ADDITIONAL SHARED DECODER ON SIAMESE MULTI-VIEW ENCODERS FOR LEARNING ACOUSTIC WORD EMBEDDINGS

Myunghun Jung, Hyungjun Lim, Jahyun Goo, Youngmoon Jung, and Hoirin Kim

School of Electrical Engineering, KAIST, Daejeon, Republic of Korea

ABSTRACT

Acoustic word embeddings — fixed-dimensional vector representations of arbitrary-length words — have attracted increasing interest in query-by-example spoken term detection. Recently, on the fact that the orthography of text labels partly reflects the phonetic similarity between the words' pronunciation, a multi-view approach has been introduced that jointly learns acoustic and text embeddings. It showed that it is possible to learn discriminative embeddings by designing the objective which takes text labels as well as word segments. In this paper, we propose a network architecture that expands the multi-view approach by combining the Siamese multi-view encoders with a shared decoder network to maximize the effect of the relationship between acoustic and text embeddings in embedding space. Discriminatively trained with multi-view triplet loss and decoding loss, our proposed approach achieves better performance on acoustic word discrimination task with the WSJ dataset, resulting in 11.1% relative improvement in average precision. We also present experimental results on cross-view word discrimination and word level speech recognition tasks.

Index Terms— acoustic word embedding, query-by-example spoken term detection, multi-view learning, Siamese network, encoder-decoder

1. INTRODUCTION

Query-by-example spoken term detection (QbE-STD) is the task of retrieving a spoken query from a set of speech utterances. With the increasing use of smart devices that can interact through the user's voice (e.g. Amazon Echo, Google Home, Apple Siri), the QbE-STD has drawn interest as a technique that can be applied to wake-up or command word detection, search engine, etc.

In earlier works, approaches to compare a spoken query with speech utterances directly were proposed. Feature vectors were calculated from the spoken query or speech utterances at frame-level, which were merged into a feature matrix. The dynamic time warping (DTW) was used to measure the similarity between the feature matrices [1, 2, 3]. Even though DTW is a very intuitive method, its matrix operation leads to high computational cost for each retrieval process. In addition, there are a lot of potential target speech utterances.

As alternatives to approaches based on DTW, approaches to represent a word as a single vector, so-called acoustic word embedding [4, 5], have been introduced. Acoustic word embeddings are fixed-dimensional vector representations of arbitrary-length words, which are differentiated from semantic word embeddings [6, 7] in the way that they reflects phonetic information. Once two words are represented as acoustic word embeddings, it is very easy to measure the similarity between them through a simple vector operation.

Most deep learning approaches using a word as input unit essentially involve the out-of-vocabulary (OOV) problem. In order to handle OOV words in the practical application of acoustic word embeddings, several studies training a Siamese network with a triplet loss [8, 9, 10, 11, 12] have been conducted to learn phonetic similarity between word segments. From the weak supervision that indicates if two word segments are of the same class or not, the network can map words which have similar phonetic properties onto close distributions in embedding space while mapping ones which have different phonetic properties onto distant distributions. When the QbE-STD adopts acoustic word embeddings, it shows better performance than the DTW-based.

In case of given a transcribed speech dataset, a supervised learning method using the weak supervision does not make the best use of the label information. In [13], W. He et al. focused on the fact that the orthography of text labels naturally reflects the phonetic similarity between the words' pronunciation and proposed a multi-view approach for jointly learning acoustic and text embeddings where word segments and text labels were used as two different input views of Siamese encoder networks. This approach modified the weak supervision and Siamese network to suit a multi-view setting.

In this paper, we propose an advanced network architecture that expands [13] by combining a decoder network to the Siamese multi-view encoders. This additional decoder is shared and coupled with the acoustic and text encoders individually and composes an encoder-decoder and an autoencoder structure. In the autoencoder where input is a text label and output is the reconstructed text label, the text embeddings are induced to have more representative capability in embedding space. On the other hand, the encoder-decoder is trained

to be able to decode the original text label from the acoustic embedding. It makes the acoustic embeddings to learn underlying phonetic information, resulting in normalizing speech variances such as gender, age, tone, and intonation. Also, when aligning distributions of acoustic and text embeddings discriminatively in the common space, the shared decoder can support this alignment. Experimental results demonstrate that our proposed approach can learn more discriminative acoustic and text embeddings than previous work on acoustic word discrimination and cross-view word discrimination tasks.

2. MULTI-VIEW APPROACH

First of all, we need to clarify some terms. The single-view means that only a word segment is used as input. Also, we distinguish the acoustic word embeddings of the multi-view approach into two subsets, acoustic and text embeddings, in accordance with their input data type.

In this section, we analyze how acoustic embeddings jointly learned with text embeddings in the multi-view approach [13] can show better performance than ones in the general single-view approach. This analysis has not been done properly in other works.

2.1. Single-view approach

In the single-view approach, a Siamese encoder network is trained with the weak supervision indicating that "word segments x and x^+ are samples of the same word" and "x and x^- are samples of different words". In Fig.1.(a), a triplet (x, x^+, x^-) enters the encoder f, where the dotted line means the Siamese network, and acoustic word embeddings $f(x)$, $f(x^+)$, and $f(x^-)$ are extracted. The main objective makes the distance between $f(x)$ and $f(x^+)$ smaller than the distance between $f(x)$ and $f(x^-)$ by a margin, which is optimized by the following triplet loss:

$$\Big[m + d\left(f(x), f(x^+)\right) - d\left(f(x), f(x^-)\right)\Big]_+, \qquad (1)$$

where $[a]_+$ denotes the rectifier function $max(a, 0)$, the distance function $d(\cdot, \cdot)$ measures the distance between the two embeddings, and m is the margin. The training process of adjusting relative distances between embeddings makes the embeddings of the same word close to each other so that they can be distributed in one cluster in the embedding space.

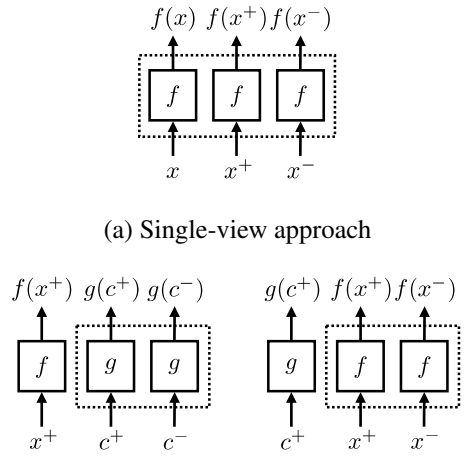
2.2. Multi-view approach

In the multi-view approach, a text label is used as another kind of input view. Like the single-view approach, weak supervision and Siamese network are used, but they should be modified to suit the multi-view setting. When pairs of a word segment and a text label (x, c) are given, the multi-view weak supervision indicates that "x^+ and c^+ are samples of the same word", "x^+ and c^- are samples of different words", and "c^+ and x^- are samples of different words". Noticing that x^- and c^- are not chosen from one pair, three pairs (x^+, c^+), $(x^-, \sim)$, and $(\sim, c^-)$ are used at a time. Here $\sim$ denotes the unused sample. As shown in Fig.1.(b), two triplets (x^+, c^+, c^-) and (c^+, x^+, x^-) that can be taken from this multi-view weak supervision enter the acoustic encoder f and text encoder g. In both cases, the Siamese network is applied to samples having the same view, not to all three samples of the triplet. According to these multi-view triplet cases, the main objective consists of two parts (obj^0+obj^2 in [13]). One makes the distance between $f(x^+)$ and $g(c^+)$ smaller than the distance between $f(x^+)$ and $g(c^-)$ by a margin, and the other makes the distance between $g(c^+)$ and $f(x^+)$ smaller than the distance between $g(c^+)$ and $f(x^-)$ by the margin. This objective is optimized by the following two multi-view triplet losses:

$$\Big[m + d\left(f(x^+), g(c^+)\right) - d\left(f(x^+), g(c^-)\right)\Big]_+, \qquad (2)$$

$$\Big[m + d\left(g(c^+), f(x^+)\right) - d\left(g(c^+), f(x^-)\right)\Big]_+. \qquad (3)$$

As an anchor component of a triplet always has a different view from their positive and negative components, distances are indirectly adjusted through cross alignment between the acoustic and text embeddings. Although the direct distance adjustment like the single-view approach is not achieved, embeddings can be effectively clustered in embedding space due to the uniqueness of text labels. The encoder f does not output the same acoustic embeddings even if input word segments express the same word, because speech appears as different instances every time. In contrast, the text label of the word is unique and it means that the encoder g outputs only



(a) Single-view approach

(b) Multi-view approach

Fig. 1. Examples of triplet and Siamese encoder network for (a) single-view approach and (b) multi-view approach.

one text embedding for one word. At every training step, this unique text embeddings act as pivot points for the acoustic embeddings to be easily centralized, which is expressed in Eq.3.

The ultimate goal of learning acoustic word embeddings is to find inherent phonetic information for each word and to group similar ones in embedding space. From this point of view, the single-view approach gathers embeddings of each word into individual clusters by extracting the common characteristics which, however, should be founded in the process of looking at many and various word samples without knowing where to focus on. On the other hand, in the multi-view approach, it is assumed that each text embedding already represents inherent phonetic information. Therefore, what networks need to do is simply to make acoustic embeddings be close to each text embedding. At the same time, text embeddings are also trained to work as better reference points, so that the initial assumption becomes more influential. Thus, the multi-view approach can learn more discriminative embeddings by utilizing text labels more effectively than the weak supervision of the single-view approach. We use this multi-view approach as a baseline for our proposed approach.

3. PROPOSED APPROACH

Contrary to single-view supervised learning based on a Siamese network, unsupervised learning methods for acoustic word embeddings mainly use an autoencoder structure [14, 15, 16, 17]. The architecture consists of an encoder network that extracts acoustic word embeddings from word segments and a decoder network that reconstructs the input segments from the embeddings. Since one output word segment must be generated from only one embedding, embeddings are trained to compress the most essential information of given inputs.

An encoder-decoder structure is a general framework, which was used for learning semantic word embeddings [7] or machine translation [18, 19]. It is the same as the autoencoder in that it consists of an encoder network and a decoder network, but it is distinguished in that it generates a different view of output from the input data. Although input and output data are obtained from different views, they refer to the fundamentally equivalent content. Therefore, the embeddings should be trained to represent underlying information that is shared between input and output data.

The most important part of the multi-view approach we analyzed was the assumption that the text embeddings represent inherent phonetic information of the words. Although text embeddings move into better reference points at every training step, this is done by the incidental effect of the multi-view triplet loss, especially Eq.2. In order to further improve the performance, it was necessary to introduce an additional objective to consolidate the assumption. Thus, we paid attention to the role of the autoencoder in learning acoustic word embeddings and thought that it could be used for the enhancement of the text embeddings. We also found that combining the decoder used in the autoencoder with the acoustic encoder makes it possible to take advantage of the encoder-decoder structure. Based on these ideas, we have motivated a study that fully utilizes the autoencoder and encoder-decoder structure together to assist in learning acoustic and text embeddings. In a similar study [20], Z. Zhu et al. showed that acoustic word embeddings can be learned by combining the Siamese network of the single-view approach with the autoencoder structure.

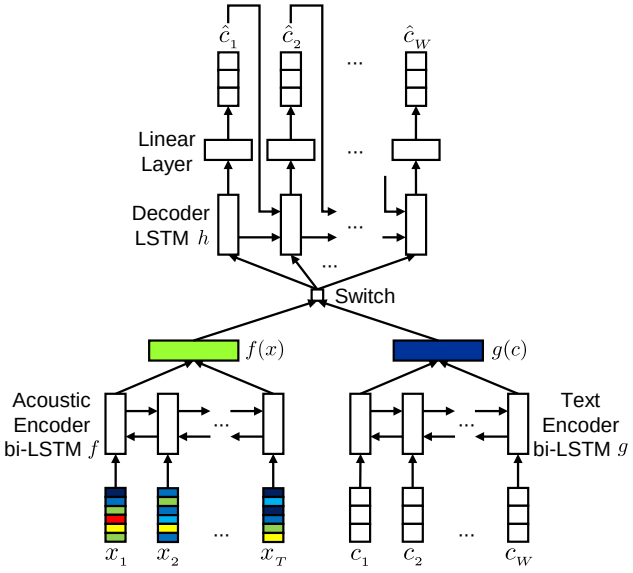


Fig. 2. Illustration of proposed network architecture.

3.1. Network architecture

In this paper, we propose an advanced network architecture that expands the multi-view approach in Sec.2.2 by combining a decoder network to the Siamese multi-view encoders. As shown in Fig.2, this additional decoder h is coupled and shared with the acoustic encoder f and text encoder g individually, so that composes acoustic-to-text encoder-decoder $(f \leftrightarrow h)$ and text-to-text autoencoder $(g \leftrightarrow h)$. Given a pair of a word segment and a text label (x, c), the encoder f takes $x = \{x_t\}_{t=1}^{T}$ and outputs the acoustic embedding $f(x)$, where x_t is the acoustic feature vector at frame t and T is the length of the x. Also, the encoder g takes $c = \{c_w\}_{w=1}^{W}$ and outputs the text embedding $g(c)$, where c_w is the w-th character one-hot vector and W is the length of the c. Through the switch, either $f(x)$ or $g(c)$ is fed into the decoder h generating a predicted or reconstructed text label $\hat{c} = \{\hat{c}_y\}_{y=1}^{W}$, where $\hat{c}_y$ is the probability distribution vector over all character classes for the y-th character.

Since acoustic feature vectors and character vectors ex-

press sequential data, we implement the encoder f and g with multi-layer bidirectional long short term memory (bi-LSTM) networks [21, 22]. At the output layer, the last hidden states of two directions are concatenated to form an embedding. For the decoder h, a multi-layer unidirectional LSTM is used, but previous output $\hat{c}_{y-1}$ is used as an auxiliary input to calculate $\hat{c}_y$ at each step $y = 2, 3, ..., W$. The hidden states of the output layer are transformed into lower dimension vectors through the fully-connected linear layer, which has the same number of output nodes as the character classes. Then $\hat{c}$ is calculated by the softmax operation.

3.2. Training objective

In training of the proposed network architecture, a new objective is used in addition to the objective used in the multi-view approach. The new objective corresponding to the decoder h is that a decoded output $\hat{c}$ should be identical to text label c regardless of whether the input embedding is $f(x)$ or $g(c)$. This objective is optimized by the following decoding loss which is the sum of two cross-entropy losses:

$$L_{decoding} = \sum_{i=1}^{N}\Bigg(-\sum_{y=1}^{W}\Big(c_y^{i,+} \cdot log\left(\hat{c}_y \mid h(f(x^{i,+}))\right) + c_y^{i,+} \cdot log\left(\hat{c}_y \mid h(g(c^{i,+}))\right)\Big)\Bigg), \quad (4)$$

where $(x^{i,+}, c^{i,+})$ is the i-th paired input data, N is the size of training mini-batch, and $\cdot$ is element-wise dot product.

By the original role of the autoencoder, we can let the text embedding itself learn the identity of the word. Deeply related to the uniqueness of text labels, the decoding loss maximizes the representative capability of the text embeddings in embedding space. Also, by letting the shared decoder generate the same output from the paired acoustic and text embedding, these embeddings are tightly aligned. Moreover, one target output is predicted from word segments having the same text label, allowing the acoustic embeddings to learn inherent phonetic information between two input views and to normalize speech variances. This normalization effect is the result of the alignment and learning underlying information, although the word segments exist in various instances.

The overall training loss L_{total} is the sum of multi-view triplet loss and decoding loss as follows:

$$L_{total} = L_{triplet} + \alpha L_{decoding}, \quad (5)$$

where α is a hyper-parameter which weights the decoding loss $L_{decoding}$, and $L_{triplet}$ is the sum of multi-view triplet losses from Eq.2, 3. $L_{triplet}$ is as follows:

$$L_{triplet} = \sum_{i=1}^{N}\Bigg(\Big[m + d\left(f(x^{i,+}), g(c^{i,+})\right) - d\left(f(x^{i,+}), g(c^{i,-})\right)\Big]_+ + \Big[m + d\left(g(c^{i,+}), f(x^{i,+})\right) - d\left(g(c^{i,+}), f(x^{i,-})\right)\Big]_+\Bigg), \quad (6)$$

where $(x^{i,-}, \sim)$ and $(\sim, c^{i,-})$ are uniformly sampled negative input pairs from all of the differently labeled pairs in the training set according to the multi-view weak supervision. In this paper we use the cosine distance, $d(\vec{p}, \vec{q}) = 1 - \frac{\vec{p}\cdot\vec{q}}{\|\vec{p}\|\|\vec{q}\|}$.

4. EXPERIMENTS AND RESULTS

4.1. Evaluation tasks

Our original purpose is to improve the performance of the QbE-STD task, but it can be substituted with a word discrimination task on the condition that the word boundary information is known. To measure the performance, we consider next two word discrimination tasks which are applicable in the multi-view setting.

The first task is *acoustic word discrimination*, where we are given two word segments to determine whether they match or not. This task is equivalent to the objective of the single-view approach and has been used in prior papers [9, 10, 11, 12, 14, 17]. We regard this task as our main evaluation task for training the proposed and baseline network architectures.

The second task is *cross-view word discrimination* corresponding to an audio-text QbE-STD where we retrieve a text query from a set of speech utterances or a spoken query from a set of text documents. If a word segment and text label are given, we have to determine whether they indicate the same word or not. This is very useful in that two different types of data can be easily integrated into one process.

In these two tasks, the cosine distance between embeddings of given two word expressions (word segment or text label) is calculated. If the distance is below a threshold, we decide they are same, otherwise they are different words. For the performance measure, we use the average precision (AP) which is the area under the precision-recall curve generated by sweeping the threshold.

4.2. Dataset

The data used for experiments was drawn from the Continuous Speech Recognition Wall Street Journal Corpus (WSJ) [23] Phase I and Phase II, specifically si_tr_s for the training set, si_dt_05 for the development set, and si_et_05, si_et_h2 for the test set. All utterances were segmented into (x, c) pairs

Table 1. Effect of various α in the development set AP for initial model and tuned model.

α	0	0.01	0.05	0.1	0.3	0.6
Initial	0.747	0.765	0.780	**0.787**	0.778	0.768
Tuned		0.814	0.817	**0.830**	0.825	0.817

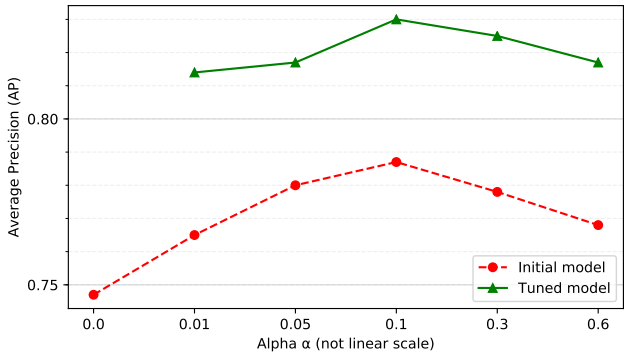


Fig. 3. Tendency of varying the hyper-parameter α for initial model and tuned model.

by the forced alignment of the transcriptions from the GMM-HMM speech recognizer trained on the same training set using the open-source Kaldi toolkit [24]. The number of pairs in training, development, and test set are 629897, 8616, and 9194 where the number of unique words are 13365, 1867, and 1728. Having 42 million possible combinations of all pairs in the test set, there are 341932 matched cases and 41918289 unmatched cases.

We represented the acoustic feature vector x_t using 40-dimensional Mel-filterbank energy which was calculated from 25 ms frame with 15 ms overlap. The 26-dimensional character one-hot vector c_w was generated by text normalizing that converts capitals and numbers into lower cases and removes quotation and punctuation, e.g. "It's 7 o'clock." $\rightarrow$ "its seven oclock" $\rightarrow$ {i/t/s}{s/e/v/e/n}{o/c/l/o/c/k}.

4.3. Experimental setup

Before the experiment, we implemented the baseline multi-view approach [13] and trained it with the model and dataset provided by the authors to verify the performance improvement compared to the single-view approaches [9, 10]. Then we established our initial model parameters as the same with the retuned baseline model on the WSJ dataset.

As the initial model for our proposed approach, 2-layer bi-LSTMs with 512 hidden units per direction were used for acoustic and text encoders. Their weights were randomly initialized. For the additional shared decoder which does not exist in the baseline architecture, we used a 2-layer LSTM with 512 hidden units. The last states of each forward directional

Table 2. Coarse grid search results in terms of the development set AP.

# of layers	1	2	**3**
	0.566	0.787	**0.830**
# of hidden units	128	256	**512**
	0.748	0.803	**0.830**
m	0.3	**0.5**	0.7
	0.802	**0.830**	0.800
N	32	128	**256**
	0.806	0.819	**0.830**

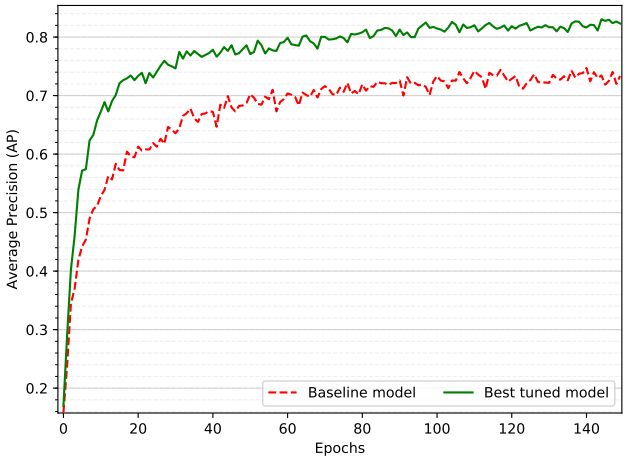


Fig. 4. Progression of the development set AP for training baseline model and best tuned model on acoustic word discrimination.

layer in one of both encoders were used as the first states of each layer in the decoder. The linear layer consists of 128 hidden nodes and 26 output softmax nodes. In training, we applied dropout with the rate of 0.4 except on the inputs of the text encoder because of sparsity. We used the Adam optimization algorithm [25] with learning rate of 0.0001, β_1 = 0.9, β_2 = 0.999, and $\epsilon = 10^{-8}$. The mini-batch size N was set to 256 to maximize the use of memory capacity of two GTX 1080 Ti GPUs. Models were trained for 150 epochs while evaluation for the development set was performed every epoch and finally the model having the highest development set AP was selected. The baseline and proposed models were implemented in PyTorch [26].

4.4. Model parameters tuning

To investigate the effectiveness of the additional shared decoder, we checked the development set AP for the initial model by changing the decoding loss weight α from 0, where the model is identical to the baseline, to 0.01, 0.05, 0.1, 0.3,

Table 3. Final test set AP for the baseline multi-view approach and our proposed approach on acoustic word discrimination and cross-view word discrimination.

Model		Acoustic	Cross-view
Baseline [13]		0.791	0.910
Proposed	Initial	0.841	0.935
	Best tuned	**0.879**	**0.948**

and 0.6. As can be seen in Table 1 and Fig.3 (red dashed line), we found that the performance increases with α ranging from 0 to 0.1 and decreases with $\alpha > 0.1$.

Our proposed model was tuned by varying the parameters from the initial model of Sec.4.3. We performed a coarse grid search in order of the number of LSTM layers over {1, 2, **3**}, the number of hidden units over {128, 256, **512**}, the margin m over {0.3, **0.5**, 0.7}, the mini-batch size N over {32, 128, **256**}, and the α over {0.01, 0.05, **0.1**, 0.3, 0.6}. The numbers in bold are the selected parameters of the best tuned model. The effect of varying the value of α is plotted in Fig.3 (green line), which confirms the tendency of the model to perform best when α is 0.1. The performance results of the coarse grid search are tabulated in Table 2.

Fig.4 shows the development set AP versus training epoch for the baseline model and best tuned model on acoustic word discrimination task. We can observe that the performance gap of about 0.07 is maintained during the whole training process.

4.5. Results

This paper does not include the results of DTW-based or single-view approaches. Because in [13], the multi-view approach achieved significant performance improvement compared to previous approaches, we omit the unnecessary verification experiments.

In Table 3, we compare the final performance on the test set between the models of the baseline multi-view approach and our proposed approach. We can clearly see that our proposed approach outperforms the baseline as in the previously observed results from the development set. Our proposed approach achieved an AP of 0.879 on acoustic word discrimination task, which is 11.1% relative improvement. On cross-view word discrimination task, both approaches achieved a high AP over 0.9, because they were trained by adjusting the distance between cross-view embeddings with the multi-view triplet loss. Here 4.2% relative improvement was obtained, 0.948.

4.6. Word-level speech recognition

The decoder h can be used in the test phase as well as in the training. We extracted the acoustic embeddings of the test set and generated the decoded outputs from them to perform the word-level speech recognition. Character error rate (CER) was calculated by comparing the decoded outputs with original text labels. Of the words in the test set, 42.4% CER was obtained for the in-vocabulary (IV) words seen in the training and 56.6% CER was obtained for the OOV words. Table 4 shows incorrectly decoded ones among the examples for randomly selected words of the test set. Although we obtained rather high CERs for OOV and also IV words, our proposed approach can be used as a pre-trained model for a word-level speech recognizer, judging from the plausible phonetic similarity which can be observed between the decoded outputs and actual text labels.

Table 4. Examples of incorrectly decoded outputs and their original text labels for randomly selected words in the test set.

Decoded output	remardin, hell, riiaa, digisting, tue, traik, ougust, blacforne, uv, genvrll, texaso, edecation, simene, turmantiu
Text label	remained, held, ryder, digesting, two, trade, august, blackburn, of, javelin, texans, education, symbol, terminate

5. CONCLUSION

In this paper, we proposed an approach for jointly learning acoustic and text embeddings by introducing an advanced multi-view network architecture where an additional decoder is coupled and shared with the acoustic and text encoders. In addition to multi-view triplet loss which allows embeddings to learn the phonetic similarity between words, decoding loss from encoder-decoder and autoencoder structures was also considered to maximize the representative capability and achieve normalization effect of embeddings. We have found that consistent performance improvements can be obtained on word discrimination tasks. Also, our proposed network architecture shows its potential to be used as a pre-trained model for a word-level speech recognizer. Future work will consider a method that directly measures the phonetic similarity between text labels so that a large amount of text corpus can be used to train text embeddings in advance.

6. ACKNOWLEDGEMENT

This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program (No.10063424, Development of distant speech recognition and multi-task dialog processing technologies for in-door conversational robots).

7. REFERENCES

[1] Hiroaki Sakoe, Seibi Chiba, A Waibel, and KF Lee, "Dynamic programming algorithm optimization for spoken word recognition," *Readings in speech recognition*, vol. 159, pp. 224, 1990.

[2] Timothy J Hazen, Wade Shen, and Christopher White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2009, pp. 421–426.

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

[4] Keith Levin, Katharine Henry, Aren Jansen, and Karen Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 410–415.

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

[6] 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.

[7] Yu-An Chung and James Glass, "Speech2Vec: A sequence-to-sequence framework for learning word embeddings from speech," in *Proceedings of Annual Conference of the International Speech Communication Association*, 2018, pp. 811–815.

[8] Samy Bengio and Georg Heigold, "Word embeddings for speech recognition," in *Proceedings of Annual Conference of the International Speech Communication Association*, 2014, pp. 1053–1057.

[9] Herman Kamper, Weiran Wang, and Karen Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4950–4954.

[10] Shane Settle and Karen Livescu, "Discriminative acoustic word embeddings: Recurrent neural network-based approaches," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 503–510.

[11] Shane Settle, Keith Levin, Herman Kamper, and Karen Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Proceedings of Annual Conference of the International Speech Communication Association*, 2017, pp. 2874–2878.

[12] Yougen Yuan, Cheung-Chi Leung, Lei Xie, Hongjie Chen, Bin Ma, and Haizhou Li, "Learning acoustic word embeddings with temporal context for query-by-example speech search," in *Proceedings of Annual Conference of the International Speech Communication Association*, 2018, pp. 97–101.

[13] Wanjia He, Weiran Wang, and Karen Livescu, "Multi-view recurrent neural acoustic word embeddings," *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017.

[14] 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," in *Proceedings of Annual Conference of the International Speech Communication Association*, 2016, pp. 765–769.

[15] Kartik Audhkhasi, Andrew Rosenberg, Abhinav Sethy, Bhuvana Ramabhadran, and Brian Kingsbury, "End-to-end ASR-free keyword search from speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1351–1359, 2017.

[16] Yu-Hsuan Wang, Hung-yi Lee, and Lin-shan Lee, "Segmental audio Word2Vec: Representing utterances as sequences of vectors with applications in spoken term detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6269–6273.

[17] Herman Kamper, "Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6535–3539.

[18] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[19] 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.

[20] Ziwei Zhu, Zhiyong Wu, Runnan Li, Helen Meng, and Lianhong Cai, "Siamese recurrent auto-encoder representation for query-by-example spoken term detection," in *Proceedings of Annual Conference of the International Speech Communication Association*, 2018, pp. 102–106.

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

[22] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[23] Douglas B Paul and Janet M Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[24] Daniel Povey, Arnab Ghoshal, and Gilles Boulianne, "The Kaldi speech recognition toolkit," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE Signal Processing Society, 2011.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *Proc. Int. Conf. on Learning Representations (ICLR)*, 2015.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.