PHONETIC-AND-SEMANTIC EMBEDDING OF SPOKEN WORDS WITH APPLICATIONS IN SPOKEN CONTENT RETRIEVAL

Yi-Chen Chen, Sung-Feng Huang, Chia-Hao Shen, Hung-yi Lee, Lin-shan Lee

National Taiwan University, Taiwan

ABSTRACT

Word embedding or Word2Vec has been successful in offering semantics for text words learned from the context of words. Audio Word2Vec was shown to offer phonetic structures for spoken words (signal segments for words) learned from signals within spoken words. This paper proposes a two-stage framework to perform phonetic-and-semantic embedding on spoken words considering the context of the spoken words. Stage 1 performs phonetic embedding with speaker characteristics disentangled. Stage 2 then performs semantic embedding in addition. We further propose to evaluate the phonetic-and-semantic nature of the audio embeddings obtained in Stage 2 by parallelizing with text embeddings.

In general, phonetic structure and semantics inevitably disturb each other. For example the words "brother" and "sister" are close in semantics but very different in phonetic structure, while the words "brother" and "bother" are in the other way around. But phonetic-and-semantic embedding is attractive, as shown in the initial experiments on spoken document retrieval. Not only spoken documents including the spoken query can be retrieved based on the phonetic structures, but spoken documents semantically related to the query but not including the query can also be retrieved based on the semantics.

Index Terms— phonetic-and-semantic embedding, spoken content retrieval

1. INTRODUCTION

Word embedding or Word2Vec [1, 2, 3, 4] has been widely used in the area of natural language processing [5, 6, 7, 8, 9, 10, 11], in which text words are transformed into vector representations of fixed dimensionality [12, 13, 14]. This is because these vector representations carry plenty of semantic information learned from the context of the considered words in the text training corpus. Similarly, audio Word2Vec has also been proposed in the area of speech signal processing, in which spoken words (signal segments for words without knowing the underlying word it represents) are transformed into vector representations of fixed dimensionality [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]. These vector representations carry the phonetic structures of the spoken words learned from the signals within the spoken words, and have been shown to be useful in spoken term detection, in which the spoken terms are detected simply based on the phonetic structures. Such Audio Word2Vec representations do not carry semantics, because they are learned from individual spoken words only without considering the context.

Audio Word2Vec was recently extended to Segmental Audio Word2Vec [26], in which an utterance can be automatically segmented into a sequence of spoken words [27, 28, 29, 30] and then transformed into a sequence of vectors of fixed dimensionality by Audio Word2Vec, and the spoken word segmentation and Audio Word2Vec can be jointly trained from an audio corpus. In this way the Audio Word2Vec was upgraded from word-level to utterance-level. This offers the opportunity for Audio Word2Vec to include semantic information in addition to phonetic structures, since the context among spoken words in utterances bring semantic information. This is the goal of this work, and this paper reports the first set of results towards such a goal.

In principle, the semantics and phonetic structures in words inevitably disturb each other. For example, the words "brother" and "sister" are close in semantics but very different in phonetic structure, while the words "brother" and "bother" are close in phonetic structure but very different in semantics. This implies the goal of embedding both phonetic structures and semantics for spoken words is naturally very challenging. Text words can be trained and embedded as vectors carrying plenty of semantics because the phonetic structures are not considered at all. On the other hand, because spoken words are just a different version of representations for text words, it is also natural to believe they do carry some semantic information, except disturbed by phonetic structures plus some other acoustic factors such as speaker characteristics and background noise [31, 32, 33, 34, 35, 36]. So the goal of embedding spoken words to carry both phonetic structures and semantics is possible, although definitely hard.

But a nice feature of such embeddings is that they may include both phonetic structures and semantics [37, 38]. A direct application for such phonetic-and-semantic embedding of spoken words is spoken document retrieval [39, 40, 41, 42, 43]. This task is slightly different from spoken term detection, in the latter case spoken terms are simply detected based on the phonetic structures. Here the goal of the task is to retrieve

all spoken documents (sets of consecutive utterances) relevant to the spoken query, which may or may not include the query. For example, for the spoken query of "President Donald Trump", not only those documents including the spoken query should be retrieved based on the phonetic structures, but those documents including semantically related words such as "White House" and "trade policy", but not necessarily "President Donald Trump", should also be retrieved. This is usually referred to as "semantic retrieval", which can be achieved by the phonetic-and-semantic embedding discussed here.

This paper proposes a two-stage framework of phonetic-and-semantic embedding for spoken words. Stage 1 performs phonetic embedding but with speaker characteristics disentangled using separate phonetic and speaker encoders and a speaker discriminator. Stage 2 then performs semantic embedding in addition. We further propose to evaluate the phonetic-and-semantic nature of the audio embeddings obtained in Stage 2 by parallelizing with text embeddings [44, 45]. Very encouraging results including those for an application task of spoken document retrieval were obtained in the initial experiments[1].

2. PROPOSED APPROACH

The proposed framework of phonetic-and-semantic embedding of spoken words consists of two stages:

Stage 1 - Phonetic embedding with speaker characteristics disentangled.

Stage 2 - Semantic embedding over phonetic embeddings obtained in Stage 1.

In addition, we propose an approach for parallelizing the audio and text embeddings to be used for evaluating the phonetic and semantic information carried by the audio embeddings. These are described in Subsections 2.1, 2.2 and 2.3 respectively.

2.1. Stage 1 - Phonetic Embedding with Speaker Characteristics Disentangled

A text word with a given phonetic structure corresponds to infinite number of audio signals with varying acoustic factors such as speaker characteristics, microphone characteristics, background noise, etc. All the latter acoustic factors are jointly referred to as speaker characteristics here for simplicity, which obviously disturbs the goal of phonetic-and-semantic embedding. So Stage 1 is to obtain phonetic embeddings only with speaker characteristics disentangled.

Also, because the training of phonetic-and-semantic embedding is challenging, in the initial effort we slightly simplify the task by assuming all training utterances have been properly segmented into spoken words. Because there exist many approaches for segmenting utterances automati-
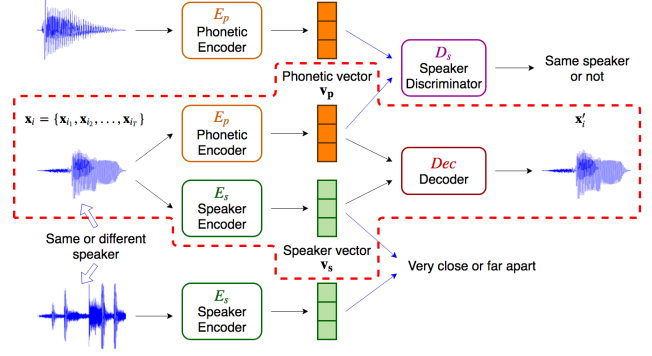
Fig. 1. Phonetic embedding with speaker characteristics disentangled.

cally [26], and automatic segmentation plus phonetic embedding of spoken words has been successfully trained and reported before [26], such an assumption is reasonable here.

We denote the audio corpus as $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{M}$, which consists of M spoken words, each represented as $\mathbf{x}_i = (\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, ..., \mathbf{x}_{i_T})$, where $\mathbf{x}_{i_t}$ is the acoustic feature vector for the tth frame and T is the total number of frames in the spoken word. The goal of Stage 1 is to disentangle the phonetic structure and speaker characteristics in acoustic features, and extract a vector representation for the phonetic structure only.

2.1.1. Autoencoder

As shown in the middle of Figure 1, a sequence of acoustic features $\mathbf{x}_i = (\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, ..., \mathbf{x}_{i_T})$ is entered to a phonetic encoder E_p and a speaker encoder E_s to obtain a phonetic vector $\mathbf{v_p}$ in orange and a speaker vector $\mathbf{v_s}$ in green. Then the phonetic and speaker vectors $\mathbf{v_p}$, $\mathbf{v_s}$ are used by the decoder Dec to reconstruct the acoustic features $\mathbf{x}'$. This phonetic vector $\mathbf{v_p}$ will be used in the next stage as the phonetic embedding. The two encoders E_p, E_s and the decoder Dec are jointly learned by minimizing the reconstruction loss below:

$$L_r = \sum_i \|\mathbf{x}_i - Dec(E_p(\mathbf{x}_i), E_s(\mathbf{x}_i))\|_2^2. \qquad (1)$$

It will be clear below how to make E_p and E_s separately encode the phonetic structure and speaker characteristics.

2.1.2. Training Criteria for Speaker Encoder

The speaker encoder training requires speaker information for the spoken words. Assume the spoken word $\mathbf{x}_i$ is uttered by speaker s_i. When the speaker information is not available, we can simply assume that the spoken words in the same utterance are produced by the same speaker. As shown in the lower

part of Figure 1, E_s is learned to minimize the following loss:

$$L_s = \sum_{s_i = s_j} \|\mathbf{v}_{\mathbf{s}i} - \mathbf{v}_{\mathbf{s}j}\|_2^2$$
$$+ \sum_{s_i \neq s_j} \max(\lambda - \|\mathbf{v}_{\mathbf{s}i} - \mathbf{v}_{\mathbf{s}j}\|_2^2, 0). \quad (2)$$

In other words, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are uttered by the same speaker ($s_i = s_j$), we want their speaker embeddings $\mathbf{v}_{\mathbf{s}i}$ and $\mathbf{v}_{\mathbf{s}j}$ to be as close as possible. But if $s_i \neq s_j$, we want the distance between $\mathbf{v}_{\mathbf{s}i}$ and $\mathbf{v}_{\mathbf{s}j}$ larger than a threshold λ.

2.1.3. Training Criteria for Phonetic Encoder

As shown in the upper right corner of Figure 1, a speaker discriminator D_s takes two phonetic vectors $\mathbf{v}_{\mathbf{p}i}$ and $\mathbf{v}_{\mathbf{p}j}$ as input and tries to tell if the two vectors come from the same speaker. The learning target of the phonetic encoder E_p is to "fool" this speaker discriminator D_s, keeping it from discriminating the speaker identity correctly. In this way, only the phonetic structure information is learned in the phonetic vector $\mathbf{v}_{\mathbf{p}}$, while only the speaker characteristics is encoded in the speaker vector $\mathbf{v}_{\mathbf{s}}$. The speaker discriminator D_s learns to maximize L_d in (3), while the phonetic encoder E_p learns to minimize L_d,

$$L_d = \sum_{s_i = s_j} D_s(\mathbf{v}_{\mathbf{p}i}, \mathbf{v}_{\mathbf{p}j}) - \sum_{s_i \neq s_j} D_s(\mathbf{v}_{\mathbf{p}i}, \mathbf{v}_{\mathbf{p}j}). \quad (3)$$

where $D_s(\cdot, \cdot)$ is a real number.

2.1.4. Overall Optimization of Stage 1

The optimization procedure of Stage 1 consists of four parts: (1) training E_p, E_s and Dec by minimizing L_r, (2) training E_s by minimizing L_s, (3) training E_p by minimizing L_d, and (4) training D_s by maximizing L_d. Parts (1)(2)(3) are jointly trained together, while iteratively trained with part (4) [46].

2.2. Stage 2 - Semantic Embedding over Phonetic Embeddings Obtained in Stage 1

As shown in Figure 2, similar to the Word2Vec skip-gram model [1], we use two encoders: semantic encoder E_{sem} and context encoder E_{ctx} to embed the semantics over phonetic embeddings $\mathbf{v}_{\mathbf{p}}$ obtained in Stage 1. On the one hand, given a spoken word $\mathbf{x}_i$, we feed its phonetic vector $\mathbf{v}_{\mathbf{p}i}$ obtained from Stage 1 into E_{sem} as in the middle of Figure 2, producing the semantic embedding (in yellow) of the spoken word $\mathbf{v}_{\mathbf{w}i} = E_{sem}(\mathbf{v}_{\mathbf{p}i})$. On the other hand, given the context window size c, which is a hyperparameter, if a spoken word $\mathbf{x}_j$ is in the context window of $\mathbf{x}_i$, then its phonetic vector $\mathbf{v}_{\mathbf{p}j}$ is a context vector of $\mathbf{v}_{\mathbf{p}i}$. For each context vector $\mathbf{v}_{\mathbf{p}j}$ of $\mathbf{v}_{\mathbf{p}i}$, we feed it into the context encoder E_{ctx} in the upper part of Figure 2, and the output is the context embedding $\mathbf{v}_{\mathbf{c}j} = E_{ctx}(\mathbf{v}_{\mathbf{p}j})$.
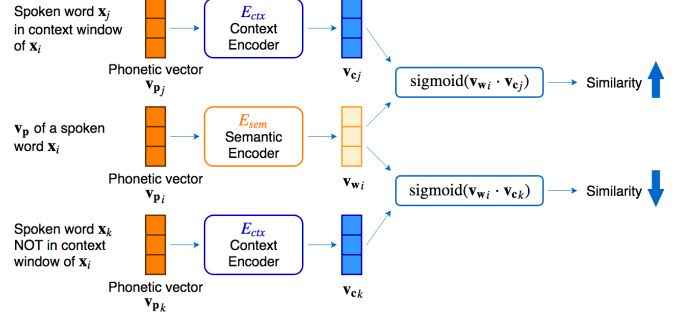

Fig. 2. Semantic embedding over phonetic embeddings obtained in Stage 1.

Given a pair of phonetic vectors $(\mathbf{v}_{\mathbf{p}i}, \mathbf{v}_{\mathbf{p}j})$, the training criteria for E_{sem} and E_{ctx} is to maximize the similarity between $\mathbf{v}_{\mathbf{w}i}$ and $\mathbf{v}_{\mathbf{c}j}$ if $\mathbf{v}_{\mathbf{p}i}$ and $\mathbf{v}_{\mathbf{p}j}$ are contextual, while minimizing the similarity otherwise. The basic idea is parallel to that of text Word2Vec. Two different spoken words having similar context should have similar semantics. Thus if two different phonetic embeddings corresponding to two different spoken words have very similar context, they should be close to each other after projected by the semantic encoder E_{sem}. The semantic and context encoders E_{sem} and E_{ctx} learn to minimize the semantic loss L_{sem} as follows:

$$L_{sem} = \sum_{(\mathbf{x_i}, \mathbf{x_j}) \text{ in context window}} -\log(\text{sigmoid}(\mathbf{v}_{\mathbf{w}i} \cdot \mathbf{v}_{\mathbf{c}j}))$$
$$+ \sum_{(\mathbf{x_i}, \mathbf{x_k}) \text{ not in context window}} -\log(\text{sigmoid}(-\mathbf{v}_{\mathbf{w}i} \cdot \mathbf{v}_{\mathbf{c}k})). \quad (4)$$

The sigmoid of dot product of $\mathbf{v}_{\mathbf{w}}$ and $\mathbf{v}_{\mathbf{c}}$ is used to evaluate the similarity. With (4), if $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same context window, we want $\mathbf{v}_{\mathbf{w}i}$ and $\mathbf{v}_{\mathbf{c}j}$ to be as similar as possible. We also use the negative sampling technique, in which only some pairs $(\mathbf{x_i}, \mathbf{x_k})$ are randomly sampled as negative examples instead of enumerating all possible negative pairs.

2.3. Parallelizing Audio and Text Embeddings for Evaluation Purposes

In this paper we further propose an approach of parallelizing a set of audio embeddings (for spoken words) with a set of text embeddings (for text words) which will be useful in evaluating the phonetic and semantic information carried by these embeddings.

Assume we have the audio embeddings for a set of spoken words $\mathbf{P_W} = \{\mathbf{p}_{\mathbf{w}1}, ..., \mathbf{p}_{\mathbf{w}i}, ..., \mathbf{p}_{\mathbf{w}M}\}$, where $\mathbf{p}_{\mathbf{w}i}$ is the embedding obtained for a spoken word $\mathbf{x}_i$ and M is the total number of distinct spoken words in the audio corpus. On the other hand, assume we have the text embeddings $\mathbf{Q_W} = \{\mathbf{q}_{\mathbf{w}1}, ..., \mathbf{q}_{\mathbf{w}j}, ..., \mathbf{q}_{\mathbf{w}M}\}$, where $\mathbf{q}_{\mathbf{w}j}$ is the embedding of the j-th text word for the M distinct text words. Although the

distributions of $\mathbf{P_W}$ and $\mathbf{Q_W}$ in their respective spaces are not parallel, that is, a specific dimension in the space for $\mathbf{p_w}$ does not necessarily correspond to a specific dimension in the space for $\mathbf{q_w}$, there should exist some consistent relationship between the two distributions. For example, the relationships among the words {France, Paris, Germany} learned from context should be consistent in some way, regardless of whether they are in text or spoken form. So we try to learn a mapping relation between the two spaces. It will be clear below such a mapping relation can be used to evaluate the phonetic and semantic information carried by the audio embeddings.

Mini-Batch Cycle Iterative Closest Point (MBC-ICP) [45] previously proposed as described below is used here. Given two sets of embeddings as mentioned above, $\mathbf{P_W}$ and $\mathbf{Q_W}$, they are first projected to their respective top K principal components by PCA. Let the projected sets of vectors of $\mathbf{P_W}$ and $\mathbf{Q_W}$ be $\mathbf{A}$ and $\mathbf{B}$ respectively. If $\mathbf{P_W}$ can be mapped to the space of $\mathbf{Q_W}$ by an affine transformation, the distributions of $\mathbf{A}$ and $\mathbf{B}$ would be similar after PCA [45].

Then a pair of transformation matrices, $\mathbf{T_{ab}}$ and $\mathbf{T_{ba}}$, is learned, where $\mathbf{T_{ab}}$ transforms a vector $\mathbf{a}$ in $\mathbf{A}$ to the space of $\mathbf{B}$, that is, $\tilde{\mathbf{b}} = \mathbf{T_{ab}a}$, while $\mathbf{T_{ba}}$ maps a vector $\mathbf{b}$ in $\mathbf{B}$ to the space of $\mathbf{A}$. $\mathbf{T_{ab}}$ and $\mathbf{T_{ba}}$ are learned iteratively by the algorithm proposed previously [45].

In our evaluation as mentioned below, labeled pairs of the audio and text embeddings of each word is available, that is, we know $\mathbf{a_i}$ and $\mathbf{b_i}$ for each word $\mathbf{w_i}$. So we can train the transformation matrices $\mathbf{T_{ab}}$ and $\mathbf{T_{ba}}$ using the gradient descent method to minimize the following objective function:

$$L_{trans} = \sum_i \|\mathbf{b_i} - \mathbf{T_{ab}a_i}\|_2^2 + \sum_j \|\mathbf{a_j} - \mathbf{T_{ba}b_j}\|_2^2$$
$$+ \lambda' \sum_i \|\mathbf{a_i} - \mathbf{T_{ba}T_{ab}a_i}\|_2^2 \qquad (5)$$
$$+ \lambda' \sum_j \|\mathbf{b_j} - \mathbf{T_{ab}T_{ba}b_j}\|_2^2.$$

where the last two terms in (5) are cycle-constraints to ensure that both $\mathbf{a_i}$ and $\mathbf{b_j}$ are almost unchanged after transformed to the other space and back. In this way we say the two sets of embeddings are parallelized.

3. EXPERIMENTAL SETUP

3.1. Dataset

We used LibriSpeech [47] as the audio corpus in the experiments, which is a corpus of read speech in English derived from audiobooks. This corpus contains 1000 hours of speech sampled at 16 kHz uttered by 2484 speakers. We used the "clean" and "others" sets with a total of 960 hours, and extracted 39-dim MFCCs as the acoustic features.

Table 1. Top-1 nearest accuracies when parallelizing the different versions of audio and text embeddings for different numbers of pairs of spoken and text words.

		(a)TXT-ph	(b)TXT-(se,1h)	(c)TXT-(se,ph)
1000 pairs	(i)AUD-ph	**0.637**	0.124	0.550
	(ii)AUD-(ph⁻+se)	0.519	0.322	**0.750**
	(iii)AUD-(ph+se)	0.598	0.339	**0.800**
3000 pairs	(i)AUD-ph	**0.465**	0.028	0.279
	(ii)AUD-(ph⁻+se)	**0.330**	0.032	0.254
	(iii)AUD-(ph+se)	**0.395**	0.033	0.313
5000 pairs	(i)AUD-ph	**0.362**	0.012	0.190
	(ii)AUD-(ph⁻+se)	**0.263**	0.022	0.173
	(iii)AUD-(ph+se)	**0.315**	0.023	0.212

Table 2. Top-10 nearest accuracies when parallelizing the different versions of audio and text embeddings for different numbers of pairs of spoken and text words.

		(a)TXT-ph	(b)TXT-(se,1h)	(c)TXT-(se,ph)
1000 pairs	(i)AUD-ph	**0.954**	0.355	0.898
	(ii)AUD-(ph⁻+se)	0.897	0.653	**0.986**
	(iii)AUD-(ph+se)	0.945	0.742	**0.994**
3000 pairs	(i)AUD-ph	**0.854**	0.120	0.654
	(ii)AUD-(ph⁻+se)	**0.758**	0.146	0.671
	(iii)AUD-(ph+se)	**0.809**	0.166	0.752
5000 pairs	(i)AUD-ph	**0.774**	0.050	0.518
	(ii)AUD-(ph⁻+se)	**0.658**	0.109	0.544
	(iii)AUD-(ph+se)	**0.717**	0.111	0.607

3.2. Model Implementation

In Stage 1, The phonetic encoder E_p, speaker encoder E_s and decoder Dec were all 2-layer GRUs with hidden layer size 128, 128 and 256, respectively. The speaker discriminator D_s is a fully-connected feedforward network with 2 hidden layers with size 128. The value of λ we used in L_s in (2) was set to 0.01.

In Stage 2, the two encoders E_{sem} and E_{ctx} were both 2-hidden-layer fully-connected feedforward networks with size 256. The size of embedding vectors was set to be 128. The context window size was 5, and the negative sampling number was 5.

For parallelizing the text and audio embeddings in Subsection 2.3, we projected the embeddings to the top 100 principle components, so the affine transformation matrices were 100×100. The mini-batch size was 200, and λ' in (5) was set to 0.5.

4. EXPERIMENTAL RESULTS

4.1. Evaluation by Parallelizing Audio and Text Embeddings

Each text word corresponds to many audio realizations in spoken form. So we first took the average of the audio embeddings for all those realizations to be the audio embedding for the spoken word considered. In this way, each word has a unique representation in either audio or text form.

We applied three different versions of audio embedding

Table 3. Some examples of top-10 nearest neighbors in AUD-(ph+se) (proposed), AUD-ph (with phonetic structure) and TXT-(se,1h) (with semantics). The words in red are the common words of AUD-(ph+se) and AUD-ph, and the words in bold are the common words of AUD-(ph+se) and TXT-(se,1h).

words	AUD-(ph+se)	AUD-ph	TXT-(se,1h)
owned	own, only, unknown, owner, land, armed, **learned**, homes, **known**, alone	owns, armed, owen, arm, own, only, oughtnt, loaned, ode, owing	visited, introduced, lived, related, **learned**, discovered, met, called, think, **known**
didn't	did, sitting, give, **doesn't**, **don't**, given, hadn't, too, bidden, listen	giving, bidden, given, getting, being, even, ridden, didnt, deane, givin	**don't**, can't, wouldn't, **doesn't**, won't, i'm, you're, shouldn't, think, want

Table 4. Spoken document retrieval performance using two different audio embeddings (AUD-(ph+se) and AUD-ph).

groundtruth	AUD-(ph+se)	AUD-ph
$D_1 + D_2$	17.8%	15.6%
D_2	2.8%	1.8%

(AUD) on the top 1000, 3000 and 5000 words with the highest frequencies in LibriSpeech: (i) phonetic embedding only obtained in Stage 1 in Subsection 2.1 (AUD-ph); (ii) phonetic-and-semantic embedding obtained by Stages 1 and 2 in Subsections 2.1, 2.2, except the speaker characteristics not disentangled (AUD-(ph⁻+se)), or L_s, L_d in (2), (3) not considered; (iii) complete phonetic-and-semantic embedding as proposed in this paper including Stages 1 and 2 (AUD-(ph+se)). So this is for ablation study.

On the other hand, we also obtained three different types of text embedding (TXT) on the same set of top 1000, 3000 and 5000 words. Type (a) Phonetic Text embedding (TXT-ph) considered precise phonetic structure but not context or semantics at all. This was achieved by a well-trained sequence-to-sequence autoencoder encoding the precise phoneme sequence of a word into a latent embedding. Type (b) Semantic Text embedding considered only context or semantics but not phonetic structure at all, and was obtained by a standard skip-gram model using one-hot representations as the input (TXT-(se,1h)). Type (c) Semantic and Phonetic Text embedding (TXT-(se,ph)) considered context or semantics as well as the precise phonetic structure, obtained by a standard skip-gram model but using the Type (a) Phonetic Text embedding (TXT-ph) as the input. So these three types of text embeddings provided the reference embeddings obtained from text and/or phoneme sequences, not disturbed by audio signals at all.

Now we can perform the transformation from the above three versions of audio embeddings (AUD-ph, AUD-(ph⁻+se), AUD-(ph+se)) to the above three types of text embeddings (TXT-ph, TXT-(se,1h), TXT-(se,ph)) by parallelizing the embeddings as described in Subsection 2.3. The evaluation metric used for this parallelizing test is the top-k nearest accuracy. If the audio embedding representation $\mathbf{a_i}$ of a word $\mathbf{w_i}$ is transformed to the text embedding $\mathbf{b_j}$ by $\mathbf{T_{ab}}$, and $\mathbf{b_j}$ is among the top-k nearest neighbors of the text embedding representation $\mathbf{b_i}$ of the same word, this transformation for word $\mathbf{w_i}$ is top-k-accurate. The top-k nearest accuracy is then the

percentage of the words considered which are top-k-accurate.

The results of top-k nearest accuracies for k=1 and 10 are respectively listed in Tables 1 and 2, each for 1000, 3000 and 5000 pairs of spoken and text words.

First look at the top part of Table 1 for top-1 nearest accuracies for 1000 pairs of audio and text embeddings. Since column (a) (TXT-ph) considered precise phonetic structures but not semantics at all, the relatively high accuracies in column (a) for all three versions of audio embedding (i)(ii)(iii) implied the three versions of audio embedding were all rich of phonetic information. But when the semantics were embedded in (ii)(iii) (AUD-(ph⁻+se), AUD-(ph+se)), the phonetic structures were inevitably disturbed (0.519, 0.598 vs 0.637). On the other hand, column (b) (TXT-(se,1h)) considered only semantics but not phonetic structure at all, the relatively lower accuracies implied the three versions of audio embedding did bring some good extent of semantics, except (i) AUD-ph, but obviously weaker than the phonetic information in column (a). Also, the Stage 2 training in rows (ii)(iii) (AUD-(ph⁻+se), AUD-(ph+se)) gave higher accuracies than row (i) (AUD-ph) (0.339, 0.332 vs 0.124 in column (b)), which implied the Stage 2 training was successful. However, column (c) (TXT-(se,ph)) is for the text embedding considering both the semantic and phonetic information, so the two versions of phonetic-and-semantic audio embedding for rows (ii)(iii) had very close distributions (0.750, 0.800 in column (c)), or carried good extent of both semantics and phonetic structure. The above are made clearer by the numbers in bold which are the highest for each row, and the numbers in red which are the highest for each column. It is also clear that the speaker characteristics disentanglement is helpful, since row (iii) for AUD-(ph+se) was always better than row (ii) for AUD-(ph⁻+se).

Similar trends can be observed in the other parts of Table 1 for 3000 and 5000 pairs, except the accuracies were lower, probably because for more pairs the parallelizing transformation became more difficult and less accurate. The only difference is that in these parts column (a) for TXT-ph had the highest accuracies, probably because the goal of semantic embedding for rows (ii)(iii) (AUD-(ph⁻+se), AUD-(ph+se)) was really difficult, and disturbed or even dominated by phonetic structures. Similar trends can be observed in Table 2 for top-10 accuracies, obviously with higher numbers for top-10 as compared to those for top-1 in Table 1.

Table 5. Some retrieval examples of chapters in D_2 using AUD-(ph+se) show the advantage of semantics information in phonetic-and-semantic embeddings. The word in red in each row indicates the word with the highest similarity to the query in the chapter.

(a) query q	(b) title of a book b	(c) chapter	(d) rank	(e) the word with the highest similarity to the query
nations	Myths and Legends of All Nations	Prometheus the Friend of Man	13/5273	...and shall marry the king of that country...
Anne	Anne of Green Gables	Mrs. Rachel Lynde Is Surprised	25/5329	...why the worthy woman finally concluded...
German	In a German Pension	Story 13: A Blaze	22/5232	...through the heavy snow towards the town...
castle	Montezuma's Castle and Other Weird Tales	THE STRANGE POWDER...	3/5141	...what is its history asked doctor Farrington...
baron	Surprising Adventures of Baron Munchausen	Chapter 22	18/5375	...at the palace and having remained in this situation...

In Table 3, we list some examples of top-10 nearest neighbors in AUD-(ph+se) (proposed), AUD-ph (with phonetic structure) and TXT-(se,1h) (with semantics). The words in red are the common words for AUD-(ph+se) and AUD-ph, and the words in bold are the common words of AUD-(ph+se) and TXT-(se,1h). For example, the word "owned" has two common semantically related words "learned" and "known" in the top-10 nearest neighbors of AUD-(ph+se) and TXT-(se,1h). The word "owned" also has three common phonetically similar words "armed", "own" and "only" in the top-10 nearest neighbors of AUD-(ph+se) and AUD-ph. This is even clearer for the function word "didn't". These clearly illustrate the phonetic-and-semantic nature of AUD-(ph+se).

4.2. Results of Spoken Document Retrieval

The goal here is to retrieve not only those spoken documents including the spoken query (e.g. "President Donald Trump") based on the phonetic structures, but those including words semantically related to the query word (e.g. "White House"). Below we show the effectiveness of the phonetic-and-semantc embedding proposed here in this application.

We used the 960 hours of "clean" and "other" parts of LibriSpeech dataset as the target archive for retrieval, which consisted of 1478 audio books with 5466 chapters. Each chapter included 1 to 204 utterances or 5 to 6529 spoken words. In our experiments, the queries were the keywords in the book titles, and the spoken documents were the chapters. We chose 100 queries out of 100 randomly selected book titles, and our goal was to retrieve query-relevant documents. For each query q, we defined two sets of query-relevant documents: The first set D_1^q consisted of chapters which included the query q. The second set D_2^q consisted of chapters whose content didn't contain q, but these chapters belonged to books whose titles contain q (so we assume these chapters are semantically related to q). Obviously D_1^q and D_2^q were mutually exclusive, and D_2^q were the target for semantic retrieval, but couldn't be retrieved based on the phonetic structures only.

For each query q and each document d, the relevance score of d with respect to q, $s(q,d)$, is defined as follows:

$$s(q,d) = \max_{w \text{ in } d} -\|R(w) - R(q)\|_2, \tag{6}$$

where $R(w)$ is the audio embedding of a word w in d. So (6) indicates the documents d were ranked by the minimum distance between a word w in d and the query q. We used mean average precision (MAP) as the evaluation metric for the spoken document retrieval test.

We compared the retrieval results with two versions of audio embedding: AUD-(ph+se) and AUD-ph. The results are listed in Table 4 for two definitions of groundtruth for the query-relevant documents: the union of D_1 and D_2 and D_2 alone. As can be found from this table, AUD-(ph+se) offered better retrieval performance than AUD-ph in both rows. Note that those chapters in D_2 in the second row of the table did not include the query q, so couldn't be well retrieved using phonetic embedding alone. That is why the phonetic-and-semantic embedding proposed here can help.

In Table 5, we list some chapters in D_2 retrieved using AUD-(ph+se) embeddings to illustrate the advantage of the phonetic-and-semantic embeddings. In this table, column (a) is the query q, column (b) is the title of a book b which had chapters in D_2^q, column (c) is a certain chapter chp in b, column (d) is the rank of chp out of all chapters whose content didn't contain q, and column (e) is a part of the content in chp where the word in red is the word in chp with the highest similarity to q. For example, in the first row for the query "nations", the chapter "Prometheus the Friend of Man" of the book titled "Myths and Legends of All Nations" is in $D_2^{nations}$. The word "nations" is not in the content of this chapter. However, because the word "king" semantically related to "nations" is in the content, this chapter was ranked the 13th among all chapters whose content didn't contain the word "nations". This clearly verified why the semantics in the phonetic-and-semantic embeddings can remarkably improve the performance of spoken content retrieval.

5. CONCLUSIONS AND FUTURE WORK

In this paper we propose a framework to embed spoken words into vector representations carrying both the phonetic structure and semantics of the word. This is intrinsically challenging because the phonetic structure and the semantics of spoken words inevitably disturbs each other. But this phonetic-and-semantic embedding nature is desired and attractive, for example in the application task of spoken document retrieval. A parallelizing transformation between the audio and text embeddings is also proposed to evaluate whether such a goal is achieved.

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] Jeffrey Pennington, Richard Socher, and Christopher Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[4] Lifu Tu, Kevin Gimpel, and Karen Livescu, "Learning to embed words in context for syntactic tasks," *arXiv preprint arXiv:1706.02807*, 2017.

[5] Xiang Yu and Ngoc Thang Vu, "Character composition model with convolutional neural networks for dependency parsing on morphologically rich languages," *arXiv preprint arXiv:1705.10814*, 2017.

[6] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[7] Barbara Plank, Anders Søgaard, and Yoav Goldberg, "Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss," *arXiv preprint arXiv:1604.05529*, 2016.

[8] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush, "Character-aware neural language models.," in *AAAI*, 2016, pp. 2741–2749.

[9] Miguel Ballesteros, Chris Dyer, and Noah A Smith, "Improved transition-based parsing by modeling characters instead of words with lstms," *arXiv preprint arXiv:1508.00657*, 2015.

[10] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective approaches to attention-based neural machine translation," *CoRR*, vol. abs/1508.04025, 2015.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[12] 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.

[13] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko, "Sequence to sequence - video to text," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 4534–4542.

[14] Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer, "Neural AMR: sequence-to-sequence models for parsing and generation," *CoRR*, vol. abs/1704.08381, 2017.

[15] Wanjia He, Weiran Wang, and Karen Livescu, "Multiview recurrent neural acoustic word embeddings," *arXiv preprint arXiv:1611.04496*, 2016.

[16] Shane Settle and Karen Livescu, "Discriminative acoustic word embeddings: Recurrent neural network-based approaches," *arXiv preprint arXiv:1611.02550*, 2016.

[17] 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," *arXiv preprint arXiv:1603.00982*, 2016.

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

[19] Samy Bengio and Georg Heigold, "Word embeddings for speech recognition," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

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

[21] Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," *arXiv preprint arXiv:1704.01631*, 2017.

[22] Shane Settle, Keith Levin, Herman Kamper, and Karen Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," *arXiv preprint arXiv:1706.03818*, 2017.

[23] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[24] Yu-An Chung and James R. Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," *CoRR*, vol. abs/1803.08976, 2018.

[25] Yi-Chen Chen, Chia-Hao Shen, Sung-Feng Huang, and Hung-yi Lee, "Towards unsupervised automatic speech recognition trained by unaligned speech and text only," *arXiv preprint arXiv:1803.10952*, 2018.

[26] Yu-Hsuan Wang, Hung-yi Lee, and Lin-shan Lee, "Segmental audio word2vec: Representing utterances as se-

quences of vectors with applications in spoken term detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference.* IEEE, 2018.

[27] Trang Tran, Shubham Toshniwal, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Mari Ostendorf, "Parsing speech: A neural approach to integrating lexical and acoustic-prosodic information," *arXiv preprint arXiv:1704.07287*, 2017.

[28] Hao Tang, Liang Lu, Lingpeng Kong, Kevin Gimpel, Karen Livescu, Chris Dyer, Noah A Smith, and Steve Renals, "End-to-end neural segmental models for speech recognition," *arXiv preprint arXiv:1708.00531*, 2017.

[29] Herman Kamper, Karen Livescu, and Sharon Goldwater, "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE.* IEEE, 2017, pp. 719–726.

[30] 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.

[31] Tejas D. Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum, "Deep convolutional inverse graphics network," *CoRR*, vol. abs/1503.03167, 2015.

[32] Wei-Ning Hsu, Yu Zhang, and James R. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," *CoRR*, vol. abs/1709.07902, 2017.

[33] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 2172–2180. Curran Associates, Inc., 2016.

[34] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," *CoRR*, 2016.

[35] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan, "Domain separation networks," *CoRR*, vol. abs/1608.06019, 2016.

[36] Zhong Meng, Zhuo Chen, Vadim Mazalov, Jinyu Li, and Yifan Gong, "Unsupervised adaptation with domain separation networks for robust speech recognition," *CoRR*, vol. abs/1711.08010, 2017.

[37] Aren Jansen, Manoj Plakal, Ratheet Pandya, Daniel PW Ellis, Shawn Hershey, Jiayang Liu, R Channing Moore, and Rif A Saurous, "Unsupervised learn-

ing of semantic audio representations," *arXiv preprint arXiv:1711.02209*, 2017.

[38] Aren Jansen, Manoj Plakal, Ratheet Pandya, Daniel PW Ellis, Shawn Hershey, Jiayang Liu, R Channing Moore, and Rif A Saurous, "Towards learning semantic audio representations from unlabeled data," *signal*, vol. 2, no. 3, pp. 7–11, 2017.

[39] Lin-shan Lee, James Glass, Hung-yi Lee, and Chun-an Chan, "Spoken content retrievalbeyond cascading speech recognition with text retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1389–1420, 2015.

[40] Berlin Chen, Kuan-Yu Chen, Pei-Ning Chen, and Yi-Wen Chen, "Spoken document retrieval with unsupervised query modeling techniques," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2602–2612, 2012.

[41] Pei-Ning Chen, Kuan-Yu Chen, and Berlin Chen, "Leveraging relevance cues for improved spoken document retrieval," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[42] Hung-yi Lee, Yun-Chiao Li, Cheng-Tao Chung, and Lin-shan Lee, "Enhancing query expansion for semantic retrieval of spoken content with automatically discovered acoustic patterns," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on.* IEEE, 2013, pp. 8297–8301.

[43] Tsung-Hsien Wen, Hung-Yi Lee, Pei-hao Su, and Lin-Shan Lee, "Interactive spoken content retrieval by extended query model and continuous state space markov decision process," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on.* IEEE, 2013, pp. 8510–8514.

[44] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou, "Word translation without parallel data," *arXiv preprint arXiv:1710.04087*, 2017.

[45] Yedid Hoshen and Lior Wolf, "An iterative closest point method for unsupervised word translation," *arXiv preprint arXiv:1801.06126*, 2018.

[46] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.

[47] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 5206–5210.