Leveraging Latent Representations of Speech for Indian Language Identification

Samarjit Karmakar *
Microsoft IDC, Hyderabad
Telangana, India
karmakar.samarjit@gmail.com

P. Radha Krishna
Department of CSE
National Institute of Technology, Warangal
Telangana, India
prkrishna@nitw.ac.in

Abstract

Identification of the language spoken from speech utterances is an interesting task because of the diversity associated with different languages and human voices. Indian languages have diverse origins and identifying them from speech utterances would help several language recognition, translation and relationship mining tasks. The current approaches for tackling the problem of languages identification in the Indian context heavily use feature engineering and classical speech processing techniques. This is a bottleneck for language identification systems, as we require to exploit necessary features in speech, required for machine identification, which are learnt by a probabilistic framework, rather than handcrafted feature engineering. In this paper, we tackle the problem of language identification using latent representations learnt from speech using Variational Autoencoders (VAEs) and leverage the representations learnt to train sequence models. Our framework attains an accuracy of 89% in the identification of 8 well known Indian languages (namely Tamil, Telugu, Punjabi, Marathi, Gujarati, Hindi, Kannada and Bengali) from the CMU/IIITH Indic Speech Database. The presented approach can be applied to several scenarios for speech processing by employing representation learning and leveraging them for sequence models.

1 Introduction

Language identification refers to the task of identifying the language being spoken when given a speech utterance. Several intelligent agents rely heavily on language identification systems for subsequent speech recognition and processing tasks. This problem is particularly interesting and important in the Indian context, given the diverse nature of Indian languages. Several Indian languages suffer regional bias and each language has its own dialect as one travels within each state (region) of the country. In this scenario, language identification would be a challenging task for traditional language identification systems which heavily rely on feature engineering from speech.

Several of the current Indian language identification systems rely on handcrafted features, which end up serving as a bottleneck to such systems. Such systems would greatly benefit by learning necessary features in speech utterances using a probabilistic framework and leveraging these representations for training deep sequence models. A few deep neural network based models have also been proposed. The authors in (Lei et al., 2014) perform posterior extraction using convolutional neural networks (CNNs), and an i-vector based system for subsequent language recognition. The authors in (MounikaK. et al., 2016) use an end-to-end deep neural network with attention mechanism for Indian language identification.

We exploit the representation learnt by a VAE (Kingma and Welling, 2013) trained on speech segments to train several sequence models widely used for natural language processing. These models use distributed word-representations (Mikolov et al., 2013) which model the words in a continuous vector space. We use the latent feature representations learnt by a VAE in the stochastic low dimensional latent representation space in a manner similar to how distributed word representations, obtained by pre-training large corpuses in an unsupervised manner, are used in natural language processing to train sequence models.

In this paper, we present a framework for Indian language identification by pre-training a probabilistic framework for representation learning and leveraging these representation for training sequence models for classification.

This work was a part of the authors' undergraduate thesis project at Dept of CSE, NIT Warangal.

Proceedings of the 17th International Conference on Natural Language Processing, pages 334–340
Patna, India, December 18 - 21, 2020.

2 Related Work

Language identification from speech has been deeply studied by various research communities. Prosodic, phoenetic and phonotactic feature based approaches for identifying language is studied in (Tong et al., 2006) and (Liang Wang et al., 2006). Many such classical feature engineering based methods require a lot of domain knowledge. With the rise of deep learning and neural networks, automatic feature and representation learning has greatly outperformed all such methods.

Language identification using Deep Convolutional Recurrent Neural Networks is studied in (Bartz et al., 2017). They use non-overlapping segments of Mel spectrograms of speech which are passed through a Convolutional Neural Network and then the features maps are passed through a Long Short Term Memory (LSTM). The final hidden state of the LSTM is used for classification. The CNN captures the spatial features, whereas the LSTM captures the temporal features.

For long speech utterances, Recurrent Neural Networks, can capture the temporal aspect of speech utterances and this was considered in (Gonzalez-Dominguez et al., 2014). The authors in (Sarthak et al., 2019) give an attention based 1D-CNN for the task of language identification directly from raw audio. This attention greatly enhances the performance of neural network based approaches.

Indian language identification using deep learning based models have been studied in (Leena et al., 2005), (MounikaK. et al., 2016), (Thirumuru et al., 2018) and (Bakshi and Kopparapu, 2017). Deep neural network based systems take in the speech utterances at each frame, classification performed frame-wise, and this may be considered as a drawback. A deep neural network with attention mechanism was considered in (MounikaK. et al., 2016). This architecture applies attention to specific parts of the input sequence, whilst memorizing important features in long temporal sequences. A 39-dimensional MFCC is considered by the authors, each for 5 second chunks of the input sequence, which are passed through a regular DNN to compute hidden layer representations. An attention mechanism is applied over this to memorize the temporal aspect and summarize the features in the whole speech utterance, giving a single context vector and this vector is subsequently passed to a classifier. Attention based Residual-Time Delay Neural Network (RES-TDNN) is studied in (Mandava and Vuppala, 2019), which further improves over trying to capture the long range temporal dependencies.

3 Proposed Framework

Our goal is to learn latent representations from speech and use these representations to train sequence models for classification. We use Variational Autoencoders (VAEs) for representation learning on small segments of Mel spectrograms of speech utterances (40 Mel-scale filter banks). The Mel spectrogram is obtained by taking the Fourier transform of the signal, followed by mapping the powers of the obtained spectrum onto the Mel scale. The Mel-frequency scale resembles the resolution of the human auditory system. The segmentation is performed along the time axis. The model is trained in a similar manner adopted in (Hsu et al., 2017). After pre-training the VAE, the encoder's latent distribution is able to encode Mel spectrograms of speech segments into a latent representations space. We use a sequence of such latent representation for each segmented Mel spectrogram of speech utterance as input to sequence models. The VAE captures important representational features for each segment of the speech utterance and the sequence model captures the temporal aspect of each speech utterance. The unsupervised representation learning parameters are optimized in a different step from when the supervised sequence learning parameters are optimized.

3.1 Variational Autoencoder

A Variational Autoencoder (VAE) comprises of two neural networks, the encoder $q_\theta(z|x)$ and the decoder $p_\phi(x|z)$. The encoder, parameterized by θ, takes in the input observation (x) and encodes it into a representation (z) sampled stochastically from the distribution of μ and σ (Gaussian parametric layers of the encoder). The decoder, parameterized by ϕ, takes in the representation (z) and decodes it back into the input observation (x). The loss function (L_{vae}) minimizes a joint objective of two losses: reconstruction loss and KL divergence loss.

$$L_{vae} = -\mathbb{E}_{q_\theta(z|x)}(p_\phi(x|z)) + \mathbb{KL}(q_\theta(z|x)||p(z)) \quad (1)$$

Here, $p(z)$ is the prior distribution (multivariate standard Normal).

We use similar hyper-parameters as used in (Hsu et al., 2017). The encoder contains 3 convolutional

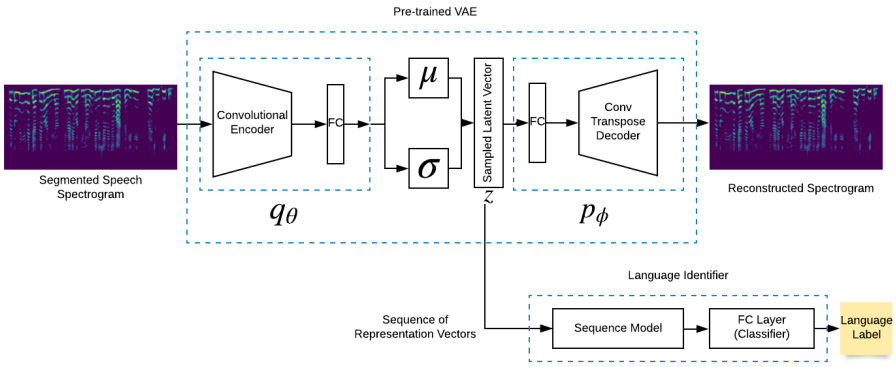


Figure 1: A view of the VAE architecture with the language identifier.

layers, followed by a fully connected layer and two Gaussian parametric layer (one for mean and another for log variance). The decoder contains an initial fully connected layer which takes in z, followed by another fully connected layer and 3 transpose-convolutional layers.

Figure 1 shows a view of the VAE architecture along with the language identifier (successor to the VAE pre-training phase).

3.2 Sequence Models

Sequence models capture the temporal aspect of the speech utterance from the given sequence of representation vectors.

For each segment of the Mel spectrogram of speech utterance, the VAE encoder produces a vector in $\mathbb{R}^n$, where n is the dimension of the representation space. We pass the sequence of these vectors for each segmented speech utterance to sequence models for classification. The input to the sequence models are a sequence representation vectors of size 128 units, i.e the dimension of the representation space of the VAE. We compute the maximum length of the sequences produced on segmentation of each speech utterance, and apply zero-padding vectors to each sequence to produce uniform length sequences.

The sequence models considered in this work are illustrated in the next sections. All the models are trained separately on the same representation vectors obtained from the pre-trained VAE encoder.

3.2.1 Long Short Term Memory (LSTM) Networks and Bi-directional LSTMs

Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks are sequence models which learn temporal characteristics and contextual information from sequences.

LSTMs have a single shortcoming, they make use of solely the previous context. Bidirectional LSTMs (Bi-LSTMs) make use of both the previous context as well as future context by iterating through the sequence in both directions to compute the hidden state vectors. We pass the forward and backward sequence through the LSTM assigning different weights and biases for each direction. This is used to compute two separate sets of activations for the sequence in forward and backward directions.

3.2.2 Bi-directional LSTM with Self-Attention

The sequence of representation vectors is passed through a bi-directional LSTM (bi-LSTM) (Schuster and Paliwal, 1997) with a hidden state of 100 memory units. A self-attention mechanism is adopted which gives attention to specific parts of the input sequence, giving a attention weight matrix of the input sequence. This mechanism is similar to the attention layer in (Cheng et al., 2016). The mechanism takes in the hidden states of the bi-LSTM at each time step. The attention weights are calculated for 30 sequence vectors, each giving attention to some specific part of the input sequence. The attention weights are applied to the output of the bi-LSTM, resulting in a matrix of hidden states giving attention to specific parts of the sequence. This is then flattened and passed through subsequent fully connected neural network layers to produce the output class logits.

3.2.3 Bi-directional LSTM with Soft-Aligned Attention

We apply a similar attention mechanism as that adopted in the encoder of (Bahdanau et al., 2014). We pass the sequence of representation vectors through a bi-LSTM with a hidden state of 100

memory units similar to the previous model. The difference lies in the attention mechanism adopted. We calculate a soft-alignment score between each of the hidden states of the bi-LSTM at each time step and the last hidden state. This gives the soft-aligned attention weights, which are applied to the output of the bi-LSTM at each time step to produce a single final hidden state vector. This is then passed through subsequent fully connected neural network layers to produce the output class logits.

3.2.4 Recurrent Convolutional Neural Networks

We apply a similar architecture as that adopted in (Lai et al., 2015). We pass the sequence of representation vectors through a bi-LSTM with a hidden state of 100 memory units similar to the previous model. The hidden state at each time step is concatenated to the corresponding input representation. This is passed through a fully connected layer which maps the concatenated vector back to the hidden state size. The architecture takes care of the right context and left context as it is a bi-LSTM, which takes care of information and representation flow in the forward and reverse direction of the speech utterance. We perform max-pooling across all the sequences and pass the output through subsequent fully connected neural network layers to produce the output class logits.

3.2.5 Transformer

We use the encoder of the Transformer architecture similar to that adopted in (Vaswani et al., 2017). We do not apply the positional encodings and masking mechanisms. The sequence of representation vectors are directly passed through two encoder layers, each of which which comprise of self attention and position-wise feed-forward layers. The hidden dimensions of the position-wise feed-forward layers are 100 units. The set of hyper-parameters adopted similar to (Vaswani et al., 2017) are ($N = 2$, $d_{model} = 128$, $d_q = d_k = d_v = 32$, $p_{dropout} = 0.3$). The output of the encoder is flattened and passed through subsequent fully connected neural network layers to produce the output class logits.

An illustration for training and language identification (testing) of the VAE and Sequence Model is given in Algorithm 1 and Algorithm 2.

Algorithm 1 VAE-Seq Language Identification Training

Input: Dataset D, VAE parameters (ϕ, θ), Sequence Model parameters (ξ)
Output: Optimized parameters ϕ, θ and ξ
Initialize parameters ϕ, θ and ξ
repeat
  Sample mini-batch $M = \{x_i\}_{i=1,2,...\|M\|}$ of audio spectrograms from D by segmenting spectrogram of audio clips
  Forward pass mini-batch M through VAE
  Update parameters ϕ and θ using $\nabla_{\phi,\theta} L_{vae}(\phi, \theta, M)$
until convergence of ϕ and θ
repeat
  Sample mini-batch $M = \{x_i, y_i\}_{i=1,2,...\|M\|}$ from D where x_i is a sequence of segmented spectrograms and y_i is label for i'th sample
  Forward pass each sample in x_i through VAE encoder parameterized by θ to convert $\{x_i, y_i\}_{i=1,2,...\|M\|}$ to $\{v_i, y_i\}_{i=1,2,...\|M\|}$ where each v_i is a sequence of representation vectors for x_i
  Forward pass each v_i through Sequence Model parameterized by ξ to give predicted labels $\{\hat{y}_i\}_{i=1,2,...\|M\|}$
  Update parameters ξ using $\nabla_\xi L_{seq}(\xi, \{\hat{y}_i\}_{i=1,2,...\|M\|}, \{y_i\}_{i=1,2,...\|M\|})$
until convergence of ξ
return ϕ, θ and ξ

Algorithm 2 VAE-Seq Language Identification

Input: Speech clip x, Optimized VAE parameters (ϕ, θ), Optimized Sequence Model parameters (ξ)
Output: Language label y
Forward pass the segmented spectrogram of x through the VAE encoder having optimized parameters θ to obtain a sequence of representation vectors v
Forward pass v through the Sequence Model having optimized parameters ξ to obtain the language label y
return y

4 Experimental Results

We pre-train the VAE on segmented speech utterances from the CMU/IIITH Indic Speech Database (cmu) (Prahallad et al., 2012). The database contains raw speech utterances in 8 languages, namely Bengali, Gujarati, Hindi, Kannada, Marathi, Punjabi, Tamil and Telugu. The raw audio is converted to a Mel spectrogram (with 40 Mel filter banks and FFT window of size 1024 units). The Mel spectrogram is then segmented along the time axis, with an overlapping window of 4 units, producing a sequence of spectrograms, each of dimensions (40×20). The VAE is then trained to learn representations for these small segments (utterances) in an unsupervised manner.

Similar pre-processing is applied on each speech utterance, prior to training the sequence models, to create a sequence of spectrograms, each of dimensions (40×20), which are then passed through the pre-trained VAE encoder to produce a sequence of representation vectors, each of size 128 units.

The sequence models are trained on the above pre-processed speech data. We use cross-entropy between the output logits and the labels as the loss metric, which is minimized using Adam optimizer (Kingma and Ba, 2014), with learning rate of 10^{-4}, β_1 of 0.999, β_2 of 0.99 and weight decay

Table 1: Comparison of models

Model	Accuracy
GMM-HMM (3 languages) (Shikhamoni Nath)	86.1%
GMM + spec-pros feat (8 languages) (Vempada et al., 2013)	58.45%
DNN (8 languages) (Vuddagiri et al., 2018)	83.17%
DNN-WA (8 languages) (Vuddagiri et al., 2018)	86.10%
VAE + Bi-LSTM (1 layer) with Self-Attention	**88.24%**
VAE + Bi-LSTM (2 layers) with Self-Attention	**89.25%**
VAE + Bi-LSTM with Soft-Aligned Attention	**87.56%**
VAE + RCNN	**86.72%**
VAE + Transformer	**86.22%**

of 10^{-5}.

The results obtained on the testing set are shown in Table 1 compared with GMM-HMM based approach (Shikhamoni Nath), GMM along with spectral and prosodic features (Vempada et al., 2013), Deep Neural Network based approach (Vuddagiri et al., 2018) and DNN with Attention (Vuddagiri et al., 2018). In the table, the results are mapped to the 8 languages under consideration. The confusion matrices obtained for each sequence model are shown in Figure 2 (Appendix). We clearly see that deep learning based approaches outperform feature engineering and classical approaches. Our approach shows a performance gain in terms of accuracy compared to previous deep learning approaches as well.

5 Conclusion

In this paper, we have introduced a new framework for Indian language identification using VAE representation learning and state-of-the-art sequence models to capture the temporal characteristics of speech. The framework performs well on identification of 8 well known languages. The framework also helps improve language identification in the future as sequence models in natural language processing become better capturing long range dependencies and other temporal aspects of sequences. It can be applied in several other speech processing scenarios as well where the task requires representation learning from speech utterances and subsequent classification using a sequence model. We see VAEs are powerful probabilistic models which can learn useful representation from speech utterances and these representations can be utilized in several downstream tasks.

References

CMU Indic speech synthesis databases. http://festvox.org/cmu_indic/.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Aarti Bakshi and Sunil Kumar Kopparapu. 2017. Spoken indian language classification using artificial neural network — an experimental study. pages 424–430.

Christian Bartz, Tom Herold, Haojin Yang, and Christoph Meinel. 2017. Language identification using deep convolutional recurrent neural networks. *ArXiv*, abs/1708.04811.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, Austin, Texas. Association for Computational Linguistics.

Javier Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and Pedro Moreno. 2014. Automatic language identification using long short-term memory recurrent neural networks. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 2155–2159.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Wei-Ning Hsu, Yu Zhang, and James Glass. 2017. Learning latent representations for speech generation and transformation. In *Interspeech*, pages 1273–1277.

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

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2267–2273. AAAI Press.

Metso Leena, K. Srinivasa Rao, and Bayya Yegnanarayana. 2005. Neural network classifiers for language identification using phonotactic and prosodic features. *Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing, 2005.*, pages 404–408.

Yun Lei, Luciana Ferrer, Aaron Lawson, Mitchell McLaren, and Nicolas Scheffer. 2014. Application of convolutional neural networks to language identification in noisy conditions. In *Odyssey*.

Liang Wang, E. Ambikairajah, and E. H. C. Choi. 2006. Multi-lingual phoneme recognition and language identification using phonotactic information. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 245–248.

L. V. D. Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Tirusha Mandava and Anil Kumar Vuppala. 2019. Attention based residual-time delay neural network for indian language identification. *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pages 1–5.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

V. MounikaK., Sivanand Achanta, R. LakshmiH., Suryakanth V. Gangashetty, and Anil Kumar Vuppala. 2016. An investigation of deep neural network architectures for language recognition in indian languages. In *INTERSPEECH*.

K. Prahallad, Naresh Kumar Elluru, Venkatesh Keri, S. Rajendran, and A. Black. 2012. The iiit-h indic speech databases. In *INTERSPEECH*.

Sarthak, Shikhar Shukla, and Govind Mittal. 2019. Spoken language identification using convnets. In *European Conference on Ambient Intelligence*.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681.

Priyankoo Sarmah Samudravijaya K Shikhamoni Nath, Joyshree Chakraborty. Machine identification of spoken indian languages.

Ramakrishna Thirumuru, Ravikumar Vuddagiri, Krishna Gurugubelli, and Anil Kumar Vuppala. 2018. Significance of accuracy in vowel region detection for robust language identification. *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 826–830.

Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li, and Eng Chng. 2006. Integrating acoustic, prosodic and phonotactic features for spoken language identification. volume 1, pages I – I.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ramu Vempada, Sudhamay Maity, and K. Rao. 2013. Identification of indian languages using multi-level spectral and prosodic features. *International Journal of Speech Technology*, 16.

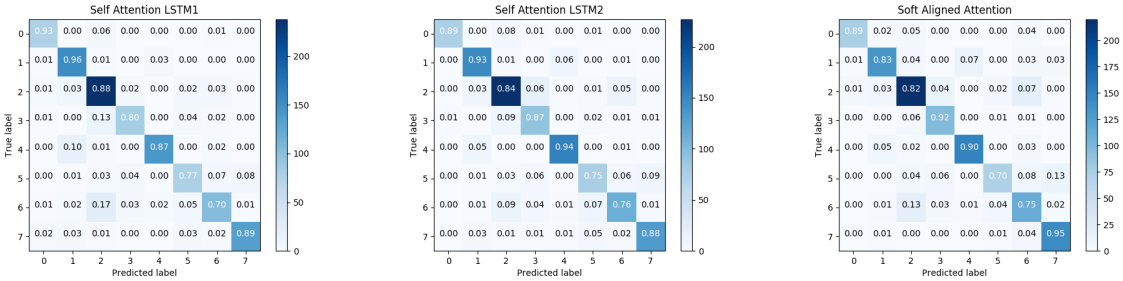
Ravi Kumar Vuddagiri, Krishna Gurugubelli, Priyam Jain, Hari Krishna Vydana, and Anil Kumar Vuppala. 2018. IIITH-ILSC Speech Database for Indain Language Identification . In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, pages 56–60.

A Appendices

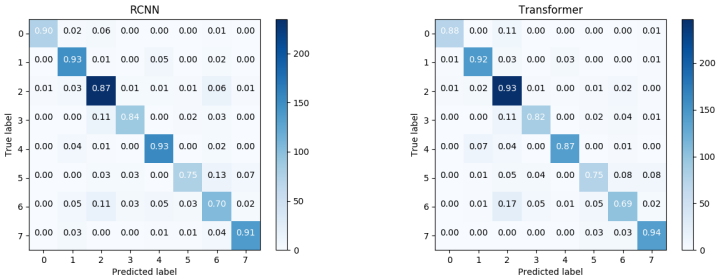
A.1 Qualitative Analysis

The T-SNE (Maaten and Hinton, 2008) embedding space of the representation in the last hidden layer assigned by each sequence model are shown in Figure 3. The visualizations give important deductions regarding the origins of each language considered.

In each T-SNE embedding space plot, we observe that Bengali and Hindi clusters appear close to each other, as Bengali and Hindi are indeed similar languages. Similar is the case with Marathi and Gujarati clusters, geographically being neighbouring states in India. The clusters of the South Indian languages of Tamil, Telugu and Kannada, geographically being neighbouring states, must appear near each other which prevails in most of the embedding space plots. The Punjabi cluster appears near the clusters of the South Indian languages of Tamil, Telugu and Kannada, an error which prevails in all the embedding space plots. There is clear distinction between the South Indian languages (believed to have Dravidian roots) and the North Indian languages (believed to have Indo-Aryan roots) in each embedding space, an important experimental finding.

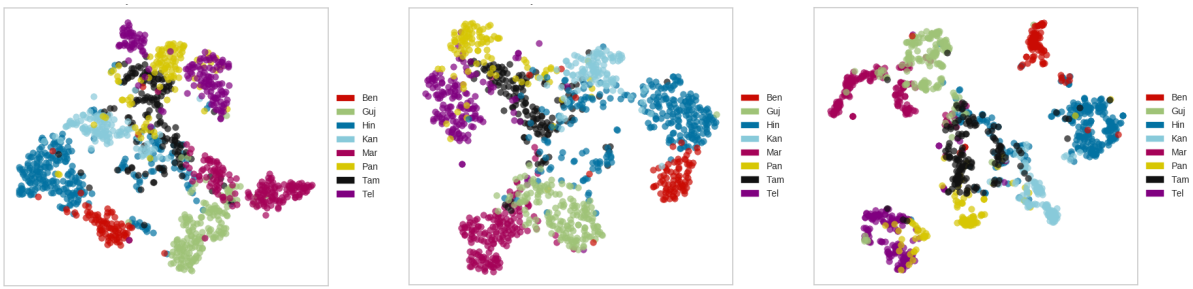


(a) Bi-LSTM with Self Attention (1 layer) (b) Bi-LSTM with Self Attention (2 layers) (c) Bi-LSTM with Soft Aligned Attention

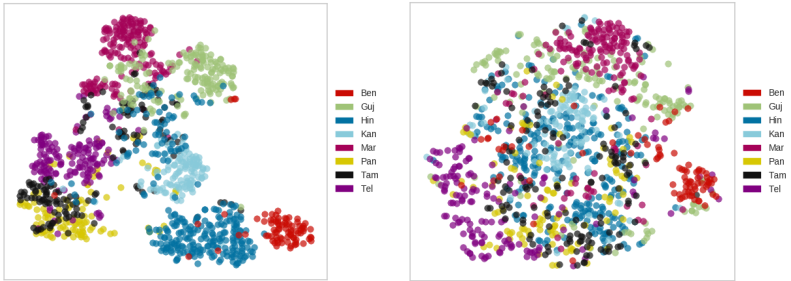


(d) RCNN (e) Transformer

Figure 2: Confusion matrices for each sequence model on test data. The corresponding labels are 0 for Bengali, 1 for Gujarati, 2 for Hindi, 3 for Kannada, 4 for Marathi, 5 for Punjabi, 6 for Tamil and 7 for Telugu.



(a) Bi-LSTM with Self Attention (1 layer) (b) Bi-LSTM with Self Attention (2 layers) (c) Bi-LSTM with Soft Aligned Attention



(d) RCNN (e) Transformer

Figure 3: T-SNE embedding space of representations in the last hidden layer assigned by each sequence model on test data.