Improving Automatic Speech Recognition with Dialect-Specific Language Models

Raj Gothi and Preeti Rao

Centre for Machine Intelligence and Data Science
Indian Institute of Technology Bombay
22m2160@iitb.ac.in
prao@ee.iitb.ac.in

Abstract. We present an end-to-end Automatic Speech Recognition (ASR) system in the context of the recent challenge tasks for Bhojpuri and Bengali. Our implementation follows the currently popular wav2vec models while we investigate ways to leverage the dialect-categorised data in order to improve ASR performance. We report overall improvements in word error rate with dialect-specific language models for each of the languages. We present an analysis that provides insights into some of the factors underlying the success of dialect-specific language models.

Keywords: Dialects · Language model · Bhojpuri ASR · Bengali ASR

1 Introduction

While ASR systems are typically built for a given language using language-specific transcribed speech and text resources, a particular challenge is the pervasive phenomenon of the language comprising of multiple dialects. These different forms of the same language, as used across its population of speakers, arise from variations ranging from geography to socio-cultural characteristics. Much prior research has addressed the robustness of ASR systems in the face of multiple dialects by dialect-specific acoustic modeling where the acoustic model (AM) is trained separately on transcribed speech of each dialect, and then used with either known or automatically labeled dialect speech [7]. Thus while dialects have been considered in speech recognition system development mainly for the accents or word pronunciation variability they introduce in spoken language, it may be noted that they can actually also encompass significant variations in grammar and vocabulary [9]. In ASR systems, the syntax and vocabulary constraints that influence the acoustic model predictions are learned via language modeling from the training data transcripts and possibly also from additional text resources of the language.

Recently, in the context of Telugu, a language with multiple dialects, Yadavalli et al. [15] showed that using a dialect-mismatched language model (LM) significantly degraded ASR performance even when a dialect-specific AM was employed. While they did not register an overall improvement in word-error rate (WER) with dialect-specific LMs, they demonstrated significant differences between matched- and mismatched-LM based ASR for each of three Telugu dialect datasets. By way of analysis, they fine-tuned a BERT-like system (INDIC-BERT [8]) for dialect identification from text to obtain sentence embeddings that exhibited clear clustering based on dialect. No further insights on the relevant dialectical factors was provided. They used the Conformer Model for their Acoustic Model (AM), but training it from scratch, i.e. without any pre-training. Additionally, they employ a separate Transformer LM for their Language Model.

In the present work, we reconsider the question of the role of dialect-specific language modeling for two other Indian languages, namely Bhojpuri (with three dialects) and Bengali (with five dialects). Our ASR system utilizes the wav2vec2 base pre-trained model with fine-tuning on our data for the Acoustic Model (AM), and an n-gram based model for the Language Model (LM). The data is obtained from the recent MADASR challenge [12]. We report improvements in WER on our dataset overall, apart from dialect-specific improvements, with matched LMs. We present an analysis of the datasets in terms of some of the factors that serve to explain the results.

2 Dataset

The dataset used in this research is MADASR [1] (Model ADaptation for ASR in low-resource Indian languages) [12]. MADASR is specifically designed to address the challenges posed by low-resource Indian languages, with a focus on Bengali and Bhojpuri. It provides valuable annotated speech data for these languages, allowing for the development and evaluation of accurate speech recognition systems.

Most Indian languages suffer from limited availability of high-quality data compared to high-resource languages, which hinders research and development efforts. This dataset contains a large amount of labeled speech data in Bengali and Bhojpuri, with 851 hours and 835 hours respectively. To account for the linguistic diversity within each language, the dataset was collected from various dialects representing distinct regions or communities. In Bengali, the dataset comprises speech data from five different dialects (D1, D2, D3, D4, and D5) with statistics as in table 1. Similarly, the Bhojpuri dataset includes data from three dialects (D1, D2, and D3) as in table 2. Every utterance in the dataset is also annotated with dialect id, allowing for easy categorization and targeted analysis. This dialect-based data allows the models to better capture and adapt to the specific linguistic characteristics present in each region. Furher, this dataset is drawn from diverse domains including Healthcare, Agriculture, Food, Technology, Sports, and others.

The text and transcriptions are in the native script for each of the languages with 72 characters in Bhojpuri and 64 in Bengali. In addition to the word-level transcription available for each audio utterance, the dataset includes a separate large text corpus for each dialect of both Bengali and Bhojpuri languages.

[1] MADASR webpage: https://sites.google.com/view/respinasrchallenge2023/home

Table 1. Training dataset statistics, including audio data hours, speaker count, dialects, total text sentences and unique words of **Bengali** Language, Dialects (D1, D2, D3, D4, D5: Dialect IDs).

Stats	D1	D2	D3	D4	D5	All Dialect
Total Hours	136	191	212	149	163	851
Speakers	379	385	402	402	412	1980
Text sentences	5K	6K	58K	5K	6K	80K
Unique words	9684	7174	47616	8356	10805	58349

Table 2. Training dataset statistics including audio data hours, speaker count, dialects, total text sentences, and unique words of **Bhojpuri** Language, Dialects (D1, D2, D3: Dialect IDs).

Stats	D1	D2	D3	All Dialect
Total Hours	263	316	256	835
Speakers	566	695	665	1926
Text sentences	34K	182K	26K	242K
Unique words	18923	65746	19878	77840

For Bengali, we have filtered out sentences from the text corpus that are not labeled with their corresponding dialect IDs. Consequently, the remaining sentences are annotated with their respective dialect IDs, making this dataset a valuable resource for conducting dialect-aware language modeling experiments. Table 1 and Table 2 list the dataset statistics of train data. We note that the audio data is more or less balanced across the dialects while the text resources are more skewed towards one of the dialects. All our ASR evaluations, of course, utilise audio data. All the results of this paper are based on the Dev dataset (audio + transcription) provided by the Challenge organisers. The ground truth for the Challenge test dataset has not been released. There are no shared speakers or sentences between the train and dev datasets. Furthermore, the Dev dataset is equally distributed across dialects for both the languages.

3 Methodology

In this section, we present the methodology employed in our research to develop an accurate speech recognition system for low-resource Indian languages, specifically Bengali and Bhojpuri. Our approach involves two main components: the Acoustic Model (AM) and the Language Model (LM). End-to-End speech recognition takes audio as input and predicts a character for each frame. By bringing in the Language Model with the E2E Acoustic Model, we rescore the logits predicted by the AM, resulting in meaningful text.

3.1 Acoustic Model

In recent years, self-supervised acoustic models have emerged as a breakthrough approach, showcasing improved results in various speech-related tasks, including speech recognition. The first component of our speech recognition system is the Acoustic Model (AM), which is based on the Wav2vec 2.0 base architecture [1]. It is a cutting-edge self-supervised speech representation learning model. Wav2vec 2.0 pretrained model comprises a CNN based feature encoder, a quantization module, and a transformer-based prediction network. The feature encoder layer processes the raw audio to obtain continuous speech representations, which are then quantized into discrete units by the quantization module. Finally, the transformer-based prediction network predicts the masked speech representation, following a contrastive learning objective.

To leverage the strengths of Wav2vec 2.0, we begin by initializing our AM with the weights of the pre-trained Wav2vec 2.0 base model. Next, we fine-tune the pre-trained Wav2vec 2.0 base model on our specific Bengali and Bhojpuri speech datasets using a transfer learning approach. This involves adding a fully connected layer on top of the transformer block, with the size of the output layer equal to the unique characters of the target languages. During the fine-tuning phase, we employ the Connectionist Temporal Classification (CTC) loss function [4]. CTC is widely used for end-to-end speech recognition tasks as it allows training without requiring explicit alignment between the input speech and the output transcriptions. This makes it suitable for sequence-to-sequence mapping tasks like automatic speech recognition. The initial component of Wav2Vec2 comprises a series of CNN layers responsible for extracting acoustically meaningful yet contextually independent features from the raw speech signal.This layer of the model has already undergone adequate training during the pretraining phase. In line with recommendations from the Wav2vec 2.0 paper [1], we choose to freeze the weights of the feature encoder network during training. All the transformer layers of pre-trained model's weights are updated during fine-tuning on the dataset. By fine-tuning the model on our labeled datasets, it learns to map the speech representations to their corresponding transcriptions.

3.2 Language Model

The Language Model (LM) is an essential component of our speech recognition system that aids in converting the speech representations generated by the Acoustic Model (AM) into meaningful text transcriptions. In our research, we explore and compare two different types of trained language models to enhance the accuracy of speech recognition.

The first language model, denoted as LM-All, is trained on the entire corpus of each language, encompassing all available dialects. This approach aims to create a language model that captures the general linguistic patterns and vocabulary present in Bengali and Bhojpuri without considering the variations due to specific dialects. By training on a diverse dataset, LM-All gains a broad understanding of the overall language characteristics. With our intention to study dialect-specific adaptations, we train separate language models, one for each dialect present in the dataset. We refer to these models as LM-Dialect. Each LM-Dialect is trained using a dialect-specific text corpus, allowing it to capture the unique linguistic variations and nuances associated with that particular dialect. This fine-grained approach can potentially facilitate better recognition of speech from diverse regions and speakers, making it particularly valuable for low-resource languages with multiple dialects like Bengali and Bhojpuri.

For language modeling, we utilize a statistical approach based on KenLM [5]. We conduct experiments with different n-gram LM models to identify the most suitable configuration for our specific datasets. During the speech recognition process, we identify the spoken dialect from the utterance's dialect ID and select the corresponding dialect-based language model (LM-Dialect). The selected LM-Dialect is used to decode the intermediate representations obtained from the AM and convert them into the corresponding text output.

4 Experiments Setup

Wav2vec2 Model Details We used pre-trained Wav2vec2 [1] base architecture with 90M parameters. The model has 3 modules, feature encoder, transformer block and linear projection. Where feature encoder contains 7 CNN layers each with 512 channels, kernel widths of (10,3,3,3,3,2,2) and strides of (5,2,2,2,2,2,2). There are 12 transformer layers with dimension 768, feed forward network dimension 3072 and 8 attention heads. The convolutional layer used for modeling relative positional embeddings has a kernel size of 128 and 16 groups. The linear projection layer has output dimension given by the size of the set of all characters in the target language and blank symbols.

Fine-Tuning For our experiments, we fine-tuned a pre-trained wav2vec 2.0 base model checkpoint using the Hugging Face Transformers library [14]. For optimization, we employed the AdamW optimizer [10]. During the fine-tuning process, we set the warmup step to 400. Within these steps, the learning rate increased linearly from 0 to 1e-4, and after reaching 400 steps, it decreased linearly. To control overfitting, we set the weight decay to 0.005 and dropout rate [13] to 0.1. The model was trained for 3 epochs on each languages. The training was conducted on hardware with an RTX 3060 and RTX 2080 Ti GPUs.

Language Model In speech recognition, training a model with the CTC loss function enables it to handle variable-length sequences without requiring explicit alignments between input and output. During inference, the model outputs probabilities over characters. The naive greedy decoding approach involves selecting the character with the highest probability at each step, which is the procedure used in the case of acoustic model (AM) alone without a language model (LM). To enhance the model's performance, an n-gram language model can be integrated into the decoding process, replacing the naive greedy decoding with n-gram-boosted beam search decoding [3]. By using pyctcdecode library [2]for the beam search algorithm and leveraging linguistic information from the language model, the system can explore a range of likely word sequences, leading to more accurate and contextually coherent transcriptions in speech recognition tasks.

In our experiments, we utilized KenLM [5] to create a language model trained on the provided speech transcriptions and an additional text corpus provided in the Challenge dataset. During the experiments, we set the beam width to 100. We tested various n-gram configurations, including 3, 4, 5, and 6. We categorized the text corpus data into dialect-specific text using the available dialect IDs.

5 Results

In this section, we present the comparison across the different investigated systems based on the Word Error Rate (WER) metric, where a lower WER indicates better performance. The systems under consideration include:

AM: This represents a decoder with Acoustic model but without the application of any language model.

AM + n-gram LM-All: In this case, the respective n-gram language model (LM) is applied to the AM model outputs. The LM-All is trained on the entire corpus without considering dialects and applied to the recognition of utterances from the corresponding dialect.

AM + n-gram LM-Dialect: Here, the AM model is combined with the n-gram language model specific to each dialect ID. The LM-dialect is trained on separate text data for individual dialects. Finally, the overall performance across the dialect datasets is reported.

Table 3. Comparison of different systems for each of **Bengali** and **Bhojpuri** Languages in terms of Percentage Word Error Rate(WER).

Model	Bengali	Bhojpuri
AM	21.8	21.21
AM + 3 gram LM-All	16.42	17.10
AM + 3 gram LM-Dialect	15.90	16.95
AM + 4 gram LM-All	16.12	16.87
AM + 4 gram LM-Dialect	15.77	16.43
AM + 5 gram LM-All	16.04	16.76
AM + 5 gram LM-Dialect	**15.62**	16.48
AM + 6 gram LM-All	16.06	16.67
AM + 6 gram LM-Dialect	15.68	**16.26**

Table 3 shows the WER results obtained from each of these models for Bengali and Bhojpuri languages across different orders of the n-gram. Incorporating

[2] https://github.com/kensho-technologies/pyctcdecode/

language modeling significantly improves the performance of speech recognition for both Bengali and Bhojpuri languages, as expected. The dialect-based language models (LM-Dialect) consistently outperform the whole corpus language models (LM-All) in terms of WER for both Bengali and Bhojpuri. This observation underscores the importance of considering dialect-specific linguistic variations when training language models, at least in Indian language scenarios. The choice of n-gram order has an impact on the performance of language models. We observed that the 5-gram LM-Dialect has achieved better performance in the Bengali language, while the 6-gram LM-Dialect has outperformed in the Bhojpuri language. Although WER improved, we noted that the OOV error rate went up with the shift to dialect-specific LM due to the mismatches between train and dev data that were otherwise compensated for by the much larger all-dialect train and dev datasets.

Table 4. Model Comparison for **Bhojpuri** Dialects in Word Error Rate (WER), Where D1, D2, D3 are dialect IDs.

Dialect model	AM	AM + 6 gram LM-All	AM + 6 gram LM-Dialect
D1	20.56	14.90	**14.70**
D2	20.89	16.00	**15.80**
D3	21.97	18.59	**17.88**
All	21.21	16.67	**16.26**

Table 5. Model Comparison for **Bengali** Dialects in Word Error Rate (WER), Where D1, D2, D3, D4, D5 are dialect IDs.

Dialect model	AM	AM + 5 gram LM-All	AM + 5 gram LM-Dialect
D1	19.76	15.02	**14.02**
D2	21.44	16.08	**15.60**
D3	18.40	14.44	**13.93**
D4	20.71	16.21	**15.73**
D5	27.90	18.19	**18.15**
All	21.21	16.04	**15.62**

Table 4 and 5 present the performance of each model on individual dialects. The results demonstrate that each dialect benefits from the application of Dialect-based Language Model (LM-Dialect) as compared to Whole Corpus Language Model (LM-All) in both Bhojpuri and Bengali languages. The dialect-based language models effectively capture linguistic variations and regional nuances, leading to more accurate transcriptions. These findings underscore the benefit of leveraging dialect-based language models to achieve superior speech recognition results for low-resource languages with multiple dialects.

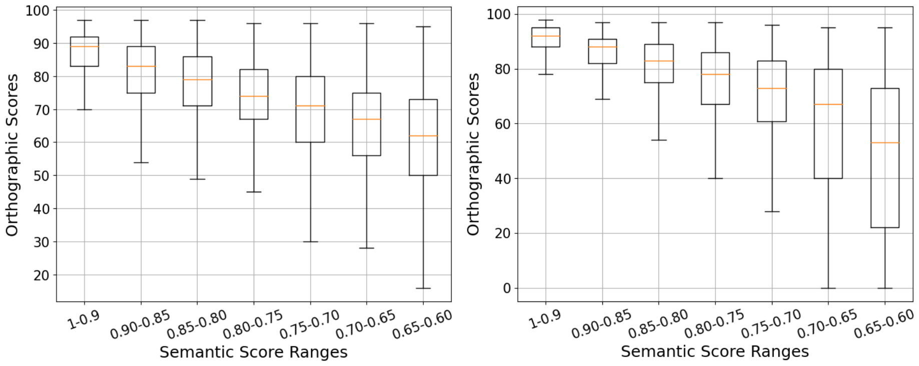
6 Analysis and Discussion

The variations in performance from language models trained on different dialects, with a fixed acoustic model, point to the influence of the underlying differences in vocabulary and grammar across dialects (rather than accent). We note that there are differences in vocabulary size across our dialect texts but the gains in using dialect-matched LMs leads us to speculate that there are differences also in the distributions of the words that are common across the dialects. We consider words which have a particularly skewed distribution across the dialects of the language, suggesting that the words are more or less unique to one dialect. We expect then that semantically similar but not identical words appear in the other dialects. Semantic similarity can be measured via the similarity of the neighbourhood context of a word. We use fastText [2] to obtain a semantic score in [0,1] where 1 indicates matching semantics. As opposed to word2vec, fastText considers subwords (character n-grams) and is therefore capable of providing better embeddings for OOV words. We use the fastText Hindi model for Bhojpuri, and the Bengali model for Bengali.

For our analysis, we select pairs of such semantically similar words after ensuring that at least one of the words in the pair qualifies as a dialect-unique word in terms of at least 80% of the total occurrences lying within a single dialect dataset. For each pair, we compute the orthographic distance between the two words as the edit distance between the grapheme (Unicode) strings [11,6]. The orthographic score is mapped to 0-100 % with 100% indicating exact match.

Fig. 1a and Fig. 1b show the distribution of the orthographic score for different extents of semantic similarity. We observe that the orthographic score is high (i.e. edit distance is low) for semantically close words. This is an experimental validation of we might expect for dialects of the same language as opposed to the case of distinct languages. Given the high incidence of distinct words across dialects that are semantically and orthographically similar, we expect strong AM-based confusions that can be more effectively resolved with dialect-specific LMs rather than a universal LM.

Fig. 2 provides examples of word pairs representing semantically similar words that are orthographically distinct, but very close in orthography (and therefore pronunciation) in the dataset. These closely related words are more likely to confuse the Acoustic Model (AM) during the speech recognition process. Blue-colored words have a higher total count in the entire dataset compared to orange-colored words. As a result, the Whole Corpus Language Model (LM-All) tends to assign higher probabilities to the blue-colored words during the speech recognition process. However, the situation changes when considering specific dialects. In certain dialects, orange-colored words may be more frequent than blue-colored words, contrary to the overall corpus trend. In such cases, the Dialect-based Language Model (LM-Dialect) becomes advantageous



(a) Bhojpuri language

(b) Bengali language

Fig. 1. Box plots of orthographic scores by semantic score ranges for Bengali and Bhojpuri languages. Each plot displays the distribution of orthographic scores for word pairs in different semantic score ranges (1-0.9, 0.90-0.85, 0.85-0.80, 0.80-0.75, 0.75-0.70, 0.70-0.65, 0.65-0.60). The x-axis represents the semantic score ranges, and the y-axis represents the orthographic scores. These plots offer insights into the model's behavior concerning semantic and orthographic similarity in diverse linguistic contexts.

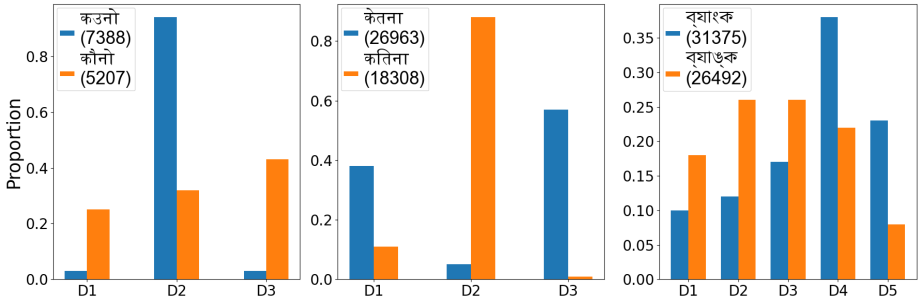


Fig. 2. The distribution across dialects of the two words of a pair where the words are semantically similar and, as a consequence, orthographically close. Bhojpuri (Left and Middle plots) and Bengali(Right plot). Legends within figure shows the words with their counts in the overall dataset.

as it is tailored to the linguistic characteristics of the specific dialect, capturing the prevalence and nuances of orange-colored words. Consequently, LM-Dialect exhibits better decoding performance than LM-All for these dialects, as it can more accurately reflect the dialect-specific language patterns.

7 Conclusions

Our research targeted a particular aspect of speech recognition in low-resource Indian languages, Bengali and Bhojpuri. We demonstrated the importance of dialect-based language models (LM-Dialect) in capturing unique linguistic variations related to vocabulary. LM-Dialect outperformed the LM applied on the whole langauge, offering contextually relevant speech recognition. We showed that a particular characteristic of dialects of the same language is the presence of distinct but semantically similar words that are also very close in orthography. This explains, at least partly, the benefits observed with dialect-specific language modeling from dialect specific text resources.

Our findings contribute to improving technology accessibility for diverse linguistic communities, fostering inclusivity and promoting linguistic diversity. Future research in this area can further enhance speech recognition systems for a wide range of languages and dialects.

References

1. Baevski, A., Zhou, H., rahman Mohamed, A., Auli, M.: wav2vec 2.0: A framework for self-supervised learning of speech representations. ArXiv **abs/2006.11477** (2020), https://api.semanticscholar.org/CorpusID:219966759
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
3. Collobert, R., Puhrsch, C., Synnaeve, G.: Wav2letter: an end-to-end convnet-based speech recognition system. arXiv preprint arXiv:1609.03193 (2016)
4. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. p. 369–376. ICML '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1143844.1143891, https://doi.org/10.1145/1143844.1143891
5. Heafield, K.: KenLM: Faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. pp. 187–197. Association for Computational Linguistics, Edinburgh, Scotland (Jul 2011), https://aclanthology.org/W11-2123
6. Inc, S.: fuzzywuzzy: Fuzzy String Matching in Python (2014), https://github.com/seatgeek/fuzzywuzzy
7. Jain, A., Upreti, M., Jyothi, P.: Improved accented speech recognition using accent embeddings and multi-task learning. In: Interspeech. pp. 2454–2458 (09 2018). https://doi.org/10.21437/Interspeech.2018-1864

8. Kakwani, D., Kunchukuttan, A., Golla, S., Gokul, N., Bhattacharyya, A., Khapra, M.M., Kumar, P.: IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 4948–4961 (2020)
9. Lei, Y., Hansen, J.H.L.: Dialect classification via text-independent training and testing for arabic, spanish, and chinese. IEEE Transactions on Audio, Speech, and Language Processing **19**(1), 85–96 (2011). https://doi.org/10.1109/TASL.2010.2045184
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bkg6RiCqY7
11. Mouselimis, L.: fuzzywuzzyR: Fuzzy String Matching (2021), r package version 1.0.5, https://CRAN.R-project.org/package=fuzzywuzzyR
12. Singh, A., Mehta, A.S., S, A.K.K., G, D., Date, G., Nanavati, J., Bandekar, J., Basumatary, K., P, K., Badiger, S., Udupa, S., Kumar, S., Savitha, Ghosh, P.K., V, P., Pai, P., Nanavati, R., Saxena, R., Mora, S.P.R., Raghavan, S.: Model adaptation for asr in low-resource indian languages (2023)
13. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(56), 1929–1958 (2014), http://jmlr.org/papers/v15/srivastava14a.html
14. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), https://www.aclweb.org/anthology/2020.emnlp-demos.6
15. Yadavalli, A., Mirishkar, G.S., Vuppala, A.: Exploring the effect of dialect mismatched language models in Telugu automatic speech recognition. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop. pp. 292–301. Association for Computational Linguistics, Hybrid: Seattle, Washington + Online (Jul 2022). https://doi.org/10.18653/v1/2022.naacl-srw.36, https://aclanthology.org/2022.naacl-srw.36