Learning word embeddings: unsupervised methods for fixed-size representations of variable-length speech segments

Nils Holzenberger [1,*], *Mingxing Du* [2], *Julien Karadayi* [2], *Rachid Riad* [2,3], *Emmanuel Dupoux* [2]

[1]CLSP, Johns Hopkins University, USA
[2]ENS, EHESS, PSL Research University, CNRS, INRIA, France
[3]INSERM, NPI, UPEC, France

nholzen1@jhu.edu, mingxing.du@polytechnique.edu, {julien.karadayi, riadrachid3, emmanuel.dupoux}@gmail.com

Abstract

Fixed-length embeddings of words are very useful for a variety of tasks in speech and language processing. Here we systematically explore two methods of computing fixed-length embeddings for variable-length sequences. We evaluate their susceptibility to phonetic and speaker-specific variability on English, a high resource language, and Xitsonga, a low resource language, using two evaluation metrics: ABX word discrimination and ROC-AUC on same-different phoneme n-grams. We show that a simple downsampling method supplemented with length information can be competitive with the variable-length input feature representation on both evaluations. Recurrent autoencoders trained without supervision can yield even better results at the expense of increased computational complexity.

Index Terms: unsupervised speech processing, audio word embeddings, ABX discrimination, same-different classification, representation learning.

1. Introduction

Techniques to efficiently embed words into a vector space [1, 2] have led to spectacular successes in several Natural Language Processing (NLP) tasks such as language modeling [3], translation [4], topic modeling [5] and anaphora resolution [6]. In speech, similar techniques could be very useful, and are already used for several applications such as query by example [7], spoken term discovery [8] and unsupervised representation learning [9, 10]. However, before applications inspired by NLP can be deployed, a complication specific to speech has to be addressed. In text, the embedding for two instances of the same word will be the exact same vector. With audio input, due to phonetic and speaker variability, they won't be identical. Instead, we are dealing with a distribution over vectors, potentially overlapping with the distribution of another word. Therefore, a prerequisite for developing useful vector representations for spoken words is to minimize the impact of this variability.

Here, we propose to systematically explore the phonetic variability of audio word embeddings using two widely different approaches. The first one has been proposed by [11]. It relies on the idea of downsampling the time sequence of acoustic information using a fixed number of samples. This is a simple idea, which requires no training, and is worth investigating, at least as a baseline. The second idea is to use Recurrent Neural Networks (RNN) as autoencoders, by training them to encode a sequence of frames into a vector, and to decode it back to the original sequence of frames. Contrary to downsampling, this technique requires training, but no labels are needed, making it therefore applicable to a variety of speech signals.

We propose to evaluate the phonetic variability of word embeddings using two metrics, based on the computation of the within-word and between-word distance of embeddings. The first one, ABX, tests the average discriminability between pairs of words. The second one, AUC, tests for the separation, across the entire corpus, of the distribution of cosine or Euclidean distance for pairs of same phoneme n-grams versus the distribution of that distance for pairs of different phoneme n-grams.

2. Related work

The framework of segmental speech recognition [12] involves an explicit segmentation of speech utterances and a fixed-size representation for each acoustic segment. [13] use mean pooling and temporal derivatives over subparts of each segment, duration and energy, while [14] warp segments into a fixed length. Recent approaches represent segments with RNNs [15], or Deep Neural Networks (DNN) together with subsampling, pooling and duration information [16, 17]. Fixed-size embeddings are also used for speaker verification and extracted with factor analysis [18] or DNNs [19]. DNNs have been used to derive embeddings by training them with a supervised objective, although not for speech recognition per se: [20, 21] alternate convolutional and pooling layers, finishing with a fixed length vector.

In [22], word embeddings are obtained unsupervisedly by first memorizing a reference set of N words, and calculating the Dynamic Time Warping [23] (DTW) distance of a new target word with each of these references. This set of distances is then projected into a subspace by dimensionality reduction. This method obtains good quality embeddings with a large reference set but suffers from a high computational cost. In [11], this approach is replaced by a simpler procedure, where a fixed-size embedding is extracted from a sequence of frames by picking 10 equally-spaced frames from the sequence. In this paper, we will explore improvements of this technique.

More recently, sequence-to-sequence autoencoders [9, 10] have been used on speech to learn fixed-length embeddings in a purely unsupervised fashion. This has the advantage of providing embeddings in low-resource settings, for datasets where a transcript is not available. This is the kind of architecture and training setup we will use in this paper.

3. Proposed approach

Even though both downsampling and recurrent autoencoders have been used in past work, they have never been directly

[*] Work done while author was at ENS

compared, nor have the hyperparameters of downsampling been thoroughly explored. In addition, we compare these models with a DTW baseline, which aligns raw features to compute the distance between two sequences.

3.1. Downsampling

The downsampling techniques used in [11] extract a fixed number of equidistant samples from a time series. Here, we explore variants of this idea.

Formally, let $\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_T}$, with $\boldsymbol{x_t} \in \mathbb{R}^{40}$ be a sequence of 40-dimensional log mel features. Equidistant downsampling samples k vectors at interval $\frac{T}{k-1}$ with proportional interpolation as needed. Let $\boldsymbol{\hat{x}_{q_i}}$ be the i-th sample, where $1 \leq i \leq k$ and $1 = q_1 < q_2 < ... < q_k = T$, $q_{j+1} - q_j = \frac{T}{k-1}$ for $j = 1, 2, ..., k-1$. Note $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ the floor and ceiling functions, then we have :

$$\boldsymbol{\hat{x}_{q_i}} = \boldsymbol{x}_{\lfloor \boldsymbol{q_i} \rfloor} \cdot (\lceil q_i \rceil - q_i) + \boldsymbol{x}_{\lceil \boldsymbol{q_i} \rceil} \cdot (q_i - \lfloor q_i \rfloor)$$

When q_i is an integer, we take $\boldsymbol{x_{q_i}}$ as the sample; otherwise, the sample is a weighted sum of its left and right neighbors. The closer the neighbor, the more weight it contributes. The embedding of $\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_T}$ is the concatenation of $\boldsymbol{\hat{x}_{q_1}}, \boldsymbol{\hat{x}_{q_2}} ..., \boldsymbol{\hat{x}_{q_k}}$.

The acoustic information in a word is not necessarily distributed uniformly along the time axis; psycholiguistics suggest that it is mostly located at the boundaries. Thus, we introduce non-equidistant downsampling: we assume the space between two samples follows a linear progression and is symmetric with respect to the center, i.e $\Delta_j := q_{j+1} - q_j = k(j-1) + b$ and $q_{j+1} - q_j = q_{k-j+1} - q_{k-j}$ for $j = 1, 2, ..., \lceil k/2 \rceil$. To illustrate the degree of non-equidistance, instead of setting k as hyperparameter, we introduce the ratio $\alpha = \frac{\Delta_{\lceil k/2 \rceil}}{\Delta_1}$. Using $q_1 = 1$ and $q_k = T$, k and b can be deduced from α. α represents the ratio of the middle space against the first space. For example if $\alpha > 1$ then we take more samples at the two ends, if $\alpha < 1$ we take more samples in the middle, if $\alpha = 1$, then we come back to equidistant downsampling.

We further extend proportional interpolation to Gaussian weighted interpolation, the idea being to avoid loosing information. Formally, consider the sequence of log mel features as a step function i.e. $\boldsymbol{f(t)} = \boldsymbol{x_j}$ for $j - 0.5 < t <= j + 0.5$, $j = 1, 2, ..., T$ (this amounts to saying that log mel features are represented at the center of the time stride). Then for i_{th} sample $\boldsymbol{\hat{x}_{q_i}}$, we introduce $g_i = \mathcal{N}(q_i, \sigma_i^2)$, a Gaussian density function centered at q_i with variance σ_i^2. We have the following formula and illustrate it in figure 1:

$$\boldsymbol{\hat{x}_{q_i}} = \frac{\int_{0.5}^{T+0.5} g_i(t) \boldsymbol{f(t)} dt}{Z_i}$$

where Z_i is the normalization term, $Z_i = \int_{0.5}^{T+0.5} g_i(t) dt$. Combining the downsampling methods with Gaussian weight, we have tested 5 variants with equidistant downsampling:

- (EPI) Proportional interpolation (baseline): no hyperparameters.
- (EAG) Absolute Gaussian weight: one hyperparameter σ, as described previously.
- (ERG) Relative Gaussian weight: one hyperparameter β, setting σ proportional to the sequence length, i.e. $\sigma = \beta T$.
- (EALG) Absolute linear Gaussian weight: two hyperparameters k, σ_1. Instead of setting σ for all samples, we set σ_j for j_{th} sample to be a linear function and symmetric with respect to the center, i.e. $\sigma_j = k(j-1) + \sigma_1$ and $\sigma_j = \sigma_{k-j+1}$ for $j = 1, 2, ..., \lceil k/2 \rceil$.
- (ERLG) Relative linear Gaussian weight: two hyperparameters k, β. The only difference from the previous one is that we set the interception proportional to the sequence length, i.e. $\sigma_j = k(j-1) + \beta T$ and $\sigma_j = \sigma_{k-j+1}$ for $j = 1, 2, ..., \lceil k/2 \rceil$.

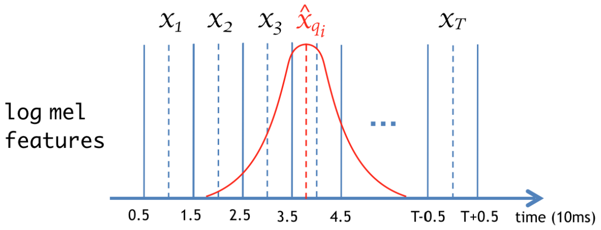

Figure 1: *Illustration of Gaussian weighted interpolation. Note that our log mel features are generated with a 10 ms time stride.*

and 5 non-equidistant downsampling methods:

- (NPI) Proportional interpolation: one hyperparameter α, as described previously.
- (NAG) Absolute Gaussian weight: two hyperparameters α, σ. α for non-equidistant downsampling, σ for Gaussian weight.
- (NRG) Relative Gaussian weight: two hyperparameters α, β. α for non-equidistant downsampling, setting σ proportional to the sequence length, i.e. $\sigma = \beta T$.
- (NALG) Absolute linear Gaussian weight: two hyperparameters α, σ_1. α for non-equidistant downsampling, setting linear Gaussian to the same slope calculated by non-equidistant downsampling, i.e. $\Delta_j := q_{j+1} - q_j = k(j-1) + b$, $\alpha = \frac{\Delta_{\lceil k/2 \rceil}}{\Delta_1}$, $q_{j+1} - q_j = q_{k-j+1} - q_{k-j}$, $\sigma_j = k(j-1) + \sigma_1$, $\sigma_j = \sigma_{k-j+1}$ for $j = 1, 2, ..., \lceil k/2 \rceil$.
- (NRLG) Relative linear Gaussian weight: two hyperparameters α, β. The only difference from the previous one is that we set the interception proportional to the sequence length, i.e. $\sigma_j = k(j-1) + \beta T$, $\sigma_j = \sigma_{k-j+1}$ for $j = 1, 2, ..., \lceil k/2 \rceil$. Note that k is computed by non-equidistant downsampling hyperparameter α.

Finally, preliminary results suggest that including length information can further improve the quality of the embeddings. To this end, we compute the $L2$ norm (approximate the energy for each frame) for each log mel feature vector, and cut off the sequence to 150 frames if larger, pad with 0 if smaller than 150 frames. After scaling the $L2$ norms with a trade-off hyperparameter γ, we concatenate them to the downsampled embedding. Our results suggest that with this $L2$ norm, we achieve an average gain on ABX error of 6.09% and 1.01% for Xitsonga and English respectively. Downsampling each sequence of log mel features to 20 samples (each the output of 40-dimensional log mel filter banks), then concatenating these samples along with the $L2$ norm, leads to a final embedding size of 950.

3.2. Recurrent autoencoders

RNNs, in particular Long-Short Term Memory networks (LSTM) [24], are the basis of state of the art solutions for numerous speech processing tasks. LSTMs can learn fixed-size

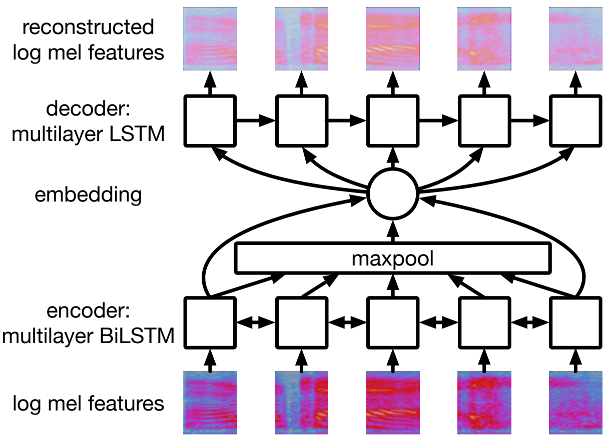

Figure 2: *RNN-based autoencoders used to learn word embeddings. For simplicity, the positional embeddings are left out.*

representations of variable-length sequences, in text [25, 26] and speech processing [9, 10]. Our proposed model, shown in figure 2, follows the encoder-decoder architecture used in [9], frequently used in machine translation [27], and borrows some ideas from segmental ASR.

Let $\boldsymbol{x_t}$ be the t-th 40-dimensional log mel feature vector, concatenated with 3 context frames on either side, a total of 7 concatenated frames (thus $\boldsymbol{x_t} \in \mathbb{R}^{280}$). The encoder, a multilayer bidirectional LSTM [28], reads a sequence $\boldsymbol{x_1}, ..., \boldsymbol{x_T}$ and encodes it into a sequence of hidden states $\boldsymbol{h_1}, ...\boldsymbol{h_T}$, where $\boldsymbol{h_t} = [\boldsymbol{h_t^{\text{forward}}}, \boldsymbol{h_t^{\text{backward}}}]$ and $[\boldsymbol{a}, \boldsymbol{b}]$ is the concatenation of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. We extract 3 representations from the encoder. The last hidden state of the forward LSTM $\boldsymbol{h_T^{\text{forward}}}$; the last hidden state of the backward LSTM $\boldsymbol{h_1^{\text{backward}}}$; an element-wise maxpooling along the time axis of the sequence of hidden states, $\boldsymbol{v} = \text{maxpool}_t \boldsymbol{h_t}$ so that $[\boldsymbol{v}]_i = \max_t [\boldsymbol{h_t}]_i$. These are concatenated and used as a representation of the full sequence: $\boldsymbol{c} = [\boldsymbol{h_T^{\text{forward}}}, \boldsymbol{h_1^{\text{backward}}}, \boldsymbol{v}]$. The vector $\boldsymbol{c}$ is fed to an LSTM decoder at every time step t, along with a learned embedding $\boldsymbol{e_t} \in \mathbb{R}^{300}$ of the integer t. The decoder's input at time step t is thus $[\boldsymbol{c}, \boldsymbol{e_t}]$, and its target output is $\boldsymbol{x_t}$. We train the network with Mean Squared Error (MSE) loss: $\sum_{t=1}^{T} ||\boldsymbol{x_t} - \boldsymbol{\hat{x}_t}||^2$ where $\boldsymbol{\hat{x}_t}$ is the decoder's output at time step t. The network is trained on segments of no more than 150 frames.

The training data has a very large amount of samples (English: 1.7M, Xitsonga: 653K), making it impractical to evaluate the neural networks on the timescale of epochs. Instead, we set aside 10% of the training data as a validation set. During training, we randomly take 100 samples from the training data and perform gradient descent using MSE loss and the Adam optimizer [29] with standard settings on this batch. After 100 training batches (i.e. 10^4 training samples), we randomly take 1000 samples from the validation set and use them to measure validation MSE. We use the neural net with the lowest validation MSE as our final model. The larger the model, the lower the MSE, so we must use the ABX score to compare architectures.

3.3. Evaluation

Given the broad scope of applications for low-resource speech technologies, and to develop methods fairly independent of their final application, we chose to evaluate the quality of embeddings intrinsically. We evaluate the learned embeddings with ABX discrimination tasks [30] across speakers. We sample triplets of (sequence A, sequence B, sequence X) where each sequence is a sequence of acoustic features; A and B were said by speaker 1; X was said by speaker 2, and the label associated with X (e.g. ”cat”, or ”ow-k”) is identical to that of either A or B. In this paper, the label is either a word or an n-gram of phonemes. Using the method we wish to evaluate, we compute embeddings $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{x}$ from sequences A, B and X. Using distance metric d, we compute pairwise distances between $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{x}$. We report two different metrics for this task.

- In the first setting, denoted *ABX*, sequences A, B and X are words. We look at the classification task of predicting whether X shares its label y_X with A or B. We predict

$$\hat{y}_X = \begin{cases} y_A & \text{if } d(\boldsymbol{a}, \boldsymbol{x}) < d(\boldsymbol{b}, \boldsymbol{x}) \\ y_B & \text{otherwise} \end{cases}$$

and report error on this classification task. Triplets are sampled such that the performance of random guessing is 50%.

- In the second setting, denoted *AUC*, there is no X; sequences A and B correspond to phoneme n-grams where $n = 2...12$. We are still looking at a classification task, with

$$\hat{y} = \begin{cases} \text{"same"} & \text{if } d(\boldsymbol{a}, \boldsymbol{b}) < \theta \\ \text{"different"} & \text{otherwise} \end{cases}$$

We look at the ROC curve obtained by measuring precision and recall for various values of θ. The score reported is the area under the ROC curve. Higher is better, and means less sensitivity with respect to parameter θ [31].

For DTW, embedding $\boldsymbol{x}$ is actually X itself, and d is the DTW cosine distance with or without length normalization. For downsampling, d is the cosine distance, shown to work best for log mel features. For neural networks, d is the Euclidean distance, as the most natural distance metric over a vector space.

3.4. Datasets

We used the English and Xitsonga datasets provided by the Zero Resource Speech Challenge 2015 [32] to design and evaluate our models. The English corpus is a subset of the Buckeye corpus [33] (denoted as ”English” in the following), and the Xitsonga corpus is the NCHLT Xitsonga Speech Corpus [34] (denoted as ”Xitsonga” in the following). Speech is represented as a sequence of 40-dimensional log mel feature vectors, computed on a sliding window of 25 ms, offset by 10 ms. The filter banks were mean and variance normalized per file (excluding silences), to help remove background noise and reduce across-speaker variability. Here are, in ms, average and standard deviation for the duration of English words: 236.2 and 153.6; English 2-to-12-grams: 547.7 and 500.1; Xitsonga words: 450.8 and 290.4; Xitsonga 2-to-12-grams: 617.4 and 327.2.

4. Results and discussion

For each downsampling method, we have conducted several grid searches. We kept the parameters achieving the best average ABX error on both corpora, since we would like parameters which can generalize well enough across different corpora. As shown in Table 1, although the best methods for English and for Xitsonga are different, the Equidistant downsampling with Absolute Linear Gaussian weight (EALG) method, which achieves the best average ABX, performs close enough to the best method on both corpora.

We investigated varying the depth and the number of hidden units of the recurrent autoencoders. Note that the size of the embedding is equal to twice the number of hidden units. Results from table 1 show, across both languages, that the larger the size

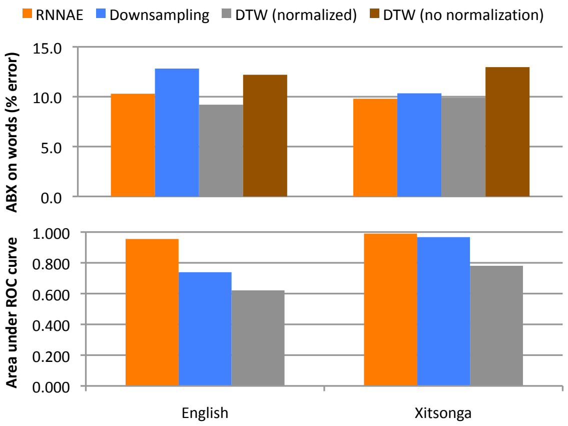

Figure 3: *ABX error (top) and AUC score (bottom) for neural networks (RNNAE), downsampling and DTW baselines. We report our best model for each task.*

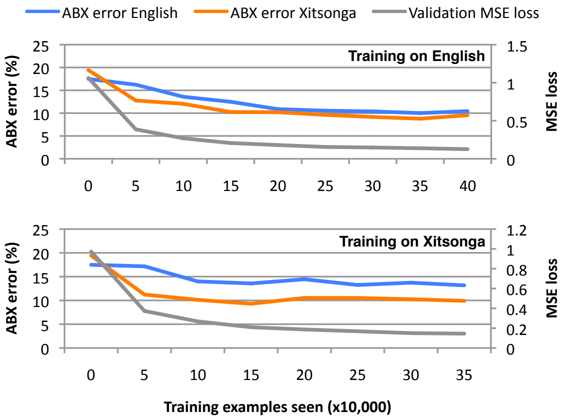

Figure 4: *ABX error on words for English and Xitsonga corpora, and MSE loss on the validation set, as a function of the number of training steps, for a neural net trained on the English (top) or Xitsonga corpus (bottom). Architecture of the encoder, mirroring the decoder, is 1000-1000-1000-400.*

of the embedding, the higher the ABX score and AUC. Depth of the network does not correlate with better scores.

On the AUC task, on both corpora, we find that recurrent autoencoder embeddings perform best, and downsampling outperforms normalized DTW by a comfortable margin. For the ABX task, normalized DTW and our methods perform somewhat on par on the Xitsonga corpus, while normalized DTW still performs best on the English corpus.

We further explored the amount to which the neural nets specialize to the language of the training data as training progresses. Figure 4 shows how the ABX score progresses on both English and Xitsonga datasets, as the MSE loss decreases on the validation set. Note that ABX error decreases even for mismatched tests sets (across a language change). This could be due to the fact that Xitsonga and English share some of their phonetic inventory, or that recurrent autoencoders are learning generally useful speech features, not necessarily tailored to the training language. Downsampling has a similar behavior, as it works consistently well across languages.

Table 1: *ABX error (%) and AUC score. "AE" means RNN autoencoder, with the architecture of the encoder indicated as the number of hidden units per layer. The decoder's architecture mirrors the encoder's.*

		English		Xitsonga	
model	hyperparameters	ABX	AUC	ABX	AUC
DTW	baseline, no length normalization	12.2	.621	13.0	.781
DTW	baseline, length normalized	9.2	.621	9.9	.781
AE	300-300-300-300	16.2	.849	12.1	.883
AE	400 - 400 - 400 - 400	12.3	.924	12.0	.947
AE	600 - 600 - 600 - 600	12.8	.934	13.0	.978
AE	1000 - 1000 - 1000 - 1000	12.9	.901	10.0	.981
AE	1000 - 1000 - 1000	**10.3**	.935	**9.8**	.974
AE	1000 - 1000	14.6	.951	10.0	.981
AE	1000 - 1000 - 1000 - 400	10.4	**.955**	9.9	**.990**
EPI	downsampling baseline	13.1	.684	10.6	.945
EAG	$\sigma = 1.6, \gamma = 0.4$	13.0	.665	10.4	.943
ERG	$\beta = 0.03, \gamma = 0.4$	13.0	.683	10.4	.914
EALG	$k = 0.2, \sigma_1 = 0.4, \gamma = 0.4$	**12.8**	.707	10.4	.961
ERLG	$k = 0.07, \beta = 0.03, \gamma = 0.4$	13.0	**.739**	10.4	.960
NPI	$\alpha = 0.9, \gamma = 0.4$	13.2	.657	10.5	**.967**
NAG	$\alpha = 1.0, \sigma_1 = 1.5, \gamma = 0.4$	13.0	.665	10.4	.958
NRG	$\alpha = 1.1, \beta = 0.02, \gamma = 0.4$	13.1	.669	**10.3**	.912
NALG	$\alpha = 0.8, \sigma_1 = 1.4, \gamma = 0.4$	13.2	.709	10.4	.966
NRLG	$\alpha = 1.1, \beta = 0.02, \gamma = 0.4$	13.0	.696	10.4	.963

5. Conclusion

In this paper, we propose two parametric methods to extract fixed-size embeddings from variable-length sequences of acoustic frames. The geometry of these embeddings captures linguistic properties, sometimes better than the geometry of the original acoustic space, as measured by DTW. We have thoroughly explored different strategies of downsampling, a method with few parameters, fast to compute and robust across languages. RNNs, trained with an unsupervised objective, and without any gold word-level segmentation of the input, manage to capture acoustic regularities, and further improve over downsampling. Using larger embeddings allows these models to fit the acoustic regularities even better, with higher ABX and AUC scores for larger networks. However, these embeddings come at a higher computational cost than downsampling.

Interestingly, one can find, for each method, hyperparameters which give very good results for two very different languages. The RNN autoencoders even show some ability to generalize across languages after training. This suggests a possibility to precompile language-general spoken word embedders. In future work, it would be interesting to further improve the RNN autoencoders by training them as denoising autoencoders, using data augmentation such as adding reverberation, and encouraging the autoencoder to ignore speaker information, e.g. with a well-designed loss function.

6. Acknowledgements

This project is funded by the European Research Council (ERC-2011-AdG-295810 BOOTPHON), the Agence Nationale pour la Recherche (ANR-10-LABX-0087 IEC, ANR-10-IDEX-0001-02 PSL*), Microsoft Research (joint MSR-INRIA center) and a Google Award Grant.

7. References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[2] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543.

[3] A. G. I. Ororbia, T. Mikolov, and D. Reitter, "Learning simpler language models with the differential state framework," *Neural Computation*, vol. 29, no. 12, 2017.

[4] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning, "Bilingual word embeddings for phrase-based machine translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1393–1398.

[5] Z. Cao, S. Li, Y. Liu, W. Li, and H. Ji, "A novel neural topic model and its supervised extension," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 2210–2216.

[6] A. Marasovic, L. Born, J. Opitz, and A. Frank, "A mention-ranking model for abstract anaphora resolution," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 221–232.

[7] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Interspeech 2017, ISCA, Stockholm, Sweden, August 20-24, 2017*, pp. 2874–2878.

[8] H. Kamper, A. Jansen, and S. Goldwater, "Unsupervised word segmentation and lexicon discovery using acoustic word embeddings," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 4, pp. 669–679, 2016.

[9] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," in *INTERSPEECH*, 2016.

[10] Y.-A. Chung and J. Glass, "Learning word embeddings from speech," in *NIPS ML4Audio Workshop*, 2017.

[11] H. Kamper, A. Jansen, and S. Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Computer Speech & Language*, vol. 46, pp. 154–174, 2017.

[12] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2009, Merano/Meran, Italy, December 13-17, 2009*, pp. 152–157.

[13] J. R. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech & Language*, vol. 17, no. 2-3, pp. 137–152, 2003.

[14] M. Ostendorf and S. Roukos, "A stochastic segment model for phoneme-based continuous speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 1857–1869, Dec 1989.

[15] H. Tang, L. Lu, L. Kong, K. Gimpel, K. Livescu, C. Dyer, N. A. Smith, and S. Renals, "End-to-end neural segmental models for speech recognition," *J. Sel. Topics Signal Processing*, vol. 11, no. 8, pp. 1254–1264, 2017. [Online]. Available: https://doi.org/10.1109/JSTSP.2017.2752462

[16] O. Abdel-Hamid, L. Deng, D. Yu, and H. Jiang, "Deep segmental neural networks for speech recognition," in *INTERSPEECH 2013, ISCA, Lyon, France, August 25-29, 2013*, 2013, pp. 1849–1853. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2013/i13_1849.html

[17] Y. He and E. Fosler-Lussier, "Segmental conditional random fields with deep neural networks as acoustic models for first-pass word recognition," in *INTERSPEECH 2015, ISCA, Dresden, Germany, September 6-10, 2015*, 2015, pp. 2640–2644. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2015/i15_2640.html

[18] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[19] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *CoRR*, vol. abs/1705.02304, 2017.

[20] A. L. Maas, S. D. Miller, T. M. O'Neil, A. Y. Ng, and P. Nguyen, "Word-level acoustic modeling with convolutional vector regression," *ICML 2012 Workshop on Representation Learning, Edinburgh, Scotland, UK*, 2012.

[21] D. Harwath and J. R. Glass, "Learning word-like units from joint audio-visual analysis," in *Proceedings of the 55th Annual Meeting of ACL, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 506–517.

[22] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*, 2013, pp. 410–415.

[23] K. Vintsyuk, "Speech discrimination by dynamic programming," *Cybernetics and Systems Analysis*, vol. 4, no. 1, pp. 52–57, 1968.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 3104–3112.

[26] H. Schwenk, K. Tran, O. Firat, and M. Douze, "Learning joint multilingual sentence representations with neural machine translation," *CoRR*, vol. abs/1704.04154, 2017.

[27] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1724–1734.

[28] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[30] G. Synnaeve, T. Schatz, and E. Dupoux, "Phonetics embedding learning with side information," in *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014*, 2014, pp. 106–111.

[31] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," Tech. Rep., 2004.

[32] M. Versteegh, R. Thiollière, T. Schatz, X. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," in *INTERSPEECH 2015, ISCA, Dresden, Germany, September 6-10, 2015*, 2015, pp. 3169–3173.

[33] M. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, "Buckeye corpus of conversational speech (2nd release)," 2007.

[34] N. J. de Vries, M. H. Davel, J. Badenhorst, W. D. Basson, F. de Wet, E. Barnard, and A. de Waal, "A smartphone-based ASR data collection tool for under-resourced languages," *Speech Communication*, vol. 56, pp. 119–131, 2014.