluated the quality of embeddings generated by SSAE with the real application of spoken term detection using the method presented in section 3, compared with other kinds of audio Word2Vec embeddings trained with signal segments generated from different segmentation methods. Mean Average Precision (MAP) was used as the performance measure.

The results are listed in Table 3. The performance for embeddings trained with ground truth word boundaries (oracle) in the last column serves as the upper bound. The random baseline in the first column simply assigned a random score to each pair of query and document. We also list the performance of standard frame-based dynamic time warping (DTW) as a primary baseline in the second column[8]. From the table, it is clear that the oracle achieved the best and significantly better performance than all other methods on all languages. SSAE outperformed the DTW baseline by a wide gap. This is probably because DTW may not be able to identify the spoken words if the speaker or gender characteristics are very different, but such different signal characteristics may be better absorbed in the audio Word2Vec training. These experimental results verified that the embeddings obtained with SSAE did carry the sequential phonetic structure information in the utterances, leading to the better performance in STD here. The performance of embeddings trained with GAS and HAC are not too far from random in most cases. It seems the performance of the spoken word segmentation has to be above some minimum level, otherwise the audio Word2Vec couldn't be reasonably trained, or spoken word segmentation boundaries had the major impact on the STD performance.

However, interestingly, although the segmentation performance of GAS was slightly better than SSAE for German, SSAE outperformed GAS a lot on spoken term detection for German. The reason is not clear yet, probably due to some special characteristics of the German language.

			Embeddings (Different Seg.)			
Lang.	Ran.	DTW	GAS	HAC	SSAE	Oracle
Czech	0.38	16.59	0.68	1.13	19.41	22.56
English	0.74	12.02	8.29	0.91	23.27	30.28
French	0.27	11.72	0.40	0.92	21.70	29.66
German	0.18	6.07	0.27	0.26	13.82	21.52

Table 3. The spoken term detection performance in Mean Average Precision (MAP) for the proposed SSAE compared to the audio Word2Vec embeddings trained with spoken words segmented with other methods for different languages. The random baseline (Ran.) simply assigned a random score to each pair of query and document. Standard frame-based DTW is the primary baseline, while the oracle segmentation is the upper bound.

5. CONCLUSION

We propose in this paper the segmental sequence-to-sequence autoencoder (SSAE), which jointly learns and performs the spoken word segmentation and audio word embedding together. This actually extends the audio Word2Vec from word-level to utterance-level. This is achieved by reinforcement learning considering both the reconstruction errors obtained with the embeddings and the reasonable number of words within the utterances. Due to the reset mechanism in SSAE, an embedding is generated only based on an audio segment, therefore can be regarded as the audio word vector representing the segment. This is verified by the improved performance in experiments on unsupervised word segmentation and spoken term detection on four languages.

6. REFERENCES

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[2] Quoc Le and Tomas Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.

[3] Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung-Yi Lee, and Lin-Shan Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," *INTERSPEECH*, 2016.

[4] Wei-Ning Hsu, Yu Zhang, and James Glass, "Learning latent representations for speech generation and transformation," *INTERSPEECH*, 2017.

[5] Guoguo Chen, Carolina Parada, and Tara N Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5236–5240.

[6] Wei-Ning Hsu, Yu Zhang, and James Glass, "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation," *Automatic Speech Recognition and Understanding (ASRU)*, 2017.

[7] Cheng-Tao Chung, Chun-an Chan, and Lin-shan Lee, "Unsupervised spoken term detection with spoken queries by multi-level acoustic patterns with varying model granularity," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7814–7818.

[8] Yaodong Zhang and James R Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 398–403.

[9] Herman Kamper, Aren Jansen, and Sharon Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Computer Speech & Language*, vol. 46, pp. 154–174, 2017.

[10] Yu-Hsuan Wang, Cheng-Tao Chung, and Hung-yi Lee, "Gate activation signal analysis for gated recurrent neural networks and its correlation with phoneme boundaries," *INTERSPEECH*, 2017.

[11] Okko Räsänen, "Basic cuts revisited: Temporal segmentation of speech into phone-like units with statistical learning at a prelinguistic level.," in *CogSci*, 2014.

[12] Yu Qiao, Naoya Shimomura, and Nobuaki Minematsu, "Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3989–3992.

[13] Chia-ying Lee and James Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 40–49.

[14] Sheng-Yi Kong and Lin-shan Lee, "Improved spoken document summarization using probabilistic latent semantic analysis (plsa)," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 1, pp. I–I.

[15] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[16] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio, "Hierarchical multiscale recurrent neural networks," *International Conference on Learning Representations (ICLR)*, 2017.

[17] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[18] Tanja Schultz, "Globalphone: a multilingual speech and text database developed at karlsruhe university.," in *INTERSPEECH*, 2002.

[19] Barret Zoph and Quoc V Le, "Neural architecture search with reinforcement learning," *International Conference on Learning Representations (ICLR)*, 2017.

[20] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[21] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[24] Chun-an Chan, *Unsupervised Spoken Term Detection with Spoken Queries*, Ph.D. thesis, National Taiwan University, 2012.