# Acoustic embeddings for speech recognition

**Utkarsh Simha**
University of California, San Diego
usimha@ucsd.edu

## Abstract

The paper investigates different approaches for generating acoustic embeddings for the task of speech recognition. We extend the previous work (Bengio and Heigold, 2014) to include learning from the context of sentences in a spoken sentence and also to attempt learning at the phoneme level as in (Tsvetkov et al., 2016). We propose different models to train these acoustic embeddings which can then be used in speech recognition systems by means of a decoder.

## 1 Introduction

Vector representations have been used successfully for text, acoustic signals, images, graphs etc. to generate embeddings which are used in various applications. These representations capture information and structure in the underlying raw data which can further be exploited by models to learn specific functions on the data. Embeddings also help to reduce the dimensionality of the data, because, though the input might lie in a higher dimension, the information that we require might actually lie on a low-dimensional manifold.

Traditional automatic speech recognition(ASR) techniques rely on extracting features from the audio signal using different feature extraction techniques and further use a hidden markov model to capture temporal dependencies in the speech signal (Gales and Young, 2007). Due to the advent of neural networks and its capability to learn internal representations, there have been many neural network models that try to learn mapping functions from the features extracted to the states in the hidden markov model (Dahl et al., 2012) and also models that perform end-to-end speech recognition on the extracted features (Graves et al., 2013)

Recently, there has been significant interest in extracting features from the audio signal by means of neural network models and representing these in an embedding form which is further processed to transcribe into sentences. *[citation required]* Using neural network models for feature extraction helps capture diverse features in the data and makes the system more end-to-end eliminating the need for external feature engineering and extraction. The use of acoustic embedding to augment automatic speech recognition engines can help replace language models and will aid in discerning between homophones (words that sound similar) thus increasing the performance on speech recognition tasks.

In this paper we investigate different approaches to generate acoustic embeddings at a phoneme-level, word-level and a sentence-level. We identify previous work in these areas, address their limitations and try to provide solutions to these problems while aiming to improve performance on the task of automatic speech recognition. However, one of the major drawbacks of using acoustic embeddings for speech recognition is the need to segment the acoustic signal based on phone or word for training the embeddings. We shall try tackling this problem later on.

## 2 Related work

### 2.1 Phoneme embeddings

The inspiration to use phoneme embeddings is drawn from (Tsvetkov et al., 2016) where they are applied to the tasks of lexical borrowing and speech synthesis. There has been previous work of using phoneme embeddings in speech synthesis (Li et al., 2016) but not many extensions to using them in speech recognition. Although (Synnaeve et al., 2014) build phonetic embeddings for the purpose of ASR, no experiments were conducted

to show performance of their model on the task. Phonemes capture a lot of acoustic information in speech signals as opposed to words which hold semantic and syntactic information.

### 2.2 Word embeddings

Word embeddings are used extensively in many natural language processing tasks such as language modelling (Mikolov et al., 2013). There have been previous work which apply word embeddings to speech recognition (Bengio and Heigold, 2014), (Settle and Livescu, 2016), (Kamper et al., 2016), but most of these approaches don't account for the context surrounding the words and are susceptible to grammatical errors due to the lack of a language model. Despite a few drawbacks, word embeddings have been very effective in reducing model complexity and improving performance for speech recognition systems.

### 2.3 Sentence embeddings

Sentence level embeddings capture context among words and are capable of representing sentences as vectors. First introduced by (Kiros et al., 2015) it has been used in many applications. As speech transcripts consist of sentences, sentence embeddings would definitely provide a richer acoustic embedding capturing word context and meaning. (Chung et al., 2016) have used a denoising sequence-to-sequence autoencoder to generate an embedding for the entire acoustic signal. As this is an unsupervised approach, they do not exploit any information of the transcripts from the labeled data to model their embeddings.

## 3 Approach

The approach builds on the aforementioned previous work and aims at tackling their limitations. We divide this section into different approaches followed for phoneme-level embeddings, word-level embeddings and sentence-level embeddings. For each model, we consider an input $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, $\mathbf{x} \in \mathbb{R}^{n \times d_1}$ where each $x_i$ represents the corresponding granular segmented embedding of the transcribed text and $d_1$ represents the embedding dimension. The corresponding acoustic embedding to be modelled is represented as $\mathbf{a} = \{a_1, a_2, \ldots, a_n\}$, $\mathbf{a} \in \mathbb{R}^{n \times d_2}$ where each $a_i$ represents the corresponding segmented frame embedded in $d_2$ vector space.

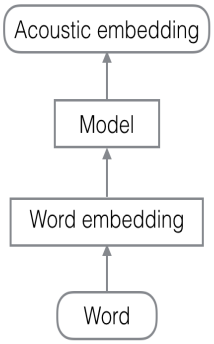


Figure 1: Acoustic word embeddings model. The model tries to compute a mapping function between semantic word embedding space and acoustic word embedding space

### 3.1 Word embeddings

For generating acoustic word embeddings, we propose a method similar to the one in (Bengio and Heigold, 2014). Instead of multiple models that would try to augment the acoustic embedding with a word embedding, we use a modification of an autoencoder (Vincent et al., 2008) where we feed in word embeddings and try to reconstruct the acoustic embedding obtained for the word as shown in Fig. 1. This differs from the traditional autoencoder where we model the acoustic embedding as the output instead of the input itself. [1] We are looking to minimize the loss between the predicted distribution $\mathbf{y}$ and the true distribution $\hat{\mathbf{y}}$: $\min \mathbf{L}(\mathbf{y}, \hat{\mathbf{y}})$ where $\mathbf{L}$ can either be the squared error or the KL-Divergence. The iniital word embeddings $\mathbf{x}$ is transformed through a deep autoencoder which learns a mapping function from semantic word space to acoustic word space. Thus the last layer of the autoencoder captures the words in an embedding space where words that sound similar are mapped near each other. Thus we have: $\mathbf{e} = \mathcal{F}(\mathbf{x})$ where $\mathcal{F}$ is a composite function which represents non-linearities across different layers of the autoencoder and $\mathbf{e}$ represents the acoustic embedding we are interested in. The main drawback of this method is that it requires speech data that are segmented according to their word boundaries. The TIMIT dataset mentioned in Section 4 consists of transcripts with the time stamps for each word segmentation.

[1] From now on when we reference autoencoder, we mean our modified version

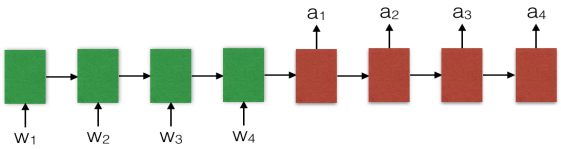


Figure 2: Sequence to sequence model for sentence-level acoustic embedding

### 3.2 Phoneme embeddings

We propose a phoneme level embedding model which is used to generate vector representations of the acoustic signal corresponding to the phonemes. The aim is to generate these representations such that similar sounding phones will be nearby in the embedding space. A simpler version of the polyglot model used in (Tsvetkov et al., 2016) can be used to generate phone-level embeddings. The phonemes are embedded using a deep-LSTM to generate $\mathbf{g} = \mathcal{G}(\mathbf{x})$ where $\mathcal{G}$ represents the embedding model. The phone-level embedding is then modelled using an autoencoder to predict the frame corresponding to the phone-level segmentation of the speech signal $\mathbf{y} = \mathcal{F}(\mathbf{g})$ where $\mathcal{F}$ represents the autoencoder that models the mapping function from phone-embedding space to acoustic embedding space.

### 3.3 Sentence embeddings

One of the approaches for acoustic embeddings is to look at sentence-level embeddings. Sentences capture word context and thus can help in discerning homophones using the context surrounding a word. Given two similar words *birth* and *berth*, the context in which it occurs in a sentence can be used to help disambiguate the correct transcription.

### 3.4 Sequence to sequence model

Sequence to sequence learning (Sutskever et al., 2014) have proven to be useful in learning useful representations for machine translation tasks where we want to learn a function that maps a sequence from one domain to a sequence in another. Thus, we can use this model to again learn a mapping from a sentence embedding space to an acoustic embedding space as shown in Fig. 2. An initial sentence embedding $\mathbf{s}$ is encoded using a deep-LSTM encoder. This is then fed into a decoder which then predicts the corresponding frames of the acoustic features extracted from the raw speech signal (discussed in Section 4.2).

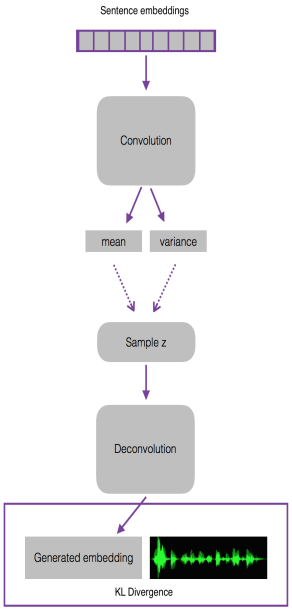


Figure 3: Convolution-deconvolution model

### 3.5 Negative sampling model

An alternative to this would be to use a method similar to (Bengio and Heigold, 2014) but for sentence embeddings. Each sentence is mapped into embedding space using a deep neural network model. This is done for the correct transcription of the corresponding speech signal and also for a randomly chosen negative sample. Preferably, these negative samples should ideally be sentences that sound similar as done in (Lazaridou et al., 2015). This will help the model to differentiate at a more fine-grained level in the embedding space. Thus the loss function for this approach would be $\mathbf{L} = \max(0, m - \text{sim}(\mathbf{e}, \mathbf{s}^+), \text{sim}(\mathbf{e}, \mathbf{s}^-))$ where $m$ is a margin parameter often set to 1, $\mathbf{e}$ is the acoustic embedding generated as described in Section 4.2, $\mathbf{s}^+$ is the correct transcription, $\mathbf{s}^-$ is the negative sample and sim represents a similarity function (commonly, cosine similarity is used).

### 3.6 Convolution-Deconvolution Model

Variational Autoencoders (VAE) by (Kingma and Welling, 2013) have been used as generative models for learning a representation for images. We build a similar model which uses a convolution network to encode the sentence embeddings to learn a latent representation, and forwarding this to a deconvolution network to reconstruct the speech signal. Thus this can be used to learn a cross representation between sentence embeddings and acoustic signal as shown in Fig. 3 This

can be trained in an unsupervised fashion by feeding the model with embeddings of the speech transcription and reconstructing the speech signal. [2]

## 4 Experimental setup

We perform our experiments on the TIMIT corpus. There are broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. Each utterance is separated by three major categories: SA (dialect sentence), SX (compact sentence), and SI (diverse sentence). The SA sentences were meant to show the dialectal variants of the speakers and thus are ignored for our task.

### 4.1 Processing training samples

Each data sample obtained from the TIMIT corpus has a corresponding transcription file which provides word-level and sentence-level segmentation of the audio wave. For obtaining phoneme level segmentation, we can use external tools CMU Sphinx (Lamere et al., 2003).

### 4.2 Acoustic Features

Acoustic features mentioned in Section 3 are generated using a convolutional neural network over the raw speech data as done by (Palaz et al., 2015). The CNN is used to extract acoustic features as opposed to using MFCCs and other feature engineering techniques. This allows us to perform end-to-end training and thus keeps the model modular. The speech data is fed into the a two layer convolution neural network which then produces acoustic features.

### 4.3 Word embeddings

The word embeddings used in Section 3 can be generated using (Mikolov et al., 2013) [3]. If existing semantic embeddings doesn't prove to be effective, these embeddings can be modelled by adding another deep neural network model to generate word representations given the word as a one-hot vector. This might avoid having semantic information in the embedding and having a semantic embedding vector space. Rather, the model will try to learn a mapping from the word representation (in a one-hot vector encoding) to the acoustic embedding of the raw speech data.

[2]Link to code can be found in section 6

[3]Code available at `code.google.com/p/word2vec`

### 4.4 Descripition of deep architecture

The deep autoencoders consists of multiple hidden layers with sigmoid or ReLU non-linearities. Regularization and dropout is performed to avoid overfitting, and batch normalization is performed to reduce the internal covariate shift and decrease training time. Adam optimizer (Kingma and Ba, 2014) is used. All hyper-parameters required for training is set using a cross-validations set.

## 5 Evaluation

### 5.1 Methodolgy

To evaluate the quality of our embeddings, we can generate embeddings for a few homophones for all the different approaches and inspect their nearest neighbors. The nearest neighbors must show that words that are acoustically similar (sound similar) to the word in question. Also, operations such as $\mathbf{v}(testing)$ - $\mathbf{v}(ing)$ must be close to $\mathbf{v}(taste)$. t-SNE plots of the embedding can be visualized to see clusters of acoustically similar words.

Another approach to evaluate the quality of the embeddings is to use them for speech recognition using the embeddings to predict corresponding transcriptions of the speech sample. If the embeddings generated were good, the performance for the task of speech recognition must be on par or above the existing models that use acoustic embeddings such as (Chung et al., 2016) and (Kamper et al., 2016).

## 6 Results

We compiled a dataset of sentences and their transcripts, segmented corresponding to their words. Using this, we built our version of the convolution-deconvolution model described in section 3.6 to generate acoustic embeddings. Unfortunately, we didn't have the computational power to run the model and evaluate the results. We leave this to future work when we obtain enough computational power to perform this. However, the code for this can be found at `http://bit.ly/acoustic-embeddings`.

## 7 Conclusion

We propose multiple architectures and approaches to generate acoustic embeddings from speech data by augmenting it with word and sentence embeddings. These approaches derive from architectures

that have been successfully applied to other domains such as images. Despite not providing empirical results, we postulate that the above methods will generate good quality acoustic embeddings that can be used in speech recognition to improve the performance. The model currently doesn't support out-of-vocabulary words and uses shallow CNN models for feature extraction. This can be improved upon in future work.

## Acknowledgments

We wish to thank Prof. Ndapandula Nakashole for her extended support and guidance during the period of the course and this assignement. We also acknowledge the support from the department of Computer Science and Engineering at the University of California, San Diego. We would also like to thank the peer-reviewers (Cuong Luong and Vraj Shah) for their insightful feedback and comments that have been incorporated in this work.

## References

Samy Bengio and Georg Heigold. 2014. Word embeddings for speech recognition. In *Proceedings of the 15th Conference of the International Speech Communication Association, (Interspeech)*.

Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung-yi Lee, and Lin-Shan Lee. 2016. Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. pages 765–769. https://doi.org/10.21437/Interspeech.2016-82.

George E. Dahl, Dong Yu, Li Deng, and Alex Acero. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing* 20:30–42.

Mark J. F. Gales and Steve J. Young. 2007. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing* 1(3):195–304. https://doi.org/10.1561/2000000004.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*. pages 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947.

Herman Kamper, Weiran Wang, and Karen Livescu. 2016. Deep convolutional acoustic word embeddings using word-pair side information. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. pages 4950–4954. https://doi.org/10.1109/ICASSP.2016.7472619.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. 2014.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *CoRR* abs/1506.06726. http://arxiv.org/abs/1506.06726.

Paul Lamere, Philip Kwok, Evandro Gouvea, Bhiksha Raj, Rita Singh, William Walker, Manfred Warmuth, and Peter Wolf. 2003. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*. volume 1, pages 2–5.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. 2015. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *ACL*.

Xu Li, Zhiyong Wu, Helen M. Meng, Jia Jia, Xiaoyan Lou, and Lianhong Cai. 2016. Phoneme embedding and its application to speech driven talking avatar synthesis. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. pages 1472–1476. https://doi.org/10.21437/Interspeech.2016-363.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Dimitri Palaz, Mathew Magimai-Doss, and Ronan Collobert. 2015. Analysis of cnn-based speech recognition system using raw speech as input. In *INTERSPEECH*.

Shane Settle and Karen Livescu. 2016. Discriminative acoustic word embeddings: Recurrent neural network-based approaches. *CoRR* abs/1611.02550. http://arxiv.org/abs/1611.02550.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR* abs/1409.3215. http://arxiv.org/abs/1409.3215.

Gabriel Synnaeve, Thomas Schatz, and Emmanuel Dupoux. 2014. Phonetics embedding learning with side information. In *SLT*.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David R. Mortensen, Alan W. Black, Lori S. Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *HLT-NAACL*.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '08, pages 1096–1103. https://doi.org/10.1145/1390156.1390294.