Contextualized Spoken Word Representations Using Convolutional Autoencoders

Prakamya Mishra
Shiv Nadar University
`pm669@snu.edu.in`

Pranav Mathur
Shiv Nadar University
`pm872@snu.edu.in`

Abstract

A lot of work has been done to build text-based language models for performing different NLP tasks, but not much research has been done in the case of audio-based language models. This paper proposes a Convolutional Autoencoder based neural architecture to model syntactically and semantically adequate contextualized representations of varying length spoken words. The use of such representations can not only lead to great advances in the audio-based NLP tasks but can also curtail the loss of information like tone, expression, accent, etc while converting speech to text to perform these tasks. The performance of the proposed model is validated by (1) examining the generated vector space, and (2) evaluating its performance on three benchmark datasets for measuring word similarities, against existing widely used text-based language models that are trained on the transcriptions. The proposed model was able to demonstrate its robustness when compared to the other two language-based models.

1 Introduction

There are several methods in which humans and computers can converse, like speaking (audio) and writing (text). At present, research in the field of NLP has advanced a lot to attain a good understanding of textual data but there are still some ways to go to properly contemplate the audio/speech data.

Word embeddings are extensively used in NLP applications since they have proven to be an extremely informative representation of the textual data. Language models like GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013) successfully transform textual words from its raw form to semantically and syntactically correct, fixed dimensional vectors. These type of word representations for the spoken words can

be widely used to process speech/audio data for tasks like Automatic Summarization (Kågebäck et al., 2014), Machine Translation (Jansen, 2017), Named Entity Recognition (Wen et al., 2020), Sentiment Analysis (Liu, 2017), Information Retrieval (Rekabsaz et al., 2017), Speech Recognition (Palaskar et al., 2019), Question-Answering (Tapaswi et al., 2016) etc.

Compared to text, not much research has been done in the field of audio-based modeling primarily due to the lack of large, reliable, clean, and publicly available datasets on which the spoken word language models can be trained. Also, spoken words unlike textual words have a different meaning when they are spoken in a different tone, expression, accent, etc, and incorporating them exponentially increases the difficulty of building such language models. Such models also face difficulties such as different people can have different pronunciations, tones, and pauses for the exact same words.

The proposed model, aims at generating syntactically and semantically adequate contextualized vector representation of the variable length audio files (instead of using fixed length audio files with multiple word utterances), where each file corresponds to a single spoken word in a speech and further validates the vector representations by evaluating it on three benchmark word similarity datasets (SimVerb, WS-SIM, WS-REL). To further increase the interpretability, this paper also provides illustrations of the vector space generated by the proposed model.

2 Related Work

A lot of work has been done in the field of NLP to give textual words sound representations. Word2Vec (Mikolov et al., 2013) has demonstrated huge improvements in embedding sub-

linear relationships into the vector space of the words but at the same time, they were unable to handle out of vocabulary words. Another comparable word representation model is GloVe (Pennington et al., 2014). GloVe works to fit a giant word co-occurrence matrix built from the matrix. GloVe helps in taking into account the semantics and also gives relatively smaller dimension vectors.

Recent advances have enabled it to apply deep learning to transform spoken word segments into fixed dimensional vectors. (Chung et al., 2016), uses fixed-length audio files and passes them through a Sequence-to-Sequence Autoencoder (SA) and Denoising SA (DSA) to generate word embeddings. They demonstrated that the phonetically similar words had close spatial representations in the vector space but they failed to meet the result standards similar to those by GloVe trained on Wikipedia. Following the above work, (Chung and Glass) used 500 hours of speeches from multiple speakers divided into fixed audio segments. They compare the results also with GloVe based on 13 different comparison measures. Both, (Chung et al., 2016) & (Chung and Glass) failed to capture the spoken words properly due to the use of fixed length audio segments. Tagliasacchi et al. (2020), proposed an audio2vec model which was built on top of the Word2Vec models (Skip-gram & CBOW) to reconstruct spectrogram slices using the contextual slices and temporal gaps. They were able to show that Audio2Vec performed better than the other existing fully-supervised models.

3 Model

The proposed model uses sequential utterances of words from a speech to learn their corresponding contextualized representations. These learned contextualized representations capture the semantic and syntactic properties of these spoken words. The input to the model is a speech S. This speech is split into individual spoken word utterances (independent variable-length audio files). The proposed model used audio spectrograms for representing the audio files of these spoken word utterances. An audio spectrogram is a visual representation of sound. So to get spectral representations, all the spoken word utterances are converted to their corresponding spectrograms (which depicts the spectral density of a sound *w.r.t* time (in our

case an utterance)). The spoken word utterance spectrograms are represented by W_i as shown in equation 1.

$$S = [W_1, W_2, ..., W_n], n \in \mathbb{R} \qquad (1)$$

In the above equation n represents the total number of spoken words present in a sentence of the speech and $W_i \in \mathbb{R}^{l_1 l_2}$ represents a spectrogram, where, l_1 is for the frequency (pitch/tone) dimension, l_2 represents time. Values in the spectrogram represents amplitude (energy/loudness) at a particular time of a particular frequency.

Words have different meanings when they are spoken in different contexts. To capture the context corresponding to spoken words, the proposed model uses a context window of size m. So the representation of a spoken word (target word) is learned based on m spoken words after and before it. This context window of size m slides over the whole speech having a target spoken word W_t (where $1 \leq t \leq n$) at the middle and m context spoken words before and after it (a total of 2m context words). These context spoken words are represented by W_{t+j} where $-m \leq j \leq m$ & $j \neq 0$.

Next, the model passes all the pairs of the target spoken word spectrograms W_t with its corresponding context spoken word spectrograms W_{t+j} into a convolutional autoencoder individually to learn the contextual representation of the target spoken word corresponding to W_t. The convolutional autoencoder is composed of two independent neural networks namely, an encoder network and a decoder network. The encoder network is represented by f_ϕ and the decoder network is represented by g_θ, where ϕ and θ are the learnable parameters corresponding to both the networks. Both f_ϕ & g_θ are used to extract the spatial features of the input spectrogram *w.r.t* to the output spectrogram. The target spoken word spectrogram W_t is given as input to the encoder network, which outputs a latent representation h. This latent representation is then given as input to the decoder network, i.e.

$$h = f_\phi(W_t) = \sigma(W_t * \phi) \qquad (2)$$

$$W_{t+j} = g_\theta(h) = \sigma(h * \theta) \qquad (3)$$

$$W_{t+j} = g_\theta(f_\phi(W_t)) \qquad (4)$$

In the equations 2 & 3, $(*)$ represents the convolution operator, σ is the *Leaky*ReLu activation function. The encoder network f_ϕ, consist of two
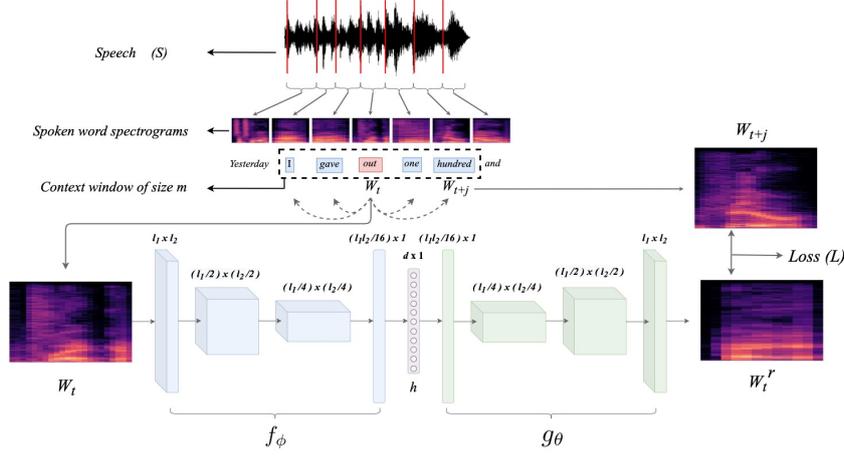
Figure 1: Proposed Model Architecture

convolutional layers on top of the input spectrogram. These convolutional layers are used for extracting hierarchical location invariant spatial features. The output of the last convolutional layer in f_ϕ is then flattened and passed to a d-dimensional dense layer (h). This dense layer (h) is the embedding layer which learns the contextual representation of the spoken word corresponding to input W_t (contextualized on the context spoken word spectrograms). The decoder network takes the embedding layer (h) as input and generates a reconstruction W_t^r by passing (h) through a dense layer and two transpose convolutional layers. The d-dimensional embedding layer (h) learns an efficient contextualized representation of the word corresponding to W_t by minimizing the loss function L (shown in equation 5). In the equation below, N represents the batch size and m represents the size of the context window.

$$L_{\phi,\theta} = \frac{1}{N} \sum_{t=1}^{N} \left(\frac{1}{2m} \sum_{j=-m;j\neq0}^{m} (\Delta_{W_t, W_{t+j}}) \right) \quad (5)$$

where,

$$\Delta_{W_t, W_{t+j}} = ||g_\theta(f_\phi(W_t)) - W_{t+j}||_2^2 \quad (6)$$

The lost function defined above helps the latent embedding to learn the contextual relationship between the target spoken word spectrograms and it's the corresponding context by calculating a reconstruction loss between the reconstruction W_t^r and the corresponding contextual spectrograms W_{t+j}. Since a word spoken in different tones has different spectrograms, the model also captures the tone in which the words are uttered in its contextual embedding. So in summary the proposed model can not only incorporate context

in the spoken word representations but can also incorporate its tone.

4 Evaluation Setup

4.1 Dataset

The proposed model uses Trump's speeches (Audio and word transcription)[1] dataset for training and testing. This dataset was chosen because it comprises of audio files and their corresponding word split JSON files. Another reason for choosing this dataset was that it contains speeches of a single person (which will eliminate the problem of having different pronunciations of the same word). These JSON files contain a direct mapping between each word spoken and the duration in which it was spoken. These mappings were used to split the full audio file into multiple audio files for each word spoken. This context mapping was used to create input-output pairs for the proposed model. The statistics about the dataset is shown in Table 1.

Table 1: Dataset Statistics

# Words	# Sentences	# Context Mappings	# Seconds
18.1k	1K	72.6K	12.9K

4.2 Training Details

The proposed model was evaluated on 10% of the data, and the rest was used for training. From the training set, 10% data was used as the validation set. It was trained for 50 epochs having a mini-batch size of 5. For optimization, Adam optimizer

---

[1]https://www.kaggle.com/etaifour/trump-speeches-audio-and-word-transcription

was used to have an initial learning rate of 0.01. A context window size of 2 was used for all the experiments (due to computational resource limitations). Early stopping with the patience of 5 epochs and dropout with a dropout rate of 0.7 was used to avoid over-fitting. The size of the latent representation d was set to 16. The size of the filters in the convolutional and de-convolutional layers was set to (4×4).

4.3 Results

The performance of the proposed model was validated by (1) inspecting the vector space and (2) evaluating its performance on three benchmark datasets for measuring word similarities, and comparing the proposed model with text-based language models (trained on the textual transcripts).
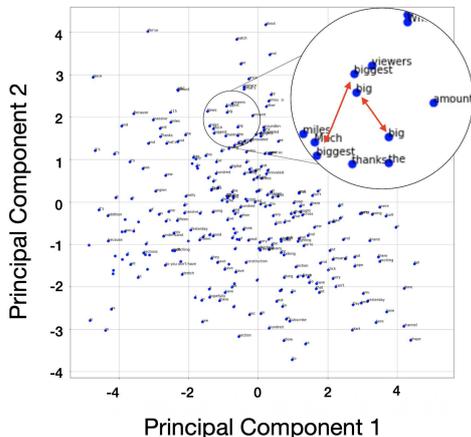

Figure 2: Vector space generated by the proposed model

To visualize the performance of the proposed model, the dimensionality of the audio vectors (16-dimensional) was reduced using principal component analysis (PCA) (Hotelling, 1933), to plot the spoken word representations in a (2-dimensional) vectors space. Figure 2 illustrates the vector space generated by the proposed model. On closeup, it can be easily seen that similar spoken words were grouped together in the vector space. For example, spoken words like *big*, *biggest*, and *much* fall in vicinity to each other. It can also be seen in figure 2 that the same spoken words (*big* & *biggest*), uttered in different tones were also in close proximity but were slightly distant from each other. This demonstrates the capability of the model to capturing semantical and syntactical similarities between different spoken words (or the same spoken word in different tones).

Table 2: Results Table

Dataset	# Word Pairs	Spearmans rank correlation coefficient ρ		
		Our Model	Word2Vec	GloVe
SimVerb	275	0.31	**0.32**	0.28
WS-SIM	33	**0.51**	0.49	0.47
WS-REL	53	0.23	**0.25**	**0.25**

The spoken word representations generated by the proposed model were evaluated on three different benchmark datasets (**SimVerb** (Gerz et al., 2016), **WS-SIM** and **WS-REL** (Agirre et al., 2009)) that are widely used for computing word similarities/relatedness between words. The comparison of the proposed model is done with the text-based language models Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). In the case of the proposed model, word similarities were obtained by measuring the cosine similarities between the spoken vector representations of the corresponding words, and in the case of Word2Vec & GloVe, similarities were computed between the corresponding word (textual) vector representations. Table 2 reports Spearmans rank correlation coefficient ρ between the human ranking (Myers et al., 2010) and the ones generated by each model. The proposed model was trained on a small dataset (small vocabulary). So the proposed model was not able to generate representations for some of the word pairs present in the above mentioned three benchmark datasets (Number of word pairs used is also shown in the table above). Despite spoken words having different tones/expressing/pause for the same words depending on the context (in contrast to text), the proposed model was able perform comparably to the existing text-based language models.

5 Conclusion

This paper introduces an unsupervised model that not only was able to successfully generate semantically and syntactically accurate contextualized representations of varying length spoken words but was also able to perform adequately on three benchmark datasets for measuring word similarities. The proposed model also demonstrated its capabilities to capture tones and expressions of the spoken words. To the best of our knowledge, this is the first work that tries to model variable-length spoken words using convolutional autoencoders. In the future, we plan to extend the capabilities of the model to handle different pronunci-

ations/accent by different speakers.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Yu-An Chung and James Glass. Learning word embeddings from speech. In *Workshop Machine Learning for Audio Signal Processing at NIPS (ML4Audio@NIPS17)*.

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*.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.

Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Stefan Jansen. 2017. Word and phrase translation with word2vec. *CoRR*, abs/1705.03127.

Mikael Kågebäck, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. 2014. Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 31–39, Gothenburg, Sweden. Association for Computational Linguistics.

Haixia Liu. 2017. Sentiment analysis of citations using word2vec. *CoRR*, abs/1704.00177.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jerome L Myers, Arnold Well, and Robert Frederick Lorch. 2010. *Research design and statistical analysis*. Routledge.

Shruti Palaskar, Vikas Raunak, and Florian Metze. 2019. Learned in speech recognition: Contextual acoustic word embeddings. *CoRR*, abs/1902.06833.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Navid Rekabsaz, Bhaskar Mitra, Mihai Lupu, and Allan Hanbury. 2017. Toward incorporation of relevant documents in word2vec. *CoRR*, abs/1707.06598.

M. Tagliasacchi, B. Gfeller, F. d. C. Quitry, and D. Roblek. 2020. Pre-training audio representations with self-supervision. *IEEE Signal Processing Letters*, 27:600–604.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yan Wen, Cong Fan, Geng Chen, Xin Chen, and Ming Chen. 2020. A survey on named entity recognition. In *Communications, Signal Processing, and Systems*, pages 1803–1810, Singapore. Springer Singapore.