A Correspondence Variational Autoencoder for Unsupervised Acoustic Word Embeddings

Puyuan Peng
Department of Statistics
University of Chicago, USA
pengpuyuan@uchicago.edu

Herman Kamper
Department of Electrical and Electronic Engineering
Stellenbosch University, South Africa
kamperh@sun.ac.za

Karen Livescu
Toyota Technological Institute at Chicago, USA
klivescu@ttic.edu

Abstract

We propose a new unsupervised model for mapping a variable-duration speech segment to a fixed-dimensional representation. The resulting *acoustic word embeddings* can form the basis of search, discovery, and indexing systems for low- and zero-resource languages. Our model, which we refer to as a maximal-sampling correspondence variational autoencoder (MCVAE), is a recurrent neural network (RNN) trained with a novel self-supervised correspondence loss that encourages consistency between embeddings of different instances of the same word. Our training scheme improves on previous correspondence training approaches through the use and comparison of multiple samples from the approximate posterior distribution. In the zero-resource setting, the MCVAE can be trained in an unsupervised way, without any ground-truth word pairs, by using the word-like segments discovered via an unsupervised term discovery system. In both this setting and a semi-supervised low-resource setting (with a limited set of ground-truth word pairs), the MCVAE outperforms previous state-of-the-art models, such as Siamese-, CAE- and VAE-based RNNs.

1 Introduction

Acoustic word embeddings (AWEs) are representations of arbitrary-length speech segments in a fixed-dimensional space, allowing for easy comparison between acoustic segments [Levin et al., 2013]. AWEs have been used to improve performance in multiple applications, such as query-by-example speech search [Settle et al., 2017, Ao and Lee, 2018, Jung et al., 2019, Yuan et al., 2018], automatic speech recognition [Bengio and Heigold, 2014, Settle et al., 2019, Shi et al., 2021], and zero-resource speech processing [Kamper et al., 2016a, Kamper et al., 2017].

While supervised AWE models have shown impressive performance, here we focus on unsupervised and semi-supervised settings where transcribed data is unavailable or very limited. This is the case for many zero- or low-resource languages, which make up most of the languages spoken in the world today [Besacier et al., 2014]. One of the first unsupervised AWE models was proposed by Chung et al. [2016], who trained an encoder-decoder RNN as an autoencoder (AE) to reconstruct unlabelled speech segments. Kamper [2019] extended this to the correspondence autoencoder RNN (CAE-RNN): Instead of trying to reconstruct an input segment directly, it tries to reconstruct another instance of the same class as the input. Since labelled data isn't available in the zero-resource setting, an unsupervised term discovery (UTD) system [Park and Glass, 2007] is used to automatically discover word-like training pairs.

Self-Supervised Learning for Speech and Audio Processing Workshop @ NeurIPS 2020.

In this work we extend these unsupervised models, proposing the maximal sampling correspondence variational autoencoder (MCVAE). In contrast to the above models, this is a generative probabilistic model that can be seen as an extension of a variational autoencoder (VAE) [Kingma and Welling, 2014]. It improves on the CAE-RNN through the use and comparison of multiple samples from the approximate posterior distribution, and is trained with a novel self-supervised correspondence loss which encourages different instances of the same (discovered) word type to have similar latent embeddings. We compare the MCVAE to previous approaches in both the unsupervised setting (where pairs from a UTD system are used for training) and the semi-supervised setting (where limited amounts of labelled data are used to provide weak supervision). In both settings we show that it outperforms previous approaches in a word discrimination task on two languages.

2 Problem formulation and existing approaches

Given unlabeled acoustic segments $(x^{(1)}, x^{(2)}, ..., x^{(N)})$ and/or segment pairs $\{(x_1^{(1)}, x_2^{(1)}), (x_1^{(2)}, x_2^{(2)}), \cdots, (x_1^{(N)}, x_2^{(N)})\}$, the goal of our AWE models[1] is to learn a function f such that (a) $f(x)$ maps the acoustic sequence x into a fixed-dimensional embedding, and (b) $f(x_1)$ and $f(x_2)$ are close according to some distance measure (e.g. cosine distance) if and only if x_1 and x_2 correspond to instances of the same linguistic type (words in our case). Below we introduce three existing AWE models.

The autoencoder recurrent neural network (AE-RNN) consists of an encoder and a decoder RNN [Chung et al., 2016]. The encoder takes a variable-length acoustic segment x and embeds it as a fixed-dimensional vector z, and the decoder then uses z to produce the reconstruction $\hat{x}$. The loss function for the AE-RNN is the empirical reconstruction error: $\frac{1}{N}\sum_{i=1}^{N}\left\|x^{(i)} - \hat{x}^{(i)}\right\|^2$. A probabilistic variant of the AE-RNN was proposed in Kamper [2019], where the model is trained as a variational autoencoder (VAE). We introduce and discuss this VAE-based model in more detail in Section 3.1, and extend it in the subsequent sections.

The correspondence autoencoder RNN (CAE-RNN), proposed in [Kamper, 2019] and further analyzed in [Matusevych et al., 2020, Kamper et al., 2020b]

, is trained with a correspondence loss, $\frac{1}{N}\sum_{i=1}^{N}\left\|x_2^{(i)} - \hat{x}_1^{(i)}\right\|^2$, where $x_1^{(i)}$ and $x_2^{(i)}$ are instances of the same word type (or belong to the same discovered cluster—see below) and $\hat{x}_1^{(i)}$ is the model output when the input is $x_1^{(i)}$. Here $x_1^{(i)}$ is the input to the encoder, and we want the model to reconstruct $x_2^{(i)}$ as its output. This loss explicitly helps the model to utilize pair information. One shortcoming is that it might be too hard a problem to reconstruct $x_2^{(i)}$ from $x_1^{(i)}$. In Section 3.2, we propose a probabilistic version of CAE-RNN to address this problem.

The SiameseRNN is a discriminative AWE model [Settle and Livescu, 2016], based on the early work of [Bromley et al., 1993]. It consists of an RNN encoder which takes acoustic segments and outputs the final state (or a transformation of the final state) as the embedding for the input segment. To encourage embeddings of segments of the same word type to be closer in the embedding space than embeddings of different-type segments, the SiameseRNN is trained by minimizing the triplet loss, $\frac{1}{N}\sum_{i=1}^{N}\max(0, m + d_{\cos}(x_a^{(i)}, x_d^{(i)}) - d_{\cos}(x_a^{(i)}, x_s^{(i)}))$, where $(x_a^{(i)}, x_s^{(i)})$ is a positive pair and $x_d^{(i)}$ is a random negative sample from the corpus. Since their introduction [Kamper et al., 2016b, Settle and Livescu, 2016], work on Siamese network-based embeddings has further improved them and explored their applications [Yang and Hirschberg, 2019, Yuan et al., 2018, Lim et al., 2018]. We will show in Sections 4.3 and 4.4 that the discriminative nature of the SiameseRNN makes it suitable for AWEs when we have high quality training pairs.

While the AE-RNN can be trained without any supervision, both the CAE-RNN and SiameseRNN require paired examples. These can be obtained from transcriptions in cases where (limited amounts of) labelled speech data are available. But in zero-resource settings, only unlabelled audio is available. In this case, we can use an unsupervised term discovery (UTD) system [Park and Glass, 2007, Jansen et al., 2010] to discover recurring word- or phrase-like patterns in the unlabelled data, thereby

[1] Other goals are possible, such as encoding semantic similarity [Chung and Glass, 2018].

automatically constructing noisy training pairs. Since the UTD system is unsupervised, the overall AWE approach is then unsupervised.

3 The maximal sampling correspondence variational autoencoder (MCVAE)

In this section we introduce our new models, starting with an extension of the VAE. We then introduce the new CVAE and MCVAE models, which can be seen as combinations of the VAE and the CAE-RNN for obtaining AWEs.

3.1 Variational autoencoder for acoustic word embeddings

A VAE was first used for AWEs in Kamper [2019], but it performed poorly. Apart from that work, the only other study to consider generative models for AWEs is that of Beguš [2020], who used generative adversarial networks (GANs). Here we extend the VAE approach of Kamper [2019].

To establish terminology, the graphical model of a VAE is shown in Figure 1, where X is the observed input (here, acoustic segment) and Z is the latent vector variable (here, the acoustic word embedding).[2] Solid lines denote the generative model $p(Z)p_\theta(X|Z)$ and dashed lines denote the variational approximation $q_\phi(Z|X)$ to the intractable posterior $p_\theta(Z|X)$. The variational approximation $q_\phi(Z|X)$ is assumed to be a diagonal Gaussian distribution whose mean and log variance is the output of an encoder network. The likelihood model $p_\theta(X|Z)$ is assumed to be a spherical Gaussian, and its mean is the output of a decoder network. ϕ and θ are weights of the encoder and decoder networks, respectively, and they are jointly learned by maximizing the evidence lower bound (ELBO):

$$\text{ELBO} = E_{Z\sim q_\phi(Z|x)} \log p_\theta(x|Z) - D_{KL}(q_\phi(Z|x)||p(Z)) \tag{1}$$

where x is a data point (acoustic segment), i.e. an instantiation of X. It can be shown that maximizing the ELBO is equivalent to minimizing $D_{KL}(q_\phi(Z|x)||p_\theta(Z|x))$ [Kingma and Welling, 2014].

The first term in equation 1 cannot be computed analytically, so we estimate it using samples of the latent variable; the objective over a training set $(x^{(1)}, x^{(2)}, ..., x^{(N)})$ then becomes:

$$J_{\text{VAE}} = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{1}{K}\sum_{k=1}^{K}\log p_\theta(x^{(i)}|z^{(k)}) - D_{KL}(q_\phi(Z|x^{(i)})||p(Z))\right\} \tag{2}$$

where $z^{(k)} \overset{\text{i.i.d}}{\sim} q_\phi(Z|x^{(i)})$, with $k = 1, 2, \cdots, K$.

For the building blocks of the encoder-decoder network, we use multi-layer RNNs with GRU [Cho et al., 2014] units. Note that other layers such as convolutional or transformer layers [Vaswani et al., 2017] could also be used.

One well-known issue with VAEs is posterior collapse [Razavi et al., 2019], where the decoder learns to ignore the latent variable Z and the approximate posterior collapses to the prior. Preventing the posterior from collapsing has been a very active research question [He et al., 2019, Razavi et al., 2019, Lucas et al., 2019]. Here we adopt the conditioning scheme used in Kamper [2019] and the KL term annealing trick used in Bowman et al. [2016] to tackle the posterior collapse problem.

Conditioning scheme. We use samples of Z as the input of the decoder RNN at every time step, as shown in Figure 3. Note that the network can be a bidirectional RNN with multiple GRU layers, in which case μ and Σ will be a concatenation of the final states of the last forward and backward layer. We also tried the conditioning scheme in Bowman et al. [2016], which uses the latent variable as initial state for the decoder and the target output from the previous step in a scheduled sampling scheme, to encourage the decoder to rely more on the initial state. However, in our experiments, this approach fails to learn a competitive representation, as measured by our discrimination task.

KL term annealing. We give a weight to the KL term in our objective function (equation 2) and gradually increase it from 0 to 1 as training proceeds. Similarly to Bowman et al. [2016], we use a sigmoid function for the weight:

$$\text{weight} = \frac{1}{1 + e^{-k(t-s_0)}}, \tag{3}$$

[2] We use uppercase letters to denote random variables and lowercase letters to denote their realizations.

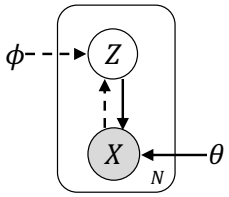

Figure 1: Graphical model for a VAE. X is the observation and Z is the latent variable. Solid lines denote the generative model and dashed lines denote the variational approximation.

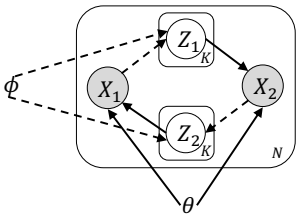

Figure 2: Graphical model for the correspondence training approach.

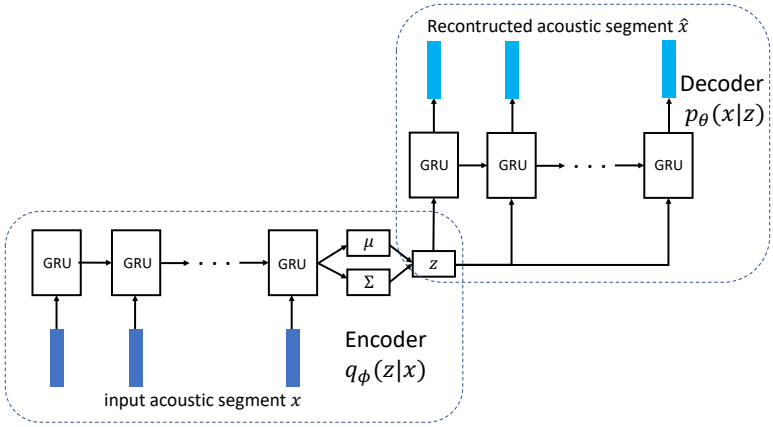

Figure 3: Network architecture of our model.

where t is the training iteration, and s_0 and k are hyperparameters that affect the starting point of the annealing and the increment in weight at each step during annealing.

3.2 A self-supervised correspondence objective

After training the VAE, it can infer the posterior of the latent variable Z and sample from the posterior to reconstruct X. If speech segments x_1 and x_2 correspond to the same word type (or belong to the same cluster in the unsupervised setting), the approximate posteriors $p_\phi(Z|x_1)$ and $p_\phi(Z|x_2)$ should be similar. We next ask the question: if we know that x_1 and x_2 correspond to the same word type, how can we explicitly express this information and use it to further improve the model?

Correspondence VAE. Inspired by Kamper [2019], we propose a probabilistic correspondence training approach, whose graphical model is shown in Figure 2. As in Figure 1, solid lines correspond to the generative model and dashed lines correspond to the inference model. To solve for parameters θ and ϕ in the graphical model via optimization, a natural choice for the objective function is

$$D_{KL}(q_\phi(Z|x_1)||p_\theta(Z|x_2)) + D_{KL}(q_\phi(Z|x_2)||p_\theta(Z|x_1)). \tag{4}$$

This objective expresses that we want the posterior conditioned on x_1 to be close to the posterior conditioned on x_2. Minimizing this loss is equivalent to maximizing

$$\begin{aligned}\text{ELBO}_1 + \text{ELBO}_2 = {} & E_{Z\sim q_\phi(Z|x_1)} \log p_\theta(x_2|Z) - D_{KL}(q_\phi(Z|x_1)||p(Z)) \\ & + E_{Z\sim q_\phi(Z|x_2)} \log p_\theta(x_1|Z) - D_{KL}(q_\phi(Z|x_2)||p(Z))\end{aligned}$$

Given data $\{(x_1^{(1)}, x_2^{(1)}), (x_1^{(2)}, x_2^{(2)}), \cdots, (x_1^{(N)}, x_2^{(N)})\}$, the objective can be approximated

$$J_{\text{CVAE}} = \frac{1}{N}\sum_{i=1}^{N}\left\{\left[\frac{1}{K}\sum_{k_2=1}^{K}\log p_\theta(x_1^{(i)}|z_2^{(k_2)})\right] - D_{KL}(q_\phi(Z|x_1^{(i)})||p(Z)) \right.$$
$$\left. + \left[\frac{1}{K}\sum_{k_1=1}^{K}\log p_\theta(x_2^{(i)}|z_1^{(k_1)})\right] - D_{KL}(q_\phi(Z|x_2^{(i)})||p(Z))\right\},$$
$$z_1^{(1)}, \cdots, z_1^{(K)} \overset{\text{i.i.d}}{\sim} q_\phi(Z|x_1^{(i)}), \quad z_2^{(1)}, \cdots, z_2^{(K)} \overset{\text{i.i.d}}{\sim} q_\phi(Z|x_2^{(i)}). \tag{5}$$

We refer to the model trained with this new objective function as a correspondence VAE, or CVAE.[3]

In practice, a pair of samples of the same word can be very different, making the task of the CVAE too challenging. We next propose a technique intended to better handle the gap between different acoustic segments of the same word type.

Maximal sampling correspondence VAE. Given the data pair (x_1, x_2), we pass x_1 to the encoder network and get the approximate posterior $q_\phi(Z|x_1)$. We then draw independent samples $z_1, z_2, ..., z_K$ from it and require x_2 to be likely only according to the "best" $p_\theta(X|z_k)$ and vice versa. In other words, given speech segment pairs $\{(x_1^{(1)}, x_2^{(1)}), (x_1^{(2)}, x_2^{(2)}), \cdots, (x_1^{(N)}, x_2^{(N)})\}$, the objective function is

$$J_{\text{MCVAE}} = \frac{1}{N}\sum_{i=1}^{N}\left\{\left[\max_{k_2}\log p_\theta(x_1^{(i)}|z_2^{(k_2)})\right] - D_{KL}(q_\phi(Z|x_1^{(i)})||p(Z)) \right.$$
$$\left. + \left[\max_{k_1}\log p_\theta(x_2^{(i)}|z_1^{(k_1)})\right] - D_{KL}(q_\phi(Z|x_2^{(i)})||p(Z))\right\},$$
$$z_1^{(1)}, \cdots, z_1^{(K)} \overset{\text{i.i.d}}{\sim} q_\phi(Z|x_1^{(i)}), \quad z_2^{(1)}, \cdots, z_2^{(K)} \overset{\text{i.i.d}}{\sim} q_\phi(Z|x_2^{(i)}). \tag{6}$$

Since we are maximizing over samples, we refer to this technique as "maximal sampling" and to the model as the maximal sampling correspondence VAE, or MCVAE.

Note that the number of samples K has different roles in the equations above. In equations 2 and 5, a larger K can reduce the variance of the estimator and therefore stabilize training. In equation 6, K represents how much the embedding space should be explored. If we assume that x_1 and x_2 are close, then a small K is preferable, since we don't need (and don't want) too much variation when reconstructing x_2 from x_1. On the other hand, if we expect x_1 and x_2 to be far apart, we might need a larger K. In our experiments, K is tuned as a hyperparameter on a validation set. In practice we put a weight on the KL term in equation 6, a common practice in the VAE literature which can be viewed as a way to balance the quality of the reconstruction and simplicity of the representation [Higgins et al., 2017, Chorowski et al., 2019]. In addition, the weight on the KL term also controls the latent space we want our model to explore: we find that a large weight on the KL term will lead to a large posterior variance and thus more variable reconstructions. Batch size is also an important factor, since the model is optimized using stochastic gradient descent; if the batch size is too small, it's possible that within one batch there are only two instances (i.e. one pair) for many word types, and this will encourage deterministic mapping between the two instances and therefore make the optimization difficult.

4 Experiments

4.1 Experimental setup

While AWEs can be used for a variety of downstream tasks, for comparison purposes we use a word discrimination (same-different) task [Carlin et al., 2011], often used in prior work to evaluate the quality of embeddings. In the same-different task, given a pair of speech segments corresponding to one word each, we must determine whether these segments are examples of the same word. For each pair of segments in the test set, we compute the cosine distance between their embeddings. Two

[3]Note that this is not to be confused with the conditional VAE.

segments can then be classified as being of the same or different type based on a threshold on the cosine distance, and a precision-recall curve is obtained by varying the threshold. The area under this curve, referred to as the average precision (AP), is used as the final evaluation metric. AP ranges from 0 to 1 and we report it as a percentage.

For all experiments, the encoder and decoder networks are both 2-layer bidirectional GRU networks, with 300 hidden units. The final states of the second forward and backward layers are concatenated and compressed to a 260-dimensional vector via a linear layer. This vector is split in half, with the mean and variance of $q_\phi(Z|x)$ each forming a half, and the final embedding is the 130-dimensional mean vector (which is the same dimensionality as in Kamper [2019]). For experiments with ground-truth training pairs, the sample size K of the latent variable Z is set to 5; for experiments with pairs discovered by UTD, K is set to 10. The variance of $p_\theta(x|z)$ is set to 0.01 for both pre-training and correspondence training. For the KL term annealing, we set k to 0.02 and s_0 to 1000. For CVAE and MCVAE, we set the weight on the KL term to be 0.001 (weight annealing is used for CVAE). We use the Adam optimizer [Kingma and Ba, 2015] with learning rate of 0.001. The batch size is 100 for most experiments except those in Section 4.3, where we also show results with batch size 400[4].

We experiment with data from English and Xitsonga. English training, validation and test sets are obtained from the Buckeye corpus [Pitt et al., 2005], each with around 6 hours of speech. For Xitsonga, we use a 2.5-hour portion of the NCHLT [Barnard et al., 2014] corpus. The English ground-truth pairs used in Section 4.4 are generated based on the word alignments provided with the Buckeye corpus. Following the same setup as in [Kamper, 2019], for the unsupervised models in Sections 4.3 and 4.4, we use terms discovered by the UTD system of [Jansen and Van Durme, 2011], respectively giving 10k and 11k segments from the English and Xitsonga corpora. Pairs are generated based on the clusters from the UTD system, resulting in 14k and 6k pairs, respectively. The training, validation and test splits are exactly the same as in [Kamper, 2019]. The input features are 13-dimensional mel-frequency cepstral coefficients (MFCCs) plus velocity and acceleration vectors.[5]

4.2 Study of model variants

In this section we report on development experiments to compare options for objective functions, as well as disentangle the effects of the pre-training and correspondence training.

Benefits of KL term annealing in pre-training. Figure 4 is a direct comparison between validation APs (in percent) given by a vanilla VAE and a VAE with the KL term annealed, trained on ground-truth segments. Figure 5 shows how the KL term annealing trick helps control the KL divergence while improving the embedding quality.

CVAE vs. MCVAE. We train CVAE and MCVAE models starting from the pre-trained annealed VAE. For UTD experiments, all 14k pairs are used; for ground-truth (GT) experiments, 10k pairs are generated based on labels. Each experiment is repeated five times with different seeds, and averaged results and standard deviations are reported in Table 1. We see that for both ground-truth pairs and UTD pairs, the MCVAE achieves a better validation AP .

Effect of pre-training and correspondence training. Table 2 shows the separate and combined effects of pre-training and correspondence training. For both UTD and ground-truth data, combining pre-training and correspondence training gives significantly better results than doing only one of the steps. However, the individual increments due to correspondence training and pre-training are different. For UTD, the improvement due to pre-training is slightly larger than that due to correspondence training, whereas for ground-truth data, the improvement due to training is much larger than that due to pre-training. This indicates the importance of pair quality in correspondence training. In addition, the pre-trained model on UTD data is slightly better than the one pre-trained on ground-truth data, indicating that pre-training is robust to the quality of the acoustic segments.

4.3 Test set results for unsupervised acoustic word embeddings

In this section, we present results for models trained on segments and pairs discovered by UTD and evaluated on test data. For English, we do early stopping using the 2.7k ground-truth pairs that

[4]This is mainly for computation considerations. We found that larger batch size lead to a better performance.

[5]While Kamper [2019] used only 13-dimensional MFCCs, we found on English development data that all models perform similarly or better when including velocity and acceleration coefficients and we therefore use 39 dimensional MFCCs throughout.

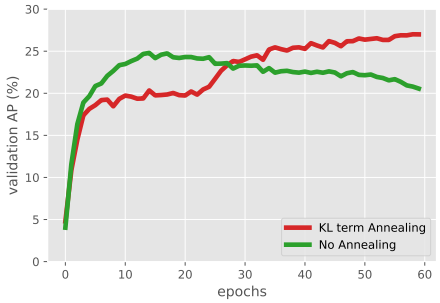

Figure 4: Comparison of APs between vanilla VAE and VAE with KL term annealed. Both models are trained on ground-truth segments.

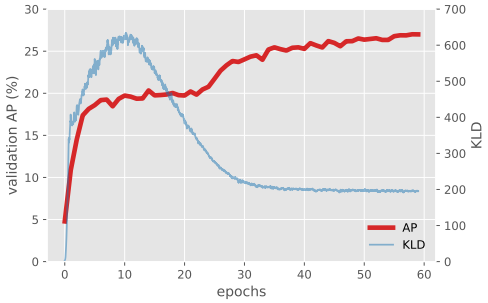

Figure 5: KL divergence term in the VAE objective plotted alongside the validation AP of the VAE with KL term annealing.

Table 1: Comparison of correspondence VAE-based models pre-trained as annealed VAEs, evaluated on the word discrimination task on validation data.

Data	Model	AP (%)
UTD	CVAE	35.4 ± 0.47
UTD	MCVAE	37.6 ± 0.78
GT	CVAE	55.1 ± 0.69
GT	MCVAE	58.8 ± 0.81

Table 2: Ablation study on the effect of correspondence training ('corr-train') and pre-training for the MCVAE, evaluated on the word discrimination task on validation data.

Data	Pre-train	Corr-train	AP (%)
UTD	✓		29.7 ± 0.49
UTD		✓	25.6 ± 0.98
UTD	✓	✓	37.6 ± 0.78
GT	✓		27.8 ± 0.42
GT		✓	43.6 ± 1.13
GT	✓	✓	58.8 ± 0.81

are available for validation. For Xitsonga, no validation set is available. Instead we use English UTD data, with size similar to that of the Xitsonga UTD data, to choose the number of epochs for pre-training and correspondence training (40 and 5 respectively).

Results are reported in Table 3 for our models together with several previous models and baselines. As a naive baseline, downsampling uses 10 equally spaced MFCC vectors from a segment (with appropriate interpolation), and then flattens them to obtain an embedding. In DTW alignment, the alignment cost of the full sequences is used to make the same-different decision. We also report results for the AE [Kamper, 2019, Chung et al., 2016], VAE [Kamper, 2019], annealed VAE, SiameseRNN [Settle and Livescu, 2016] and CAE-RNN [Kamper, 2019] acoustic word embedding models.[6]

Among all models, the MCVAE performs the best. It not only outperforms its closest AWE competitor (the CAE-RNN) by 11.3% and 37.9% relative on English and Xitsonga, respectively, but also outperforms DTW, which uses the full uncompressed sequences. To our knowledge, this is the first time an unsupervised acoustic word embedding approach outperforms DTW on the Buckeye corpus. The SiameseRNN is not competitive in this setting, potentially because it is more reliant on high-quality training pairs than the other models (which is supported by prior work [Kamper et al., 2020a]).

4.4 Semi-supervised acoustic word embeddings

In this section we study three AWE models—the CAE-RNN, SiameseRNN and MCVAE—in a semi-supervised setting, in order to mimic the low-resource language processing scenario. In this setting, we start with models from the previous section that have been trained in an unsupervised way, and continue training them but now using ground-truth pairs. To show how different amounts of ground-truth data improve the results, we increase the number of training pairs from only 1k to 50k.

[6]We fully tune the network architecture for SiameseRNN, leading to a 2-layer bidirectional GRU network with 400 hidden units. We also fully tune the architecture of CAE-RNN and find that the architecture described in Kamper [2019] still gives the best validation results. Finally, we also tune the batch size for the two models; unlike for MCVAE, this does not have a large effect on these models, and a batch size of 300 is best for both.

Table 3: Word discrimination performance on test data for models trained on UTD segments.

	Average Precision (%)	
Model	English	Xitsonga
SiameseRNN	17.5 ± 0.39	25.1 ± 1.02
AE	26.4 ± 0.51	18.0 ± 0.35
VAE	27.7 ± 0.42	15.7 ± 0.73
Annealed VAE (ours)	29.2 ± 0.67	17.0 ± 0.31
CAE-RNN	35.5 ± 0.22	32.2 ± 0.88
MCVAE (ours)	37.6 ± 0.65	40.2 ± 0.52
MCVAE (ours, large batch size)	$\mathbf{39.5 \pm 0.23}$	$\mathbf{44.4 \pm 0.59}$
Downsampling [Kamper, 2019]	21.7	13.6
DTW alignment [Kamper, 2019]	35.9	28.1

To examine the models' performance under different training data distributions, we generate pairs in two ways: (1) **Random pairs:** randomly sample from the dataset, and thus the distribution of pairs will follow the word frequency distribution in the corpus. This will lead to an imbalanced training set, but the data distribution will be similar to that of the validation set. (2) **Balanced pairs:** generate pairs with a more balanced distribution. To achieve this, we set an upper bound for the number of segments of the same word type based on the total number of pairs we want to generate. For example, to get 25k pairs in total, we generate at most 5 pairs of each word. The result is shown in Figure 6.

There are several interesting things to notice in Figure 6. First, the MCVAE is the most data-efficient model within the range of data sizes that we consider here, as most of the time the MCVAE outperforms other models by a large margin. Second, the encoder-decoder based models (the MCVAE and CAE-RNN) are more robust to changes in the training data distribution than the discriminative SiameseRNN. Among the two encoder-decoder models, the MCVAE is more robust than the CAE-RNN, as gap between between the random and balanced training set results is narrowed as the amount of training data increases. Third, the SiameseRNN performs surprisingly well in the low to median data regime (3k - 25k) when training pairs are balanced; however, it reaches its best performance at 11k and plateaus afterwards.

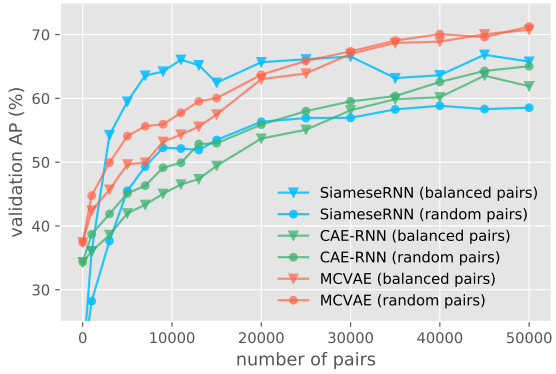

Figure 6: Validation AP vs. number of ground-truth pairs for supervised training, for models pre-trained on UTD data.

5 Conclusion

We have presented the MCVAE, a generative model for unsupervised and semi-supervised learning of acoustic word embeddings. We have shown that it is robust to the amount, distribution, and quality of training data. On the English Buckeye corpus, this is the first time that an unsupervised acoustic word embedding approach outperforms DTW. Future work includes explicitly incorporating label information for training when available (as in, e.g., He et al. [2017], Bengio and Heigold [2014]), generating higher quality segment pairs for unsupervised training, and applying the model in downstream tasks such as query-by-example speech search.

References

C.-W. Ao and H.-y. Lee. Query-by-example spoken term detection using attention-based multi-hop networks. In *Proc. ICASSP*, 2018.

E. Barnard, M. Davel, C. J. van Heerden, F. Wet, and J. Badenhorst. The NCHLT speech corpus of the South African languages. In *Proc. SLTU*, 2014.

G. Beguš. CiwGAN and fiwGAN: Encoding information in acoustic data to model lexical learning with generative adversarial networks. *arXiv preprint arXiv:2006.02951*, 2020.

S. Bengio and G. Heigold. Word embeddings for speech recognition. In *Proc. INTERSPEECH*, 2014.

L. Besacier, E. Barnard, A. Karpov, and T. Schultz. Automatic speech recognition for under-resourced languages: A survey. *Speech Commun.*, 56:85–100, 2014.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio. Generating sentences from a continuous space. In *CoNLL*, 2016.

J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a "Siamese" time delay neural network. In *Int. J. Pattern Recognit. Artif. Intell.*, 1993.

M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky. Rapid evaluation of speech representations for spoken term discovery. In *Proc. INTERSPEECH*, 2011.

K. Cho, B. van Merrienboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. EMNLP*, 2014.

J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord. Unsupervised speech representation learning using WaveNet autoencoders. *IEEE Trans. Acoust., Speech, Signal Process.*, 2019.

Y.-A. Chung and J. Glass. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. In *Proc. INTERSPEECH*, 2018.

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.

J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *Proc. ICLR*, 2019.

W. He, W. Wang, and K. Livescu. Multi-view recurrent neural acoustic word embeddings. In *Proc. ICLR*, 2017.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. ICLR*, 2017.

A. Jansen and B. Van Durme. Efficient spoken term discovery using randomized algorithms. In *Proc. ASRU*, 2011.

A. Jansen, K. Church, and H. Hermansky. Towards spoken term discovery at scale with zero resources. In *Proc. INTERSPEECH*, 2010.

M. Jung, H. jun Lim, J. Goo, Y. Jung, and H. Kim. Additional shared decoder on Siamese multi-view encoders for learning acoustic word embeddings. In *Proc. ASRU*, 2019.

H. Kamper. Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models. In *Proc. ICASSP*, 2019.

H. Kamper, A. Jansen, and S. Goldwater. Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE Trans. Acoust., Speech, Signal Process.*, 24(4):669–679, 2016a.

H. Kamper, W. Wang, and K. Livescu. Deep convolutional acoustic word embeddings using word-pair side information. In *Proc. ICASSP*, 2016b.

H. Kamper, K. Livescu, and S. Goldwater. An embedded segmental k-means model for unsupervised segmentation and clustering of speech. In *Proc. ASRU*, 2017.

H. Kamper, Y. Matusevych, and S. Goldwater. Improved acoustic word embeddings for zero-resource languages using multilingual transfer. *arXiv preprint arXiv:2006.02295*, 2020a.

H. Kamper, Y. Matusevych, and S. Goldwater. Multilingual acoustic word embedding models for processing zero-resource languages. In *Proc. ICASSP*, 2020b.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

D. P. Kingma and M. Welling. Auto encoding variational Bayes. In *Proc. ICLR*, 2014.

K. Levin, K. Henry, A. Jansen, and K. Livescu. Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings. In *Proc. ASRU*, 2013.

H. Lim, Y. Kim, Y. Jung, M. Jung, and H. Kim. Learning acoustic word embeddings with phonetically associated triplet network. *arXiv preprint arXiv:1811.02736*, 2018.

J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. In *Proc. NeurIPS*, 2019.

Y. Matusevych, H. Kamper, and S. Goldwater. Analyzing autoencoder-based acoustic word embeddings. In *ICLR Workshop on Bridging AI and Cognitive Science (BAICS)*, 2020.

A. S. Park and J. R. Glass. Unsupervised pattern discovery in speech. *IEEE Trans. Acoust., Speech, Signal Process.*, 2007.

M. Pitt, K. Johnson, E. Hume, S. F. Kiesling, and W. Raymond. The Buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability. *Speech Commun.*, 2005.

A. Razavi, A. v. d. Oord, B. Poole, and O. Vinyals. Preventing posterior collapse with delta-VAEs. In *Proc. ICLR*, 2019.

S. Settle and K. Livescu. Discriminative acoustic word embeddings: Recurrent neural network-based approaches. In *Proc. SLT*, 2016.

S. Settle, K. Levin, H. Kamper, and K. Livescu. Query-by-example search with discriminative neural acoustic word embeddings. In *Proc. INTERSPEECH*, 2017.

S. Settle, K. Audhkhasi, K. Livescu, and M. Picheny. Acoustically grounded word embeddings for improved acoustics-to-word speech recognition. In *Proc. ICASSP*, 2019.

B. Shi, S. Settle, and K. Livescu. Whole-word segmental speech recognition with acoustic word embeddings. In *Proc. SLT*, 2021.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017.

Z. Yang and J. Hirschberg. Linguistically-informed training of acoustic word embeddings for low-resource languages. In *Proc. INTERSPEECH*, 2019.

Y. Yuan, C.-C. Leung, L. Xie, H. Chen, B. Ma, and H. Li. Learning acoustic word embeddings with temporal context for query-by-example speech search. In *Proc. INTERSPEECH*, 2018.