Language Transfer of Audio Word2Vec: Learning Audio Segment Representations without Target Language Data

Abstract

Audio Word2Vec offers vector representations of fixed dimensionality for variable-length audio segments using Sequence-to-sequence Autoencoder (SA). These vector representations are shown to describe the sequential phonetic structures of the audio segments to a good degree, with real world applications such as query-by-example Spoken Term Detection (STD). This paper examines the capability of language transfer of Audio Word2Vec. We train SA from one language (source language) and use it to extract the vector representation of the audio segments of another language (target language). We found that SA can still catch phonetic structure from the audio segments of the target language if the source and target languages are similar. In query-by-example STD, we obtain the vector representations from the SA learned from a large amount of source language data, and found them surpass the representations from naive encoder and SA directly learned from a small amount of target language data. The result shows that it is possible to learn Audio Word2Vec model from high-resource languages and use it on low-resource languages. This further expands the usability of Audio Word2Vec.

Introduction

Embedding audio word segments into fixed-length vectors has many useful applications in natural language processing such as speaker identification BIBREF0 , audio emotion classification BIBREF1 , and spoken term detection (STD) BIBREF2 , BIBREF3 , BIBREF4 . In these applications, audio segments are usually represented as feature vectors to be applied to a standard classifiers which determines the speaker's identification, emotion or whether the input queries are included. By representing the audio segments in fixed-length vectors instead of using the original segments in variable lengths, we can

reduce the effort for indexing, accelerate the speed of calculation, and improve the efficiency for the retrieval task BIBREF5 , BIBREF6 , BIBREF7 .

Recently, deep learning has been used for encoding acoustic information into vectors BIBREF8 , BIBREF9 , BIBREF10 . Existing works have shown that it is possible to transform audio word segments into fixed dimensional vectors. The transformation successfully produces vector space where word audio segments with similar phonetic structures are closely located. In BIBREF10 , the authors train a Siamese convolutional neural network with side information to obtain embeddings that separate same-word pairs and different-word pairs. Human annotated data is required under this supervised learning scenario. Besides supervised approaches BIBREF11 , BIBREF10 , BIBREF12 , BIBREF13 , unsupervised approaches are also proposed to reduce the annotation effort BIBREF14 . As for the unsupervised learning for the audio embedding, LSTM-based sequence-to-sequence autoencoder demonstrates a promising result BIBREF14 . The model is trained to minimize the reconstruction error of the input audio sequence and then provides the embedding, namely Audio Word2Vec, from its bottleneck layer. This is done without any annotation effort.

Although deep learning approaches have produced satisfactory result, the data-hungry nature of the deep model makes it hard to produce the same performance with low-resource data. Both supervised and unsupervised approaches assume that a large amount of audio data of the target language is available. A question arises whether it is possible to transfer the Audio Word2Vec model learned from a high-resource language into a model targeted at a low-resource language. While this problem is not yet to be fully examined in Audio Word2Vec, works in neural machine translation (NMT) successfully transfer the model learned on high-resource languages to low-resource languages. In BIBREF15 , BIBREF16 , the authors first train a source model with high-resource language pair. The source model is used to initialize the target model which is then trained by low-resource language pairs.

For audio, all languages are uttered by human beings with a similar vocal tract structure, and therefore share some common acoustic patterns. This fact implies that knowledge obtained from one spoken language can be transferred onto other languages. This paper verifies that sequence-to-sequence autoencoder is not only able to transform audio word segments into fixed-length vectors, the model is also transferable to the languages it has never heard before. We also demonstrate its promising applications with a query-by-example spoken term detection (STD) experiment. In the query-by-example STD experiment, even without tunning with partial low-resource language segments, the autoencoder can still produce high-quality vectors.

Audio Word2Vec

The goal for Audio Word2Vec model is to identify the phonetic patterns in acoustic feature sequences such as MFCCs. Given a sequence INLINEFORM0 where INLINEFORM1 is the acoustic feature at time INLINEFORM2 , and INLINEFORM3 is the length, Audio Word2Vec transforms the features into fixed-length vector INLINEFORM4 with dimension INLINEFORM5 based on the phonetic structure.

RNN Encoder-Decoder Network

Recurrent Neural Networks (RNNs) has shown great success in many NLP tasks with its capability of capturing sequential information. The hidden neurons form a directed cycle and perform the same task for every element in a sequence. Given a sequence INLINEFORM0 , RNN updates its hidden state INLINEFORM1 according to the current input INLINEFORM2 and the previous INLINEFORM3 . The hidden state INLINEFORM4 acts as an internal memory at time INLINEFORM5 that enables the network to capture dynamic temporal information, and also allows the network to process sequences of variable length. However, in practice, RNN does not seem to learn long-term dependencies due to the vanishing gradient problem BIBREF17 , BIBREF18 . To conquer such difficulties, LSTM BIBREF19 and GRU

BIBREF20 , BIBREF21 were proposed. While LSTM achieves many amazing results BIBREF22 , BIBREF23 , BIBREF24 , BIBREF25 , BIBREF26 , BIBREF20 , BIBREF27 , the relative new GRU performs just as well with less parameters and training effort BIBREF28 , BIBREF29 , BIBREF30 , BIBREF31 .

RNN Encoder-Decoder BIBREF26 , BIBREF32 consists of an Encoder RNN and a Decoder RNN.The Encoder RNN reads the input sequence INLINEFORM0 sequentially and the hidden state INLINEFORM1 of the RNN is updated accordingly. After the last symbol INLINEFORM2 is processed, the hidden state INLINEFORM3 is interpreted as the learned representation of the whole input sequence. Then, by taking INLINEFORM4 as input, the Decoder RNN generates the output sequence INLINEFORM5 sequentially, where INLINEFORM6 and INLINEFORM7 can be different, or the length of INLINEFORM8 and INLINEFORM9 can be different. Such RNN Encoder-Decoder framework is able to handle variable-length input. Although there may exist a considerable time lag between the input symbols and their corresponding output symbols, LSTM and GRU are able to handle such situation well due to their powerfulness in modeling long-term dependencies.

Sequence-to-sequence Autoencoder

Figure FIGREF3 depicts the structure of Sequence-to-sequence Autoencoder (INLINEFORM0), which integrates the RNN Encoder-Decoder framework with Autoencoder for unsupervised learning of audio segment representations. INLINEFORM1 consists of an Encoder RNN (the left part of Figure FIGREF3) and a RNN Decoder (the right part). Given an audio segment represented as an acoustic feature sequence INLINEFORM2 of any length INLINEFORM3 , the RNN Encoder reads each acoustic feature INLINEFORM4 sequentially and the hidden state INLINEFORM5 is updated accordingly. After the last acoustic feature INLINEFORM6 has been read and processed, the hidden state INLINEFORM7 of the Encoder RNN is viewed as the learned representation INLINEFORM8 of the input sequence (the purple

block in Figure FIGREF3). The Decoder RNN takes INLINEFORM9 as the initial state of the RNN cell, and generates a output INLINEFORM10 . Instead of taking INLINEFORM11 as the input of the next time step, a zero vector is fed in as input to generate INLINEFORM12 , and so on. This structure is called the historyless decoder. Based on the principles of Autoencoder BIBREF33 , BIBREF34 , the target of the output sequence INLINEFORM13 is the input sequence INLINEFORM14 . In other words, the RNN Encoder and Decoder are jointly trained by minimizing the reconstruction error, measured by the general mean squared error INLINEFORM15 . Because the input sequence is taken as the learning target, the training process does not need any labeled data. The fixed-length vector representation INLINEFORM16 will be a meaningful representation for the input audio segment INLINEFORM17 because the whole input sequence INLINEFORM18 can be reconstructed from INLINEFORM19 by the RNN Decoder.

Using historyless decoder is critical here. We found out that the performance in the STD experiment was undermined despite the low reconstruction error. This shows that the vector representations learned from INLINEFORM0 do not include useful information. This might be caused by a strong decoder as the model focuses less on including more information into the vector representation. We eventually solved the problem by using a historyless decoder. Historyless decoder is a weakened decoder. The input of the decoder is removed, and this forces the model to rely more on the vector representation. The historyless decoder is also used in recent NLP works BIBREF35 , BIBREF36 , BIBREF37 .

Language Transfer

In the study of linguistic, scholars define a set of universal phonetic rules which describe how sounds are commonly organized across different languages. Actually, in real life, we often find languages sharing similar phonemes especially the ones spoken in nearby regions. These facts implies that when switching target languages, we do not need to learn the new audio pattern from scratch due to the transferability in spoken languages. Language transfer has shown to be helpful in STD BIBREF38 , BIBREF39 ,

BIBREF40 , BIBREF41 , BIBREF42 , BIBREF43 , BIBREF44 , BIBREF45 . In this paper, we focus on studying the capability of transfer learning of Audio Word2Vec.

In the proposed approach, we first train an INLINEFORM0 using the high-resource source language, as shown in the upper part of Fig. FIGREF4 , and then the encoder is used to transform the audio segment of a low-resource target language. It is also possible to fine-tune the parameters of INLINEFORM1 with the target language. In the following experiments, we found that in some cases the STD performance of the encoder without fine-tuning with the low-resource target language can be as good as the one with fine-tuning.

An Example Application: Query-by-example STD

The audio segment representation INLINEFORM0 learned in the last section can be applied in many possible scenarios. Here in the preliminary tests we consider the unsupervised query-by-example STD, whose target is to locate the occurrence regions of the input spoken query term in a large spoken archive without speech recognition. Figure FIGREF5 shows how the representation INLINEFORM1 proposed here can be easily used in this task. This approach is inspired from the previous work BIBREF6 , but completely different in the ways to represent the audio segments. In the upper half of Figure FIGREF5 , the audio archive are segmented based on word boundaries into variable-length sequences, and then the system exploits the trained RNN encoder in Figure FIGREF3 to encode these audio segments into fixed-length vectors. All these are done off-line. In the lower left corner of Figure FIGREF5 , when a spoken query is entered, the input spoken query is similarly encoded by the same RNN encoder into a vector. The system then returns a list of audio segments in the archive ranked according to the cosine similarities evaluated between the vector representation of the query and those of all segments in the archive. Note that the computation requirements for the online process here are extremely low.

Experimental Setup

Here we provide detail of our experiment including the dataset, model setup, and the baseline model.

Dataset

Two corpora across five languages were used in the experiment. One of the corpora we used is LibriSpeech corpus BIBREF46 (English). In this 960-hour English dataset, 2.2 million audio word segments were used for training while the other 250 thousand segments were used as the database to be retrieved in STD and 1 thousand segments as spoken queries. In Section 6.1, we further sampled 20 thousand segments from 250 thousand segments to form a small database to investigate the influence of database size. English served as the high-resource source language for model pre-training.

The other dataset is the GlobalPhone corpus BIBREF47 , which includes French (FRE), German (GER), Czech (CZE), and Spanish (ESP). The four languages from GlobalPhone were used as the low-resource target languages. In Section 6.2, 20 thousand segments for each language were used to calculate the average cosine similarity. For the experiments of STD, the 20 thousands segments served as the database to be retrieved, and the other 1 thousand used for query and 4 thousand for fine-tuning.

MFCCs of 39-dim were used as the acoustic features. The length of the input sequence was limited to 50 frames. All datasets were segmented according to the word boundaries obtained by forced alignment with respect to the reference transcriptions. Although the oracle word boundaries were used here for the query-by-example STD in the preliminary tests, the comparison in the following experiment was fair since all approaches used the same segmentation. Mean average precision (MAP) was used as the evaluation measure for query-by-example STD.

Proposed Model: Sequence Autoencoder (SASA)

Both the proposed model (INLINEFORM0) and baseline model (INLINEFORM1 , described in the next subsection) were implemented with Tensorflow. The network structure and the hyper parameters were set as below:

Both RNN Encoder and Decoder consisted one hidden layer of GRU cells BIBREF20 , BIBREF21 . The number of units in the layer would be discussed in the experiment.

The networks were trained by SGD without momentum. The initial learning rate was 1 and decayed with a factor of 0.95 every 500 batches.

Baseline: Naive Encoder (NENE)

We used naive encoder (INLINEFORM0) as the baseline approach. In this encoder, the input acoustic feature sequence INLINEFORM1 = (INLINEFORM2), where INLINEFORM3 was the 39-dimension MFCC feature vector at time t, were divided into INLINEFORM4 partitions with roughly equal length INLINEFORM5 . Then, we averaged each partition into a single 39-dimension vector, and finally got the vector representation through concatenating the INLINEFORM6 average vectors sequentially into a vector representation of dimensionality INLINEFORM7 . Although INLINEFORM8 is simple, similar approaches have been used in STD and achieved successful results BIBREF2 , BIBREF3 , BIBREF4 .

Experiments

In this section, we first examine how changing the hidden layer size of the RNN Encoder/Decoder, the dimension of Audio Word2Vec, affects the MAP performance of query-by-example STD (Section 6.1).

After obtaining the best hidden layer size, we analyze the transferability of the Audio Word2Vec by comparing the cosine similarity of the learned representations to phoneme sequence edit distance (Section 6.2) . Visualization of multiple word pairs in different target languages is also provided (Section 6.3). Last but not least, we performed the query-by-example STD on target languages (Section 6.4). These experiments together verify that INLINEFORM0 is capable of extracting common phonetic structure in human language and thus is transferable to various languages.

Analysis on Dimension of Audio Word2Vector

Before evaluating the language transfer result, we first experimented on the primary INLINEFORM0 model in the source language (English). The results are shown in Fig. FIGREF12 . Here we compare the representations of INLINEFORM1 and INLINEFORM2 . Furthermore, we examined the influence of the dimension of Audio Word2Vector in terms of MAP. We also compared the MAP results on large testing database (250K segments) and small database (20K).

In Fig. FIGREF12 , we varied the dimension of Audio Word2Vector as 100, 200, 400, 600, 800 and 1000. To match up the dimensionality with INLINEFORM0 , we tested INLINEFORM1 with dimensionality 117, 234, 390, 585, 819, 1014 (INLINEFORM2) and denoted them by INLINEFORM3 where INLINEFORM4 is the dimensionality. INLINEFORM5 get higher MAP values than INLINEFORM6 no matter the vector dimension and the size of database. The highest MAP score INLINEFORM7 can achieve is 0.881 (INLINEFORM8 on small database), while the highest score of the INLINEFORM9 model is 0.490 (INLINEFORM10 on small database). The size of database has large influence on the results. The MAP scores of the two models both drop in the large database. For example, INLINEFORM11 drops from 0.490 to 0.158, decaying by 68%, and the performance of INLINEFORM12 drops from 0.881 to 0.317, decaying by 64%. As shown in Fig. FIGREF12 , larger dimensionality does not imply better performance in query-by-example STD. The MAP scores gradually improve until reaching the dimensionality of 400 in

INLINEFORM13 and 234 in INLINEFORM14 , and start to decrease as the dimension increases. In the rest of the experiments, we would use 400 GRU units in the INLINEFORM15 hidden layer, and set INLINEFORM16 (INLINEFORM17).

Analysis of Language Transfer

To evaluate the quality of language transfer, we trained the Audio Word2Vec model by INLINEFORM0 from the source language, English, and applied it on different target languages, French (FRE), German (GER), Czech (CZE), and Spanish (ESP). We computed the average cosine similarity of the vector representations for each pair of the audio segments in the retrieval database of the target languages (20K segments for each language), and compare it with the phoneme sequence edit distance (PSED). The average and variance (the length of the black line on each bar) of the cosine similarity for groups of pairs clustered by the phoneme sequence edit distances (PSED) between the two words are shown in Fig. FIGREF14 . For comparison, we also provide the results obtained from the English retrieval database (250K segments), where the segments were not seen by the model in training procedure.

In Fig. FIGREF14 , the cosine similarities of the segment pairs get smaller as the edit distances increase, and the trend is observed in all languages. The gap between each edit distance groups, i.e. (0,1), (1,2), (2,3), (3,4), is obvious. This means that INLINEFORM0 learned from English can successfully encode the sequential phonetic structures into fixed-length vector for the target languages to some good extend even though it has never seen any audio data of the target languages. Another interesting fact is the corresponding variance between languages. In the source language, English, the variances of the five edit distance groups are fixed at 0.030, which means that the cosine similarity in each edit distance group is centralized. However, the variances of the groups in the target languages vary. In French and German, the variance grows from 0.030 to 0.060 as the edit distance increases from 0 to 4. For Czech/Spanish, the variance starts at a larger value of 0.040/0.050 and increases to 0.050/0.073. We suspect that the

fluctuating variance is related to the similarity between languages. English, German and French are more similar compared with Czech and Spanish. Among the four target languages, German has the highest lexical similarity with English (0.60) and the second highest is French (0.27), while for Czech and Spanish, the lexical similarity scores is 0 BIBREF48 .

Visualization

In order to further investigate the performance of INLINEFORM0 , we visualize the vector representation of two sets of word pairs differing by only one phoneme from French and German as below:

French Word Pairs: (parler, parlons), (noter,notons), (rappeler, rappelons), (utiliser, utilisons)

German Word Pairs: (tag, tage), (spiel, spiele), (wenig, wenige), (angriff, angriffe)

To show the vector representations in Fig. FIGREF18 , we first obtained the mean value of representations for the audio segments of a specific word, denoted by INLINEFORM0 (word). Then the average representation INLINEFORM1 was projected from 400-dimensional to 2-dimensional using PCA BIBREF49 . The result of the difference vector from each word pair, e.g. INLINEFORM2 (parlons) - INLINEFORM3 (parler), is shown. Although the representations for French and German word audio segments were extracted from the model trained by English audio word segments and never heard any French and German, the direction and magnitude of the different vectors are coherent. In Fig. FIGREF18 , INLINEFORM4 (parlons) - INLINEFORM5 (parler) is close to INLINEFORM6 (utilison) - INLINEFORM7 (utiliser); and INLINEFORM8 (tage) - INLINEFORM9 (tag) is close to INLINEFORM10 (wenige) - INLINEFORM11 (wenig) in Fig. FIGREF18 .

Language Transferring on STD

Besides analyzing the cosine similarity of the learned representations, we also apply them to the query-by-example STD task. Here we compare the retrieval performance in MAP of INLINEFORM0 with different levels of accessibility to the low-resource target language along with two baseline models, INLINEFORM1 and INLINEFORM2 trained purely by the target languages. For the four target languages, the total available amount of audio word segments in the training set were 4 thousands for each language. In Table TABREF20 , we took different partitions of the target language training sets to fine tune the INLINEFORM3 pretrained by the source languages. The amount of audio word segments in these partitions are: 1K, 2K, 3K, 4K, and 0, which means no fine-tuning.

From Table TABREF20 , INLINEFORM0 trained by source language generally outperforms the INLINEFORM1 trained by the limited amount of target language (" INLINEFORM2 No Transfer"), proving that with enough audio segments, INLINEFORM3 can identify and encode universal phonetic structure. Comparing with NE, INLINEFORM4 surpasses INLINEFORM5 in German and French even without fine-tuning, whereas in Czech, INLINEFORM6 also achieves better score than INLINEFORM7 with fine-tuning. However, in Spanish, INLINEFORM8 achieved a MAP score of 0.13 with fine-tuning, slightly lower than 0.17 obtained by INLINEFORM9 . Back to Fig. FIGREF14 , the gap between phoneme sequence edit distances 2 and 3 in Spanish is smaller than other languages. Also, as discussed earlier in Section 6.2, the variance in Spanish is also bigger. The smaller gap and bigger variance together indicate that the model is weaker on Spanish at identifying audio segments of different words and thus affects the MAP performance in Spanish.

Conclusion and Future Work

In this paper, we verify the capability of language transfer of Audio Word2Vec using

Sequence-to-sequence Autoencoer (INLINEFORM0). We demonstrate that INLINEFORM1 can learn the sequential phonetic structure commonly appearing in human language and thus make it possible to apply an Audio Word2Vec model learned from high-resource language to low-resource languages. The capability of language transfer in Audio Word2Vec is beneficial to many real world applications, for example, the query-by-example STD shown in this work. For the future work, we are examining the performance of the transferred system in other application scenarios, and exploring the performance of Audio Word2Vec under automatic segmentation.