Unsupervised and Weakly Supervised Acoustic Word Embeddings via CVAE

Puyuan Peng, Yi Nian

University of Chicago

pengpuyuan@uchicago.edu, nian@uchicago.edu

Abstract

Deep learning has helped greatly in improving the quality of supervised and unsupervised acoustic word embeddings (AWS). For unsupervised AWS, two popular deep learning based approaches are: Siamese networks[1, 2] and Autoencoder-based networks[3, 4]. In this project, we propose Correspondence Variational Autoencoder (CVAE) and its modified version CVAE2. We compare our models with other autoencoder-based models on a word discrimination task. Result shows that CVAE and CVAE2 substantially outperform existing models.

Index Terms:Acoustic word embeddings, zero-resource speech processing, unsupervised learning.

1. Introduction

Learning representations have gained significant traction across fields in machine learning including natural language processing, computer vision and speech. Acoustic word embedding (AWS) has shown it's advantages in many applications like Query-by-Example Search [5] and low-resource speech processing[4]. This report focus on unsupervised and weakly supervised acoustic word embeddings. The problem is defined as follows: Given unlabeled acoustic word (subword, phrase) segements $(x_1, x_2, ..., x_N)$ (unsupervised) or word pairs $\{(x_1^{(1)}, x_1^{(2)}), (x_2^{(1)}, x_2^{(2)}), \cdots, (x_N^{(1)}, x_N^{(2)})\}$ (weakly supervised), we want to learn a representation function f s.t. (a) $f(x)$ map x into lower-dimensional embedding; (b) $f(x_i)$ and $f(x_j)$ are close according to some distance measure λ iff x_i and x_j correspond to the same or similar acoustic units.

Note that since we can generate word pairs from the original unlabeled data using unsupervised term discovery system[6] or some data augmentation techniques that can be borrowed from computer vision [7]. So the difference between unsupervised acoustic embedding and weakly supervised embeddings is not the whether we have paired data but the quality of the pairs that we can generate from the original data. There are two popular approaches for unsupervised acoustic word embedding -

1. Siamese networks[1, 2, 8]
   It's a pair of networks with tied parameters which is trained to optimize a distance function between representations of two data instances
2. Autoencoder-based networks[4, 3, 9]
   it minimizes the reconstruction loss or regularized reconstruction loss and uses (part of) the output of the encoder as the embeddings of data.

In this report, we propose an autoencoder-based model, namely Correspondence Variational Autoencoder (CVAE) and it's modified version CVAE2, we conducted empirical analysis to compare our proposed model with Autoencoder (AE)[3], Variational Autoencoder (VAE)[10, 4] and Correspondence Autoencoder (CAE)[4] on their ability to generate high quality embeddings. Although our ultimate goal is to use the learned representation function for downstream tasks, for demonstration purposes, we measure the quality of embeddings by average precision each model gives on a word discrimination task.

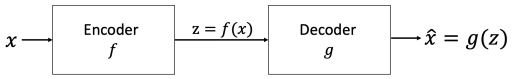

Figure 1: *Forward propagation in AE*

2. Autoencoder-based models

In this section, we will introduce AE, CAE, VAE, and CVAE in a general fasion, i.e. the general architecture, forward propagation and loss function. In turns of the the building blocks of the neural nets, we can choose them based on the types of data we have. For example, use RNN for MFCCs sequence and use CNN for spectragram or waveform.

2.1. Autoencoder and Correspondence AE

Autoencoder (AE) is a non-probabilistic data compression model. It's encoder takes input x and pass it through several layers and output a lower dimensional representation of the input x, which is usually denoted as z. It then uses z as the input of decoder and output the reconstructed data denoted as $\hat{x}$, which has the same dimension as x. The general architecture and forward propagation of AE is shown in Figure.1 AE updates it weights by minimizing the reconstruction error shown below:

$$\mathcal{L}_{\text{AE}} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{x_i} - x_i\|_2^2 \tag{1}$$

Correspondence Autoencoder (CAE) uses the same architecture and propagation scheme as AE, but it takes paired data $\{(x_1^{(1)}, x_1^{(2)}), (x_2^{(1)}, x_2^{(2)}), \cdots, (x_N^{(1)}, x_N^{(2)})\}$ and updates it's weight by minimizing the correspondence loss

$$\mathcal{L}_{\text{CAE}} = \frac{1}{2N}\sum_{i=1}^{N}\sum_{(j,k)\in I} \left\|\hat{x}_i^{(j)} - x_i^{(k)}\right\|_2^2 \tag{2}$$

where $I = \{(1,2),(2,1)\}$

Both AE and CAE uses the output of encoder $z = f(x)$ as the embedding for input x.

2.2. Variational Autoencoder

Variational Autoencoder (VAE) is the probabilistic counterpart of Autoencoder, The encoder takes input x and gives approximate posterior of latent variable z: $q_\phi(z|x)$, and we take M

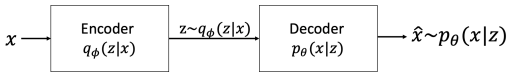

Figure 2: *Forward propagation in VAE*

samples from this posterior as input of the decoder, which then output the likelihood model for x: $p_\theta(x|z)$. VAE finds the best likelihood model $p_\theta(x|z)$ and approximate posterior $q_\phi(z|x)$ jointly by maximizing the evidence lower bound (ELBO) shown in equation3. For a full treatment of VAE, please refer to the original paper[10].

$$\text{ELBO} = E_{Z\sim q_\phi(Z|X)} \log p_\theta(X|Z) - D_{KL}(q_\phi(Z|X)||p(Z)) \tag{3}$$

Here we show the forward propagation and model architecture in Figure.2

In practice, we maximize the Monte Carlo estimator of ELBO

$$\mathcal{L}_{\text{VAE}} = \frac{1}{NM}\{\sum_{i=1}^{N}[\sum_{m=1}^{M} \log p_\theta(x_i|z_m)] - D_{KL}(q_\phi(Z|x_i)||p(Z))\} \tag{4}$$

Note that certain treatment (e.g. reparameterization) is needed[11] in order to make sure the gradient (error) can be back propagated through random variables z; since $D_{KL}(q_\phi(Z|x_i)||p(Z))$ usually has an analytical form, we don't need a Monte Carlo estimator for it.

As for the parametric form for $q_\phi(z|x)$ and $p_\theta(x|z)$, the most popular choice for real valued data is to set both of them to be Guassian distribution, i.e. $q_\phi(z|x) = \mathcal{N}(z|\mu(x), \Sigma(x)), p_\theta(x|z) = \mathcal{N}(x|\mu(z), \Sigma(z))$. This is also our setting in the later experiments. We will use $\mu(x)$, which is part of the output of encoder, as the embeddings of x.

2.3. Correspondence Variational Autoencoder (CVAE) and CVAE2

To motivate the idea of CVAE. We first take a step back to gain an intuition for AE and VAE. We can think of AE as PCA (linear AE is indeed equivalent to PCA), which we find a lower dimensional space that explains as much variation in the original data as possible. We can regard VAE as Bayesian Probabilistic PCA[12], which instead explains the variation of the data by introducing a lower dimensional latent variable z and learning conditional distribution for data (condition on z). It then uses the data to get the posterior for z. VAE is significantly more flexible than AE, because there are two layers of randomness in VAE (randomness in z and randomness in x) that AE doesn't have.

This flexibility gives VAE (and it's variants) the power to generate vivid pictures that don't exist, for example[10, 13]. But for our purposes, we want to restrict this kind of flexibility towards the results that the approximate posterior $q_\phi(z|x)$ generated by similar x's should be as similar as possible. **Our approach is using a correspondence-style loss for VAE - given paired data (x_1, x_2), we pass x_1 to the encoder which output the distribution for z; then sample z from this distribution and feed it to decoder and get likelihood $p_\theta(x|z)$ for x_1. Instead of measuring the $p_\theta(x_1|z)$ in the objective function, we measure $p_\theta(x_2|z)$.** We called this model Correspondence Variational Autoencoder (CVAE). CVAE has objective function:

$$\mathcal{L}_{\text{CVAE}} = \frac{1}{2NM}\sum_{i=1}^{N}\sum_{(j,k)=I}\{[\sum_{m=1}^{M} \log p_\theta(x_i^{(k)}|z_m)] - D_{KL}(q_\phi(Z|x_i^{(j)})||p(Z))\} \tag{5}$$

where $I = \{(1,2),(2,1)\}$

However, even x_1 and x_2 are from the same distribution (weakly supervised setting), due to the noisiness in sample x_1 and x_2, we should never expect the learned posterior $q_\phi(z|x_1)$ and $q_\phi(z|x_2)$ are the same and therefore the likelihood $p_\theta(x_1|z)$ and $p_\theta(x_2|z)$. Unfortunately, $\mathcal{L}_{\text{CVAE}}$ is actually expecting the likelihood of x_2 to be fairly close to x_1. This unrealistic expectation may pose some trouble on the optimization. To ease the optimization, we further relax the loss function which is shown in equation 6 and we call it CVAE2.

$$\mathcal{L}_{\text{CVAE2}} = \frac{1}{2N}\sum_{i=1}^{N}\sum_{(j,k)=I}\underset{m}{\text{argmax}}[\log p(x_i^{(k)}|z_m)] - D_{KL}(q_\phi(z|x_i^{(j)})||p(z)) \tag{6}$$

where $I = \{(1,2),(2,1)\}$

Just as VAE does, we will use Gaussian distribution for both approximate posterior and likelihood, and choose the mean of approximate posterior as the embedding.

3. Experiments

In this section, we introduce the dataset we used and the building blocks we chose for the models introduced in section 2. We present the preformance of these models on a word discrimination task. Result shows that our model CVAE and CVAE2 substantially outperform other models.

3.1. Data and model architecture

The data is drawn from the Switchboard English conversational speech corpus. The spoken word segments range in duration from 50 to 200 frames (0.5 - 2 seconds). The train/dev/test contain 9971/10966/11024 of acoustic segments and character sequences labels, corresponding to 1687/3918/3390 unique words.

The input to the models is a variable length sequence of 108-dimensional vectors (one per frame) of standard mel frequency cepstral coefficients (MFCCs) and their first and second derivatives.

Since the data are variable length sequence, we choose the building block of the models to be Gated Recurrent Unit (GRU)[14]. The architecture for AE and CAE is shown in figure3 and architecture for VAE, CVAE and CVAE2 is shown in figure 4 Encoder and decoder of the five models have the same architecture: three hidden unidirectional GRU layers with hidden units 300, 300, 300; dimension of latent variable z is 130; for number of samples for latent variable z (denote as M): For VAE, $M = 1$; For CVAE, we tried both $M = 1$ and $M = 5$; for CVAE2 we set $M = 5$. As stated in section 2.3, we use Gaussian distribution for both $q_\theta(z|x)$ and $p_\theta(x|z)$, and specifically, we fixed the covariance matrix of $p_\theta(x|z)$ to

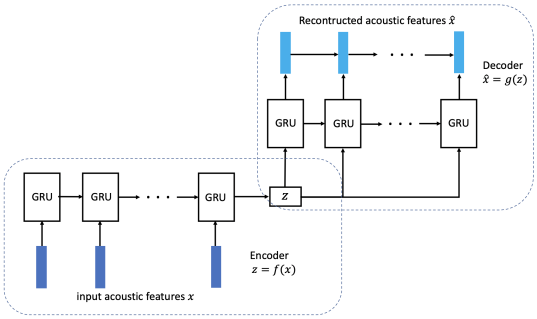

Figure 3: *Architecture for AE and CAE*

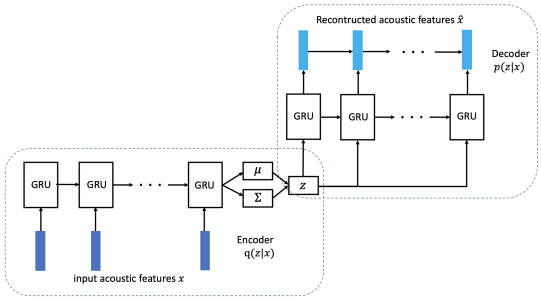

Figure 4: *Architecture for VAE and CVAE*

be $\sigma^2 I$ and set the prior $p(z) = \mathcal{N}(0, \sigma^2 I)$. σ^2 are the same for prior and likelihood and equals to 10^{-5}.

We used the original training set for training AE and VAE. For CAE and CVAE, since we actually have labels for each data points, we generated 20K pairs from training set. Since the pairs are generated based on their labels, we consider the quality of these pairs to be very high compare to pairs found by UTD or generated using acoustic data augmentation methods. Therefore, we consider our experiment on CAE, CVAE and CVAE2 to be weakly supervised.

3.2. Result

We trained the AE and VAE for 50 epoches and they all stop decreasing their losses after about 20 epochs, we used the trained AE and VAE that perform the best on validation set as the pre-trained model for CAE, CVAE and CVAE2 for 30 epoch each. CAE and CVAE(2) all stoped decreasing loss within 20 epochs. All models are trained using Adam optimization [15] with a learning rate of 0.001. Although we pre-trained CAE with AE, CVAE and CVAE2 with VAE, we did not verify the necessarity of pre-training in our case.

Then we choose the fives models that has the best performance on validation set as the final models to be tested.

We use word discrimination task which is also called same-different task [4]. In the same-different task, given a pair of acoustic segments(a real word),we must discriminate that whether these segments are examples of the same words. To do this using the above embedding models, a set of words in the test data are embedded using the specific approach. For every word pair in this set (we randomly separated the testset which contains 11024 acoustic word segments into batches that each

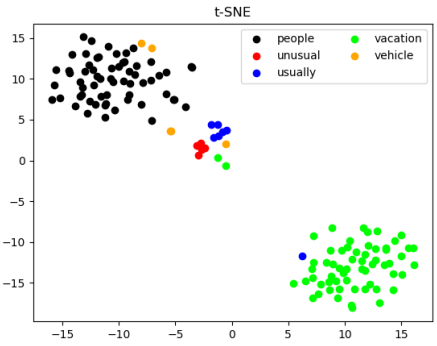

Figure 5: *t-SNE for CVAE2 with M=5*

batch has 300 words), the cosine distance between their embeddings is calculated. Two words can then be classified as being of the same or different type based on some threshold, and a precision-recall curve is obtained by varying the threshold. The area under this curve is used as final evaluation metric, referred to as the average precision (AP).

Table 1 shows the result. AE and VAE gives relatively low

Table 1: *Performance on test set*

model	**AP (%)**
AE	3.97
VAE	2.19
CAE	18.08
CVAE M = 1 (our approach)	24.50
CVAE M = 5 (our approach)	21.91
CVAE2 M = 5 (our approach)	**25.36**

average precision. This indicates the importance of correspondence training. In particular, VAE perform even worse than AE, we suspect that this is because VAE is too flexible and therefore very sensitive to the quality of data and initialization. [4] also has some discussion on training VAE for AWS. For the correspondence models CAE, CVAE and CVAE2, CVAE2 with sample size M equals to 5 achieves the highest average precision.

Note that it's under our expectation that non-correspondence models perform much worse than correspondence models since correspondence models use pairs of words which introduces additional information. If we generate pairs using UTD or data augmentation techniques, we expect that there will not be such a big gap between the performance of correspondence models and non-correspondence models.

Figure 5 shows the t-SNE[16] visualization of the CVAE2. We see that the model does a better job for words with a lot of instances ('people', 'vacation') than with fewer instance ('usually', 'vehicle'); And clearly, for words that are acoustically similar, they appears to be are closer ('unusual' and 'usually').

4. Future work

There are mainly three tasks that are left as future work. First, we need to test our model in a truly unsupervised setting. Which means we don't use labels to generate pairs but use unsupervised term discovery system or data augmentation techniques to

get acoustic word pairs for training. Second, try contrastive loss - correspondence loss takes care the similarity of two words that are from the same distribution, but doesn't explicitly consider the distance between word from different distributions. This can be tackled using a contrastive loss[17]. Third, compare the autoencoder-based model with Siamese networks.

5. References

[1] H. Kamper, W. Wang, and K. 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.

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

[3] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L. Lee, "Unsupervised learning of audio segment representations using sequence-to-sequence recurrent neural networks," in *Proc. Interspeech*, 2016, pp. 765–769.

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

[5] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," *arXiv preprint arXiv:1706.03818*, 2017.

[6] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2007.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[8] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[9] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[10] D. P. Kingma and M. Welling, "Auto encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[11] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[12] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[13] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," *arXiv preprint arXiv:1802.04364*, 2018.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[17] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, "A theoretical analysis of contrastive unsupervised representation learning," *arXiv preprint arXiv:1902.09229*, 2019.