Improving performance of Transformer based Low Resource Speech Recognition for Indian Languages

Vishwas M. Shetty, Metilda Sagaya Mary N J, S. Umesh

Indian Institute of Technology Madras, India

Objective

Address the Speech Recognition problem of Low Resource Indian Language using Transformers

Explore different ways of incorporating language information

- At character level.
- At acoustic feature level.

Our method of providing language information at feature level gave the best recognition performance.

Transformers in Speech Recognition

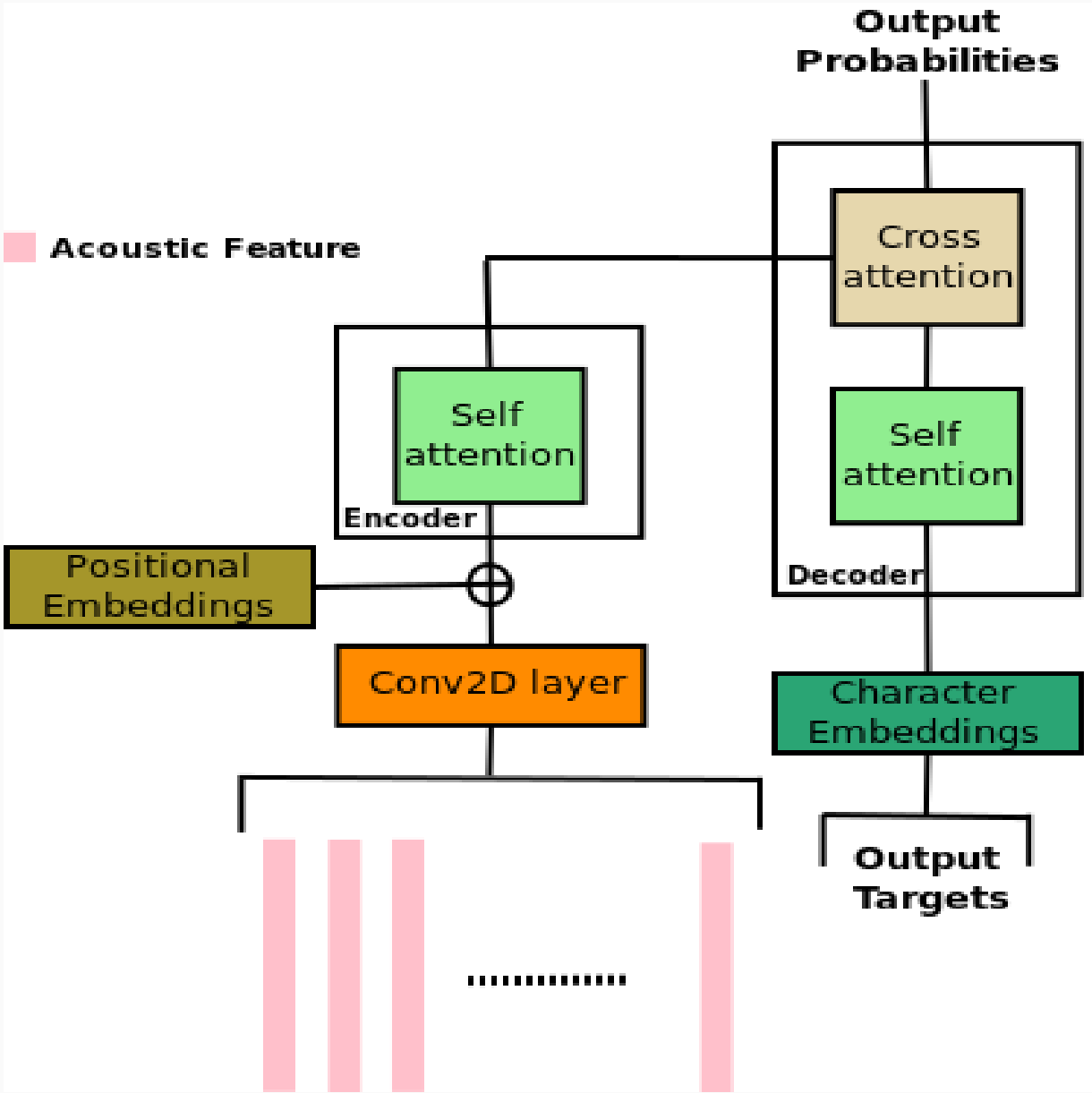
Transformers in ASR

Transformer

- An E2E framework based solely on attention.
- It comprises of a series of ”Attention” (Self/Cross) and ”Feed forward networks”
- Have shown promising results in several NLP tasks.

This has motivated its application to Automatic Speech Recognition.

Transformers in ASR



Dataset Details

Dataset Details

Language		Train	Dev	Eval
Gujarati	Dur(hrs)	40	5	5
	# Utter	22807	3075	3419
Tamil	Dur(hrs)	40	5	5
	# Utter	39131	3081	2609
Telugu	Dur(hrs)	40	5	5
	# Utter	44882	3040	2549

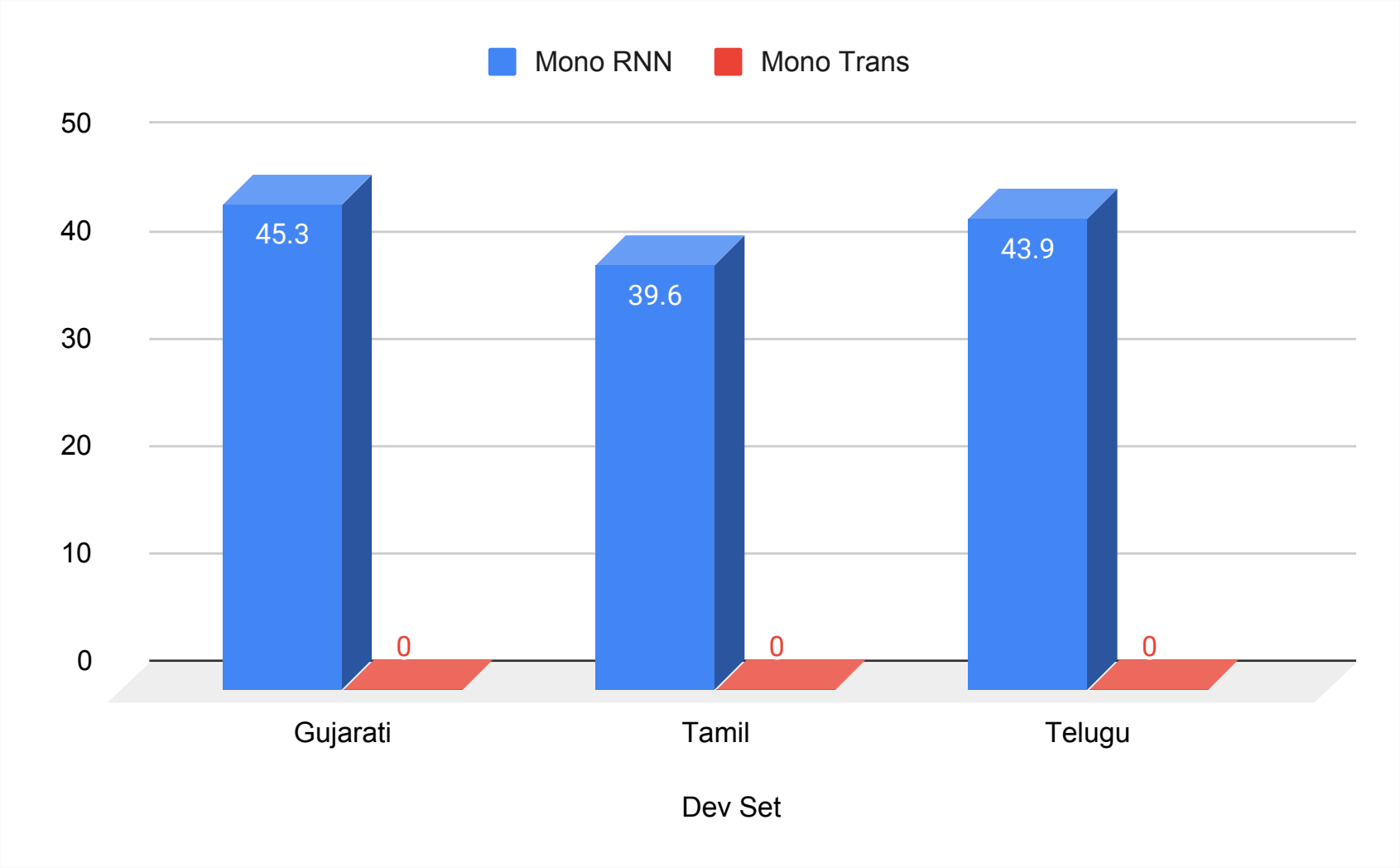
Transformers versus RNNs

Transformers versus RNNs

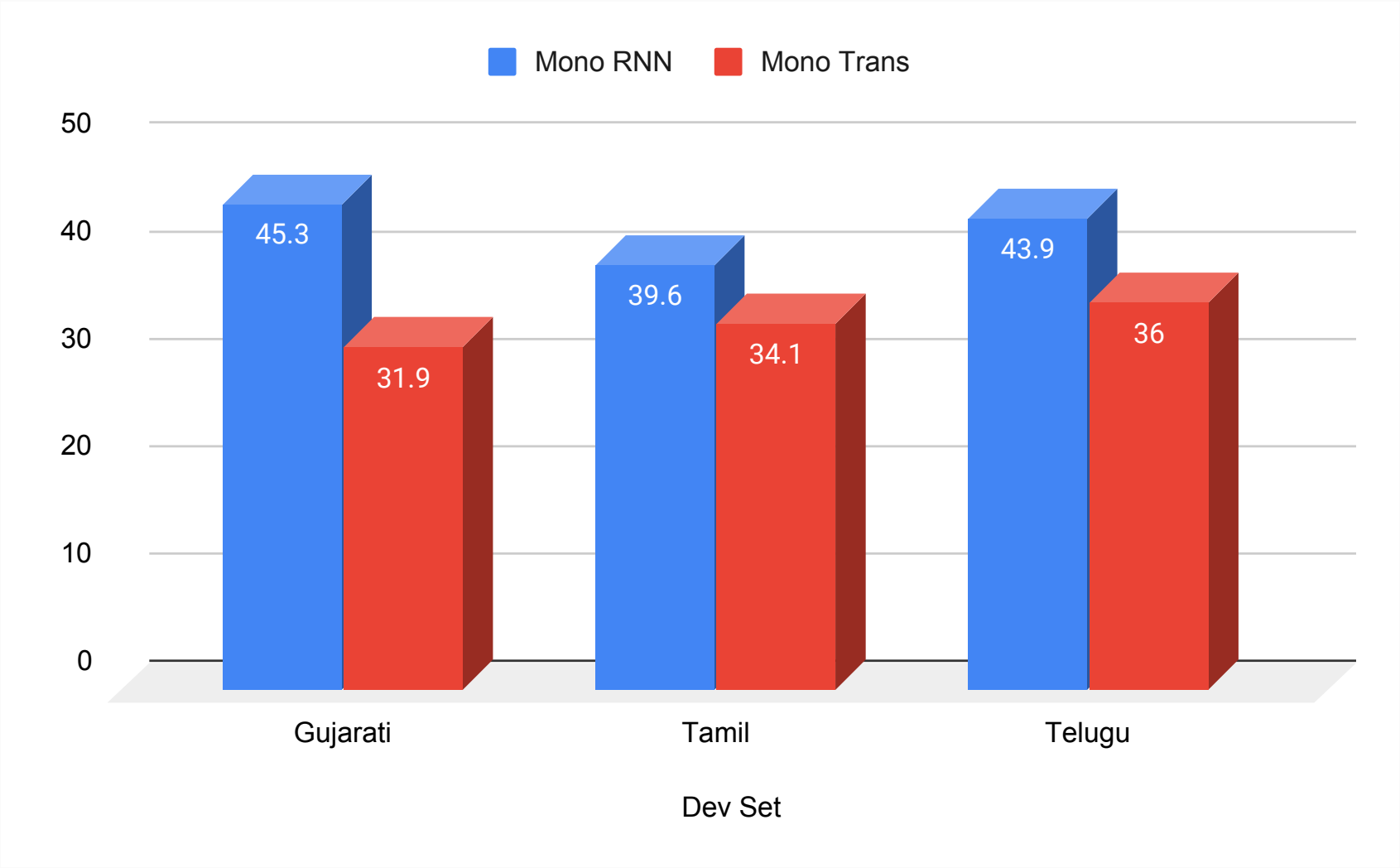
Models were trained based on Hybrid CTC/Attention approach using ESPNET tool on GTX 1080 GPU.

- RNN based model:
  - Four layer encoder with 320 BLSTM units
  - One layer decoder with 300 LSTM units
  - ”Location” aware attention was used
  - Multi-task learning co-efficient - 0.5
- Transformer model
  - Twelve layer encoder with 2048 units
  - One layer decoder with 1024 units
  - Attention dimension of 256
  - Multi head attention with four attention heads
  - Multi-task learning co-efficient - 0.3

Transformers versus RNNs

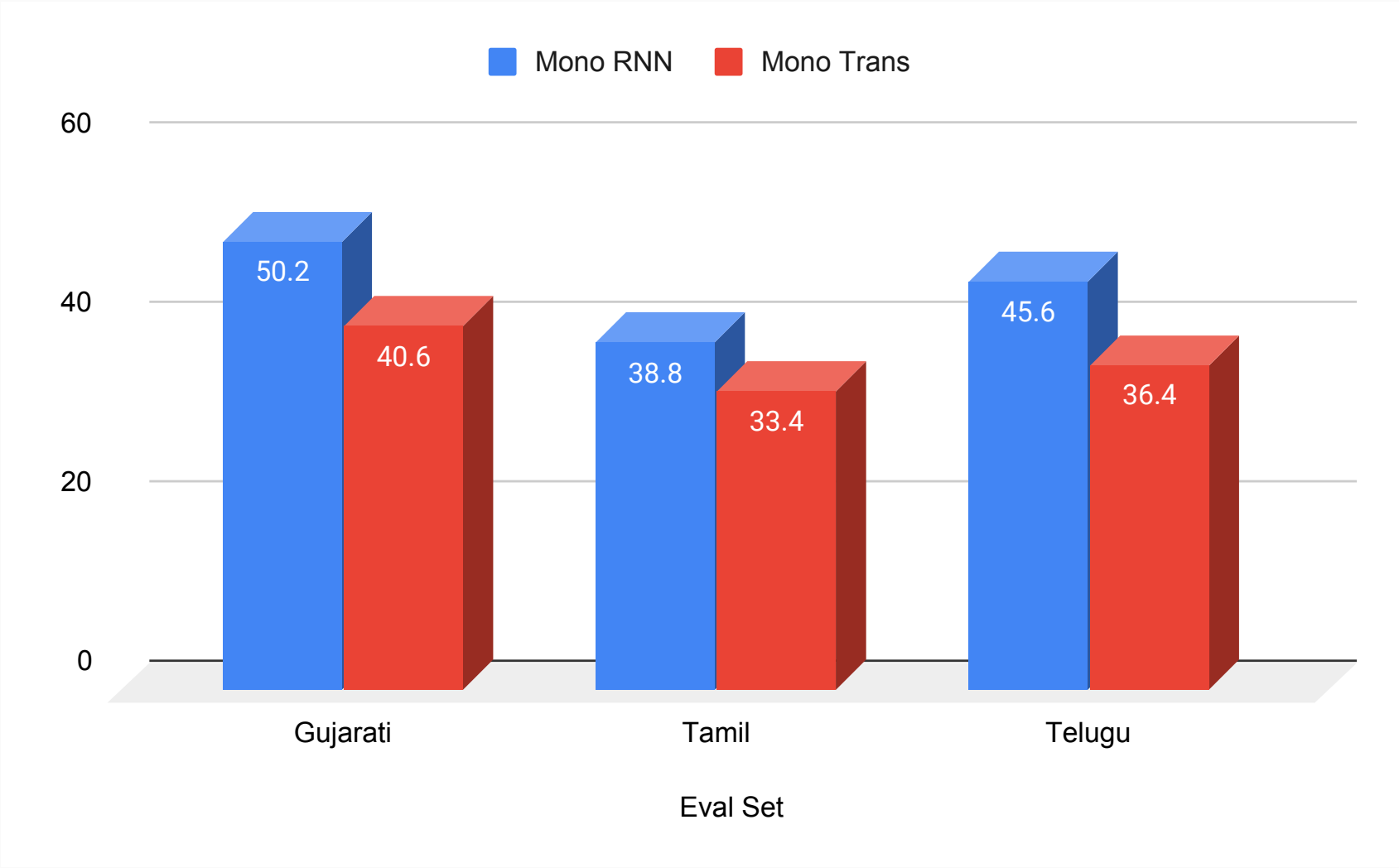


Transformers versus RNNs



Transformers versus RNNs

Transformers versus RNNs



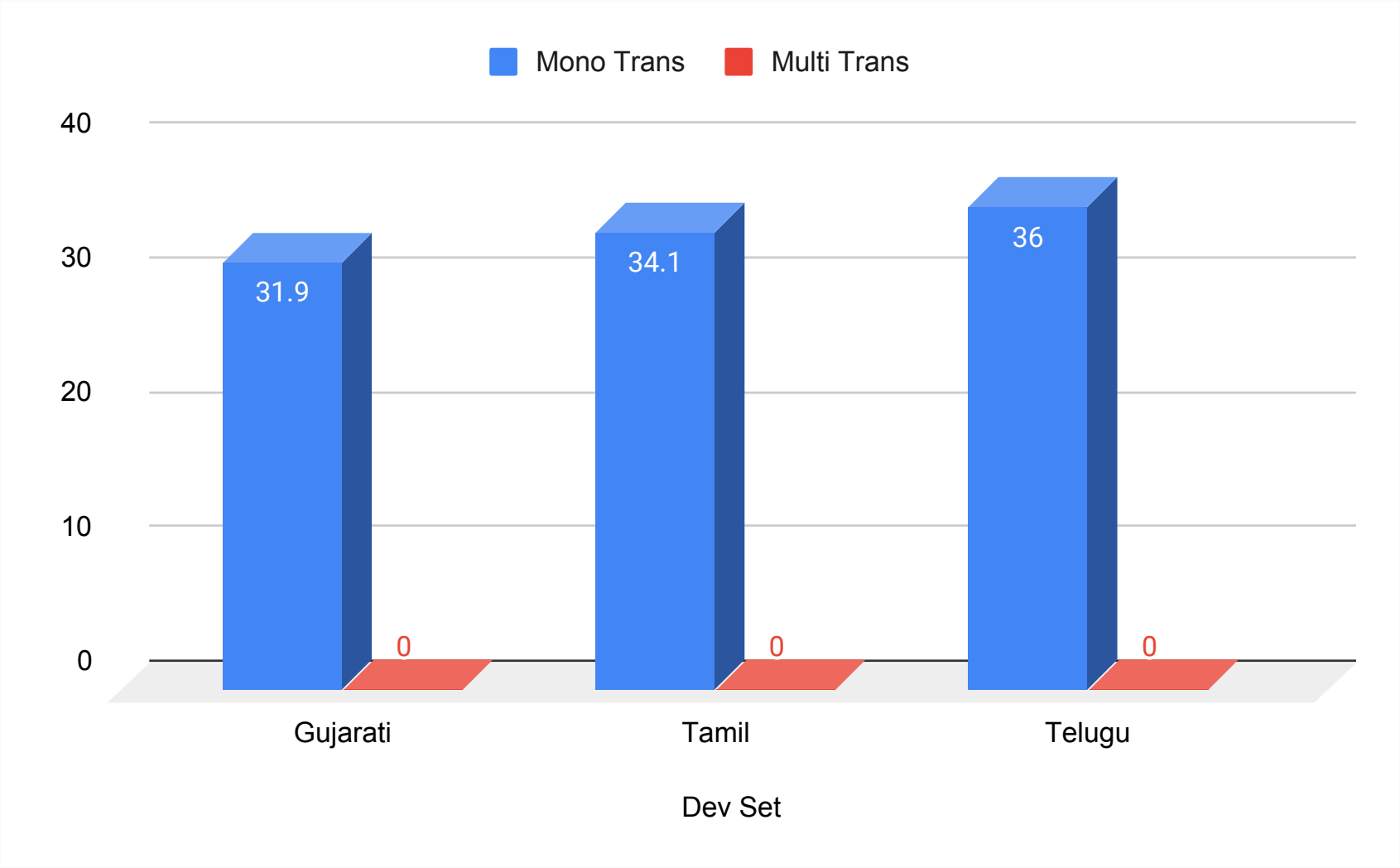
Can we do better than this?

Multilingual Models - Pool data from all the languages

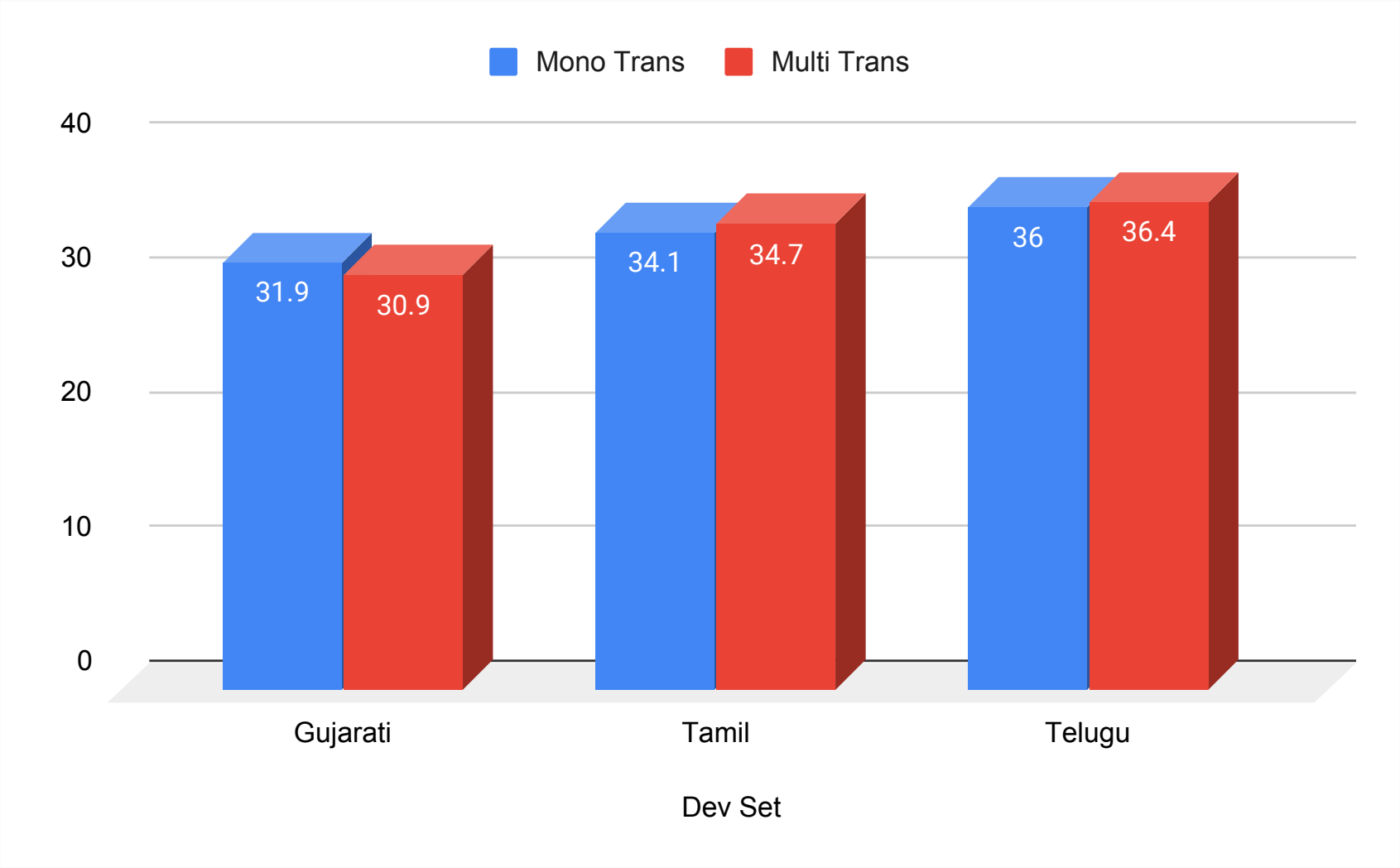
- Build one single model by combining data from all the languages.
- The target character set is the union of characters from individual languages - 64 Guj, 48 Tam and, 64 Tel

Language	Transcript
Gujarati	મેટ્રો રેલનું કામ
Tamil	பிரதமர் மன்மோகன் சிங்
Telugu	ఆంధ్రప్రదేశ్ రాజధాని అమరావతికి

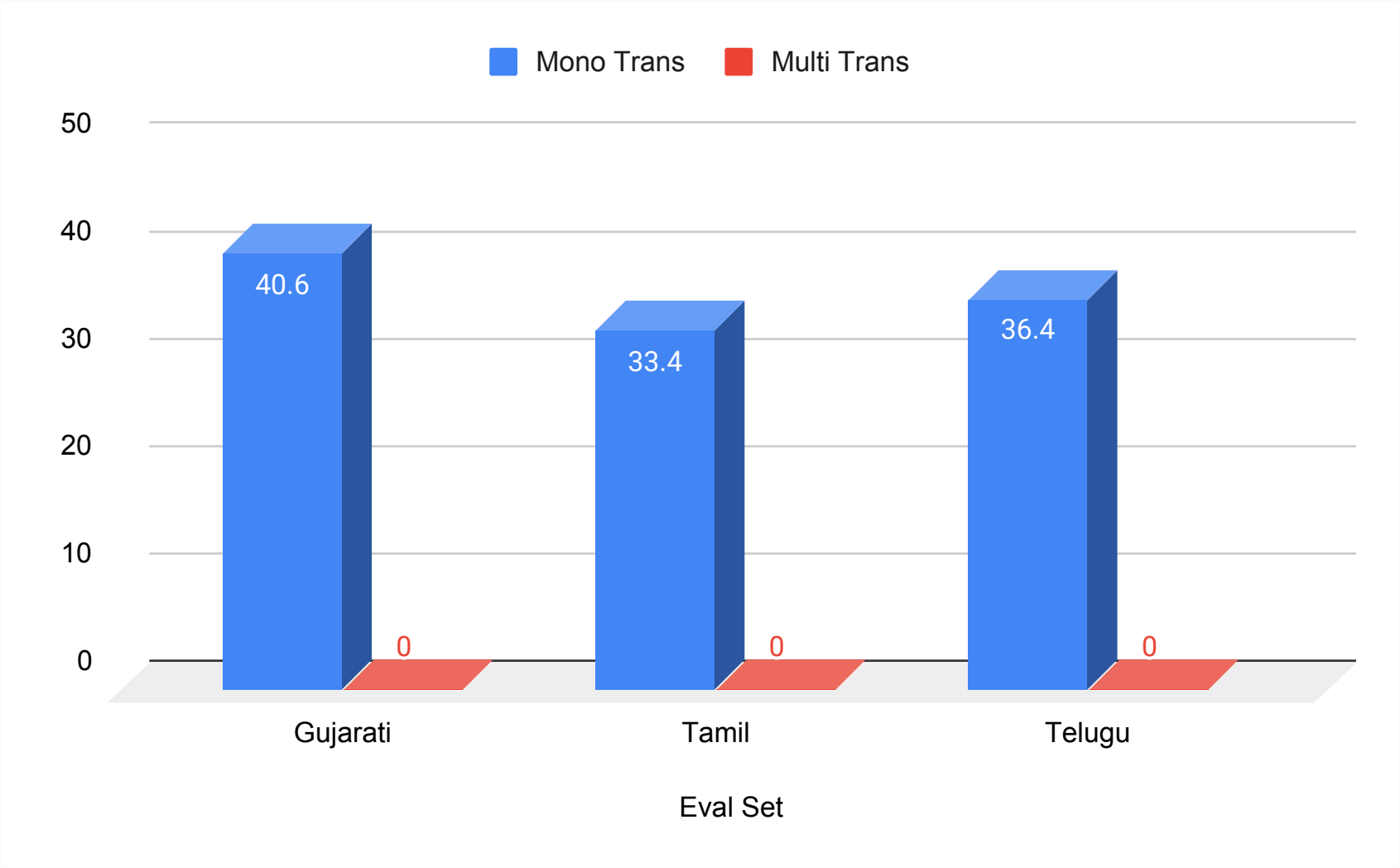
Multilingual Model – Pooled data



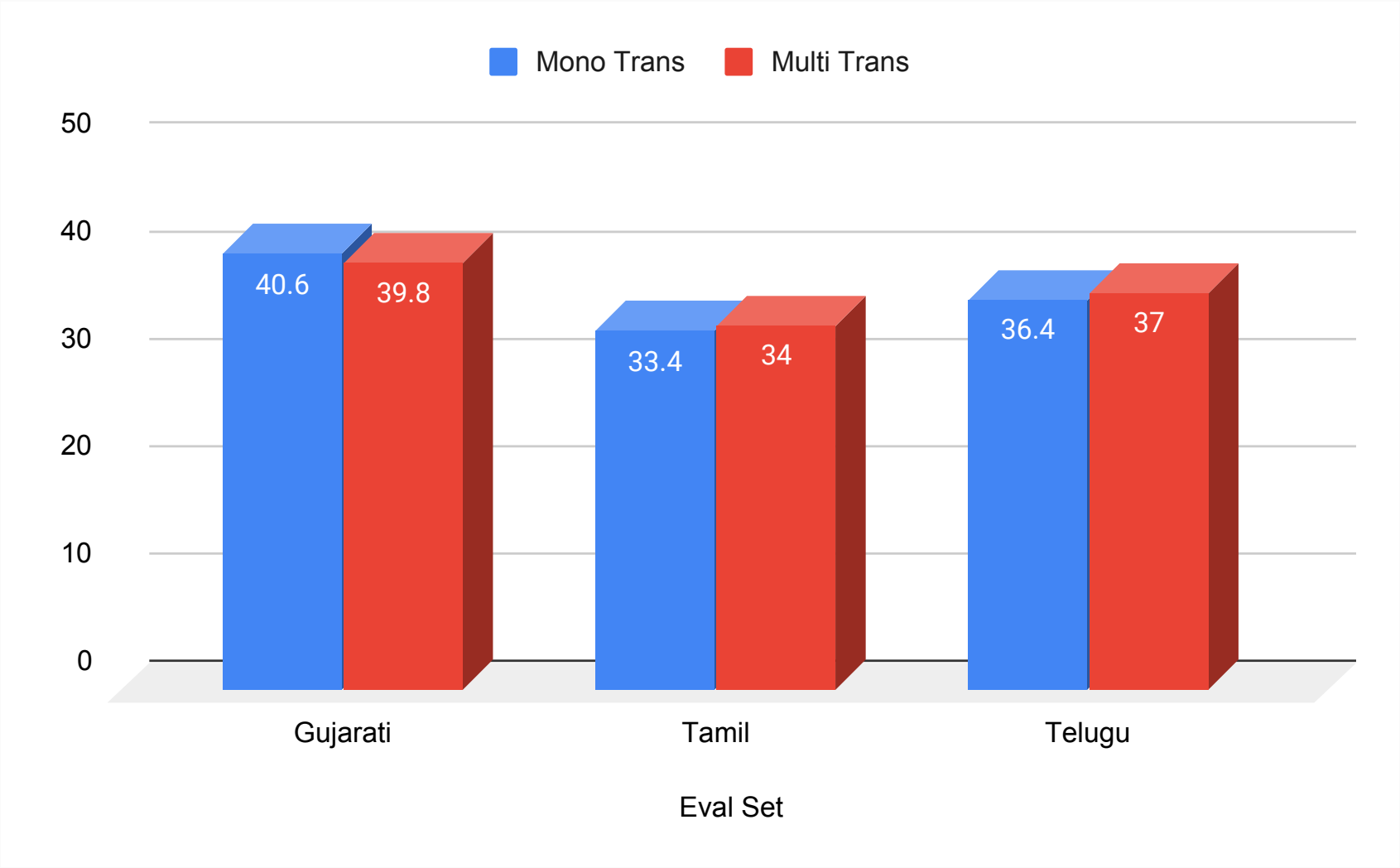
Multilingual Model – Pooled data



Multilingual Model – Pooled data



Multilingual Model – Pooled data



Can we make use of the language information?

Two possible Strategies

Assuming Language information known during training:

- Provide language information at the decoder –character level
- Provide language information at the encoder –feature level

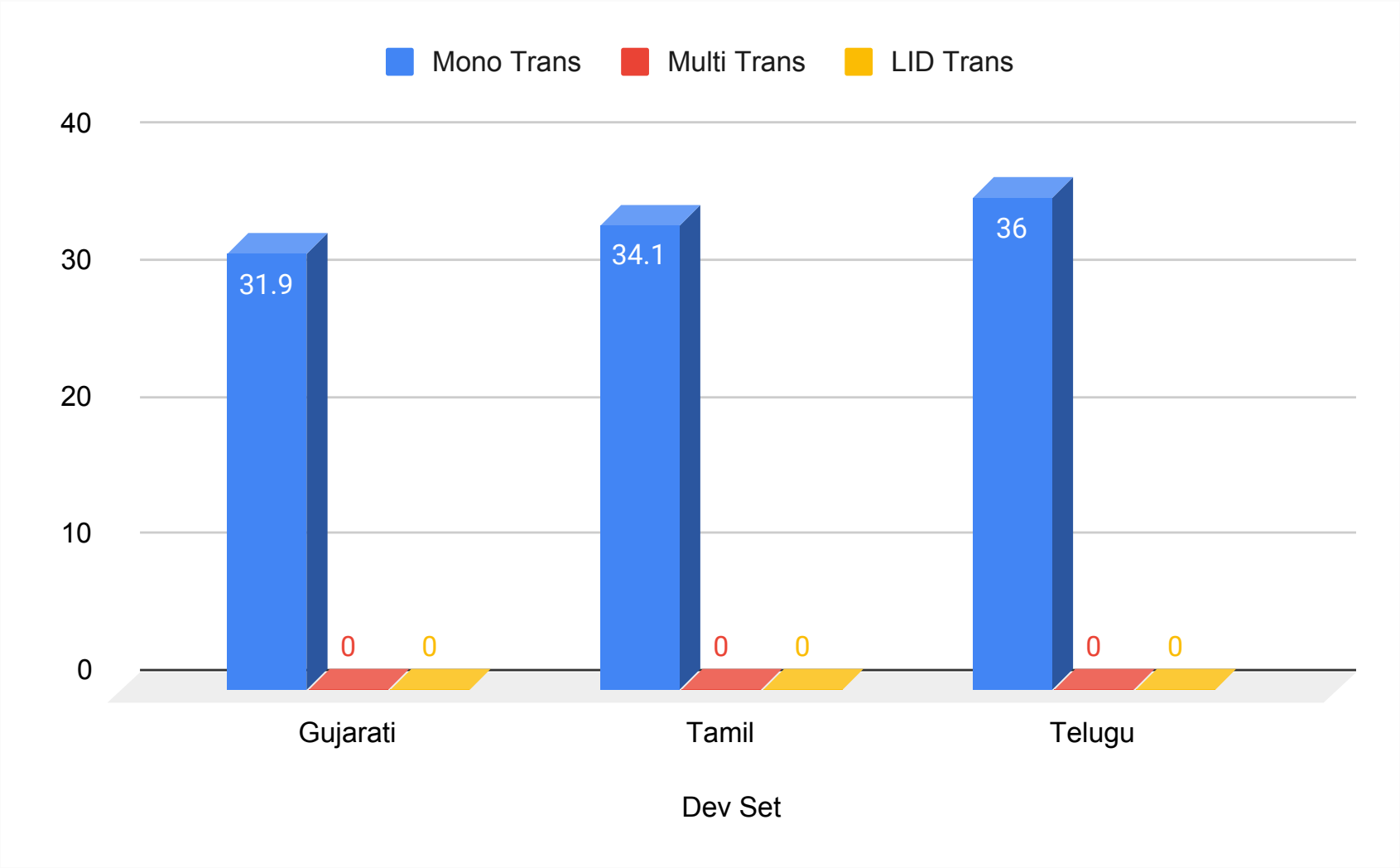
Strategy 1: Provide Language information at the decoder

Language	Example
Gujarati	<guj_beg> આતંકી હુમાલો હુમલો <guj_end>
Tamil	<tam_beg> கண்டிப்பா கண்டிப்பா <tam_end>
Telugu	<tlg_beg> అనంతరం <tlg_end>

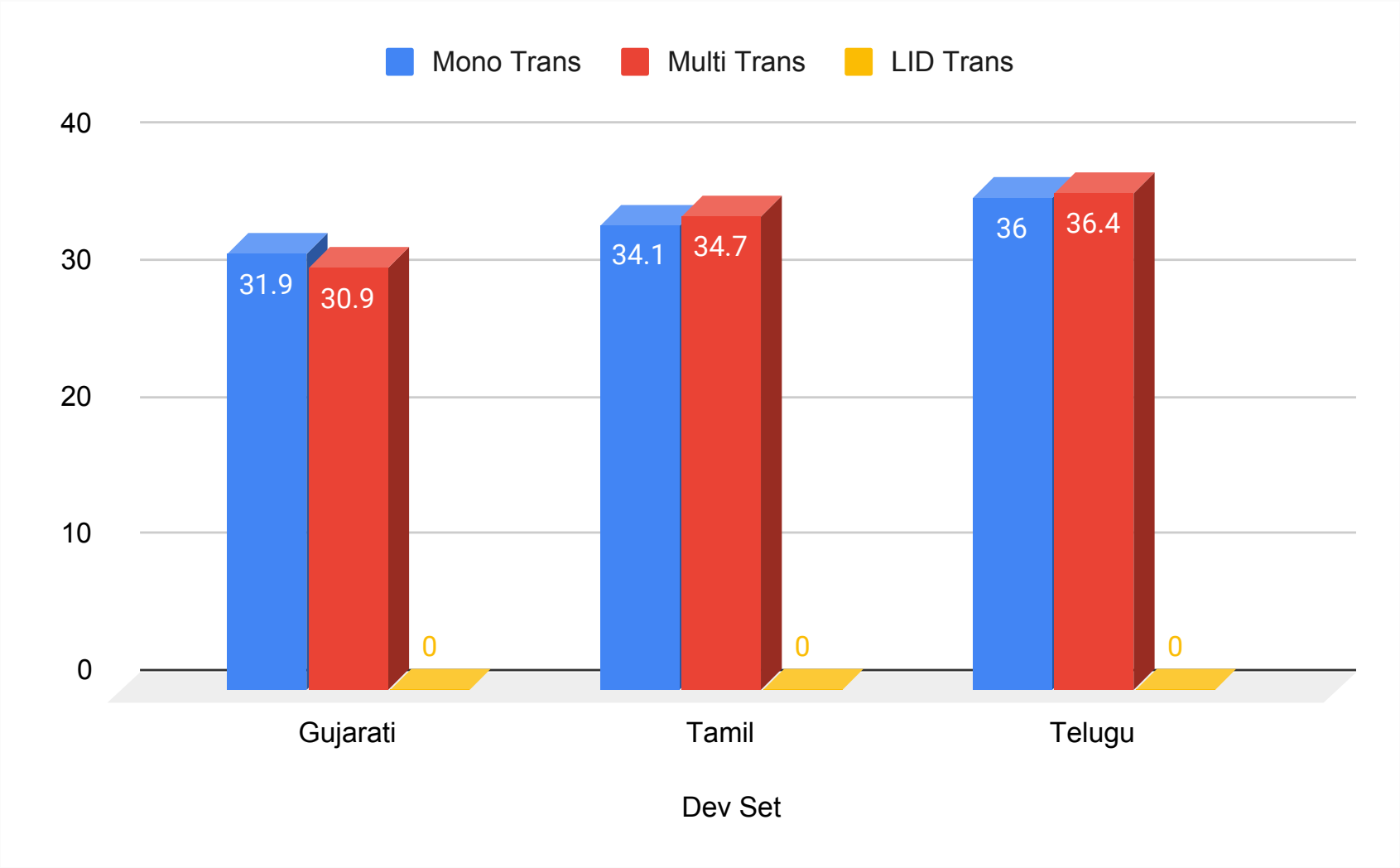
Table 1: Examples of LID in the target sequence

No LID used during testing – Hence Universal Speech recognizer.

Strategy 1: Provide Language information at the decoder



Strategy 1: Provide Language information at the decoder



Strategy 1: Provide Language information at the decoder

Strategy 1: Provide Language information at the decoder

Strategy 1: Provide Language information at the decoder

Strategy 1: Provide Language information at the decoder

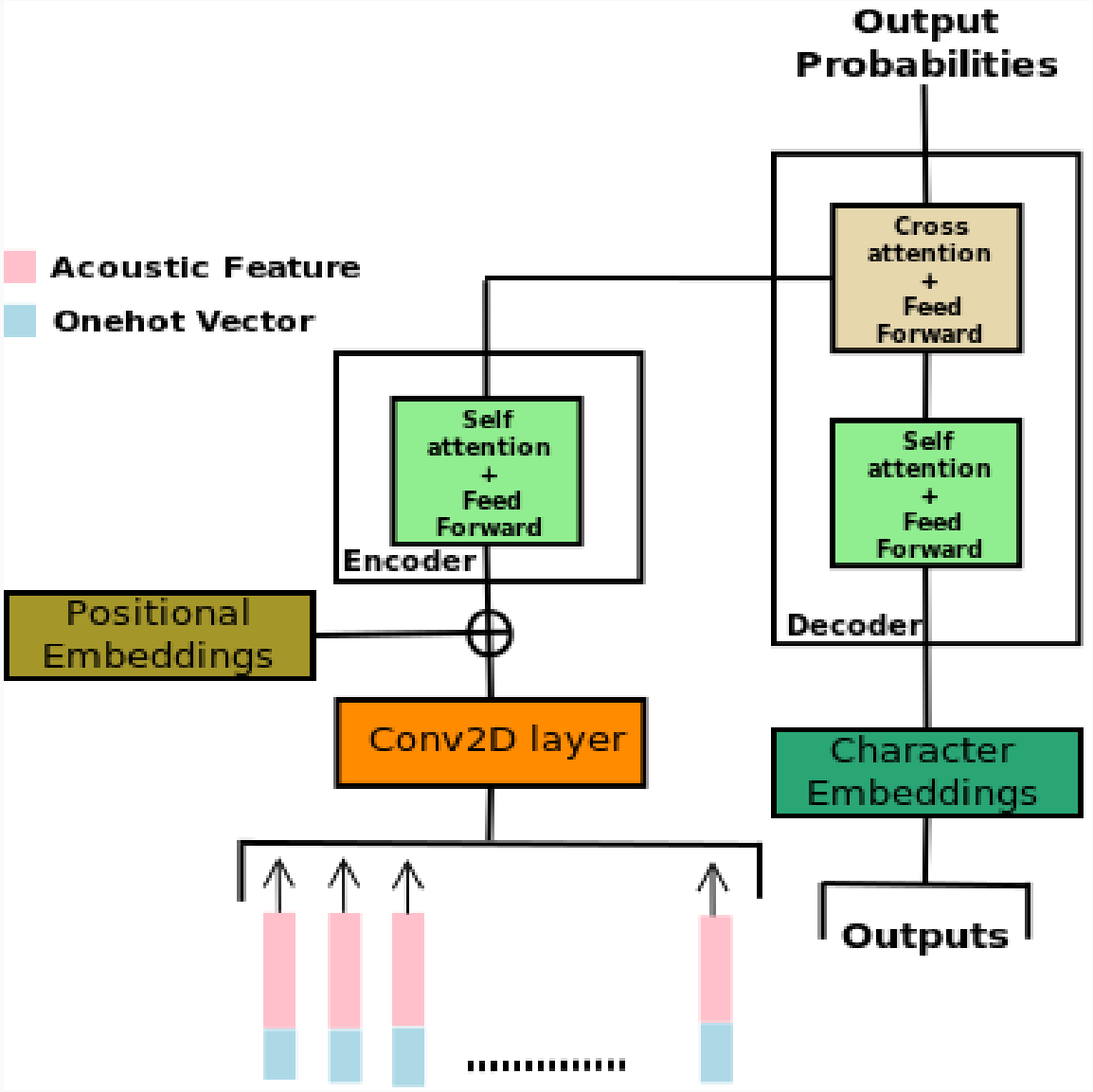
Strategy 2: Language information at the encoder

Provide language information at feature level:

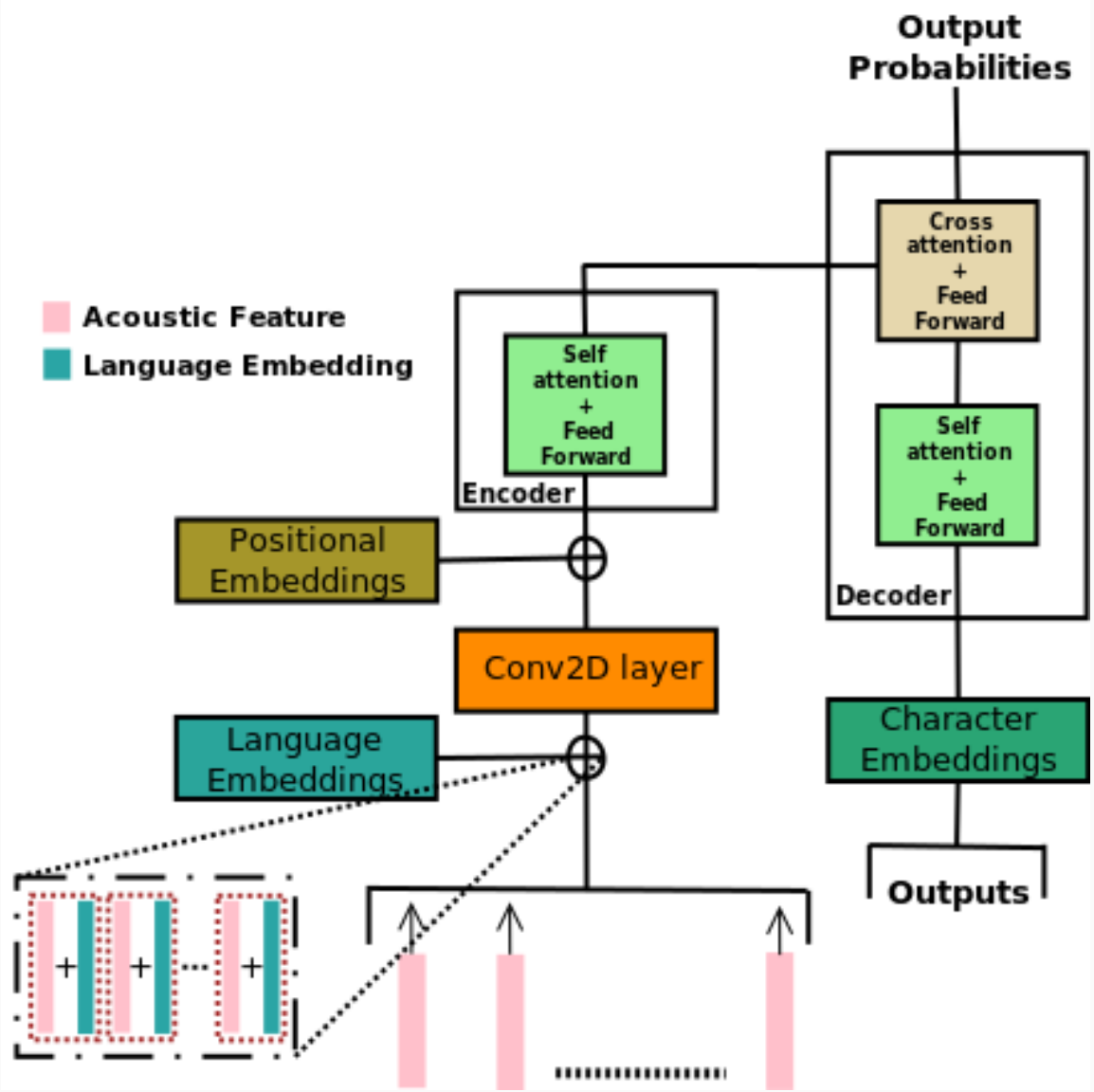
- Append one hot vector
- learn feature embeddings for the language

Language information is used while decoding, hence these models are **Language Specific Models**

Strategy 2a: Appending One hot vector



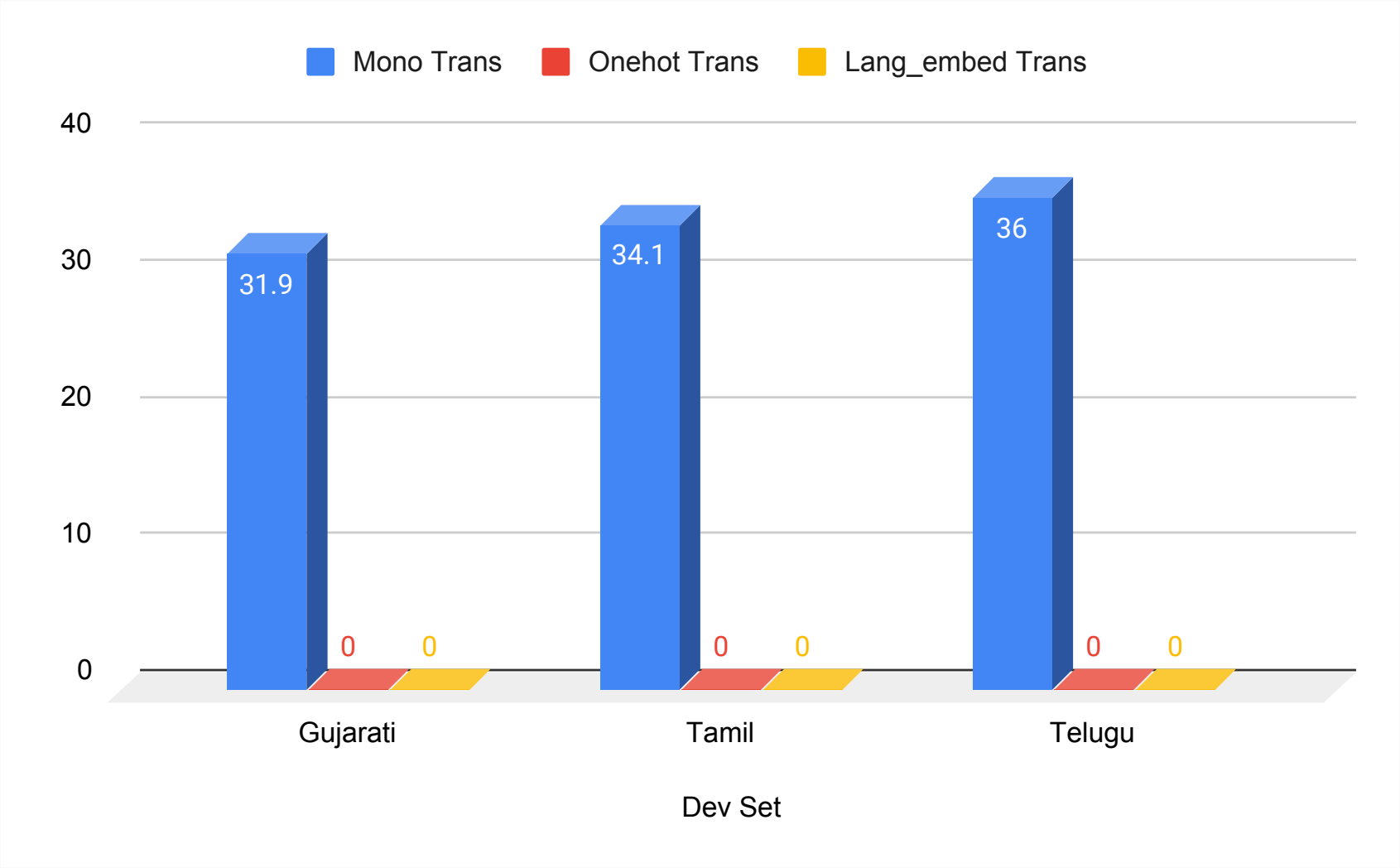
Strategy 2b: Learning Language embedding



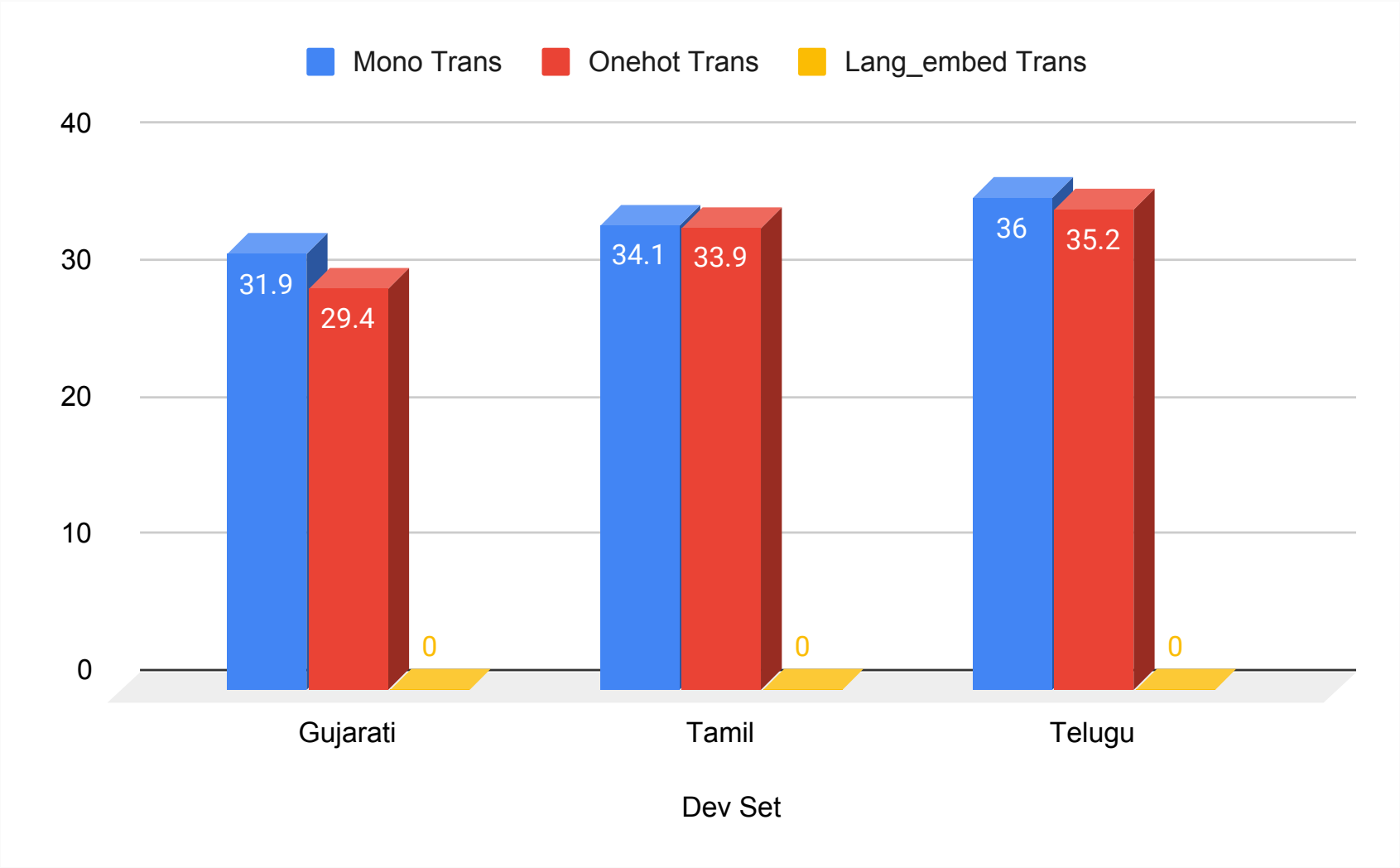
Strategy 2b: Learning Language embedding

- Analogous to how Character Embedding is learnt at the decoder
- Embedding vectors are initialized to a random vector with the dimension of acoustic feature
- Given an utterance - the targets belong to only one language - this information is used

Strategy 2: Results



Strategy 2: Results



Strategy 2: Results

Strategy 2: Results

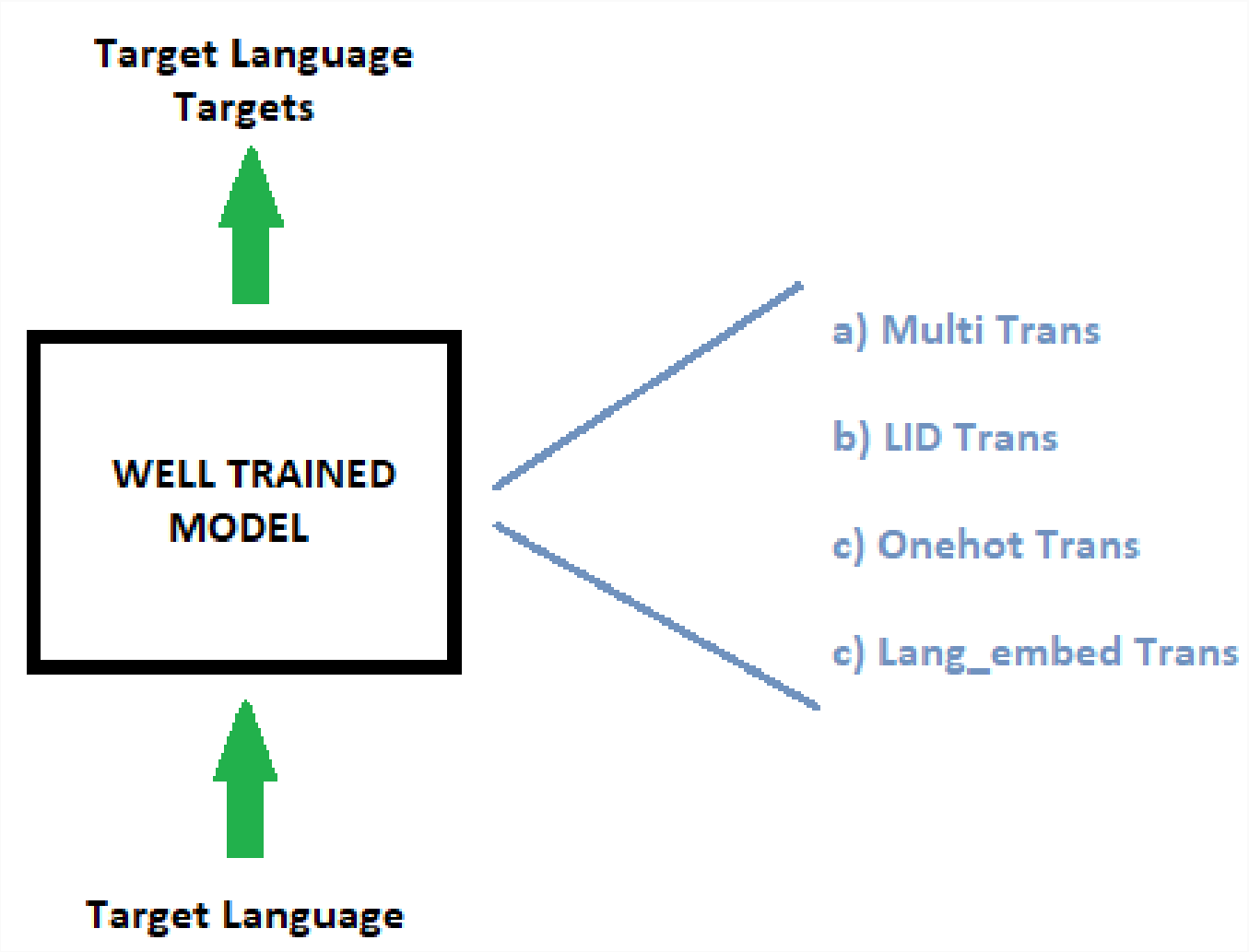
Strategy 2: Results

Strategy 2: Results

Fine tuning

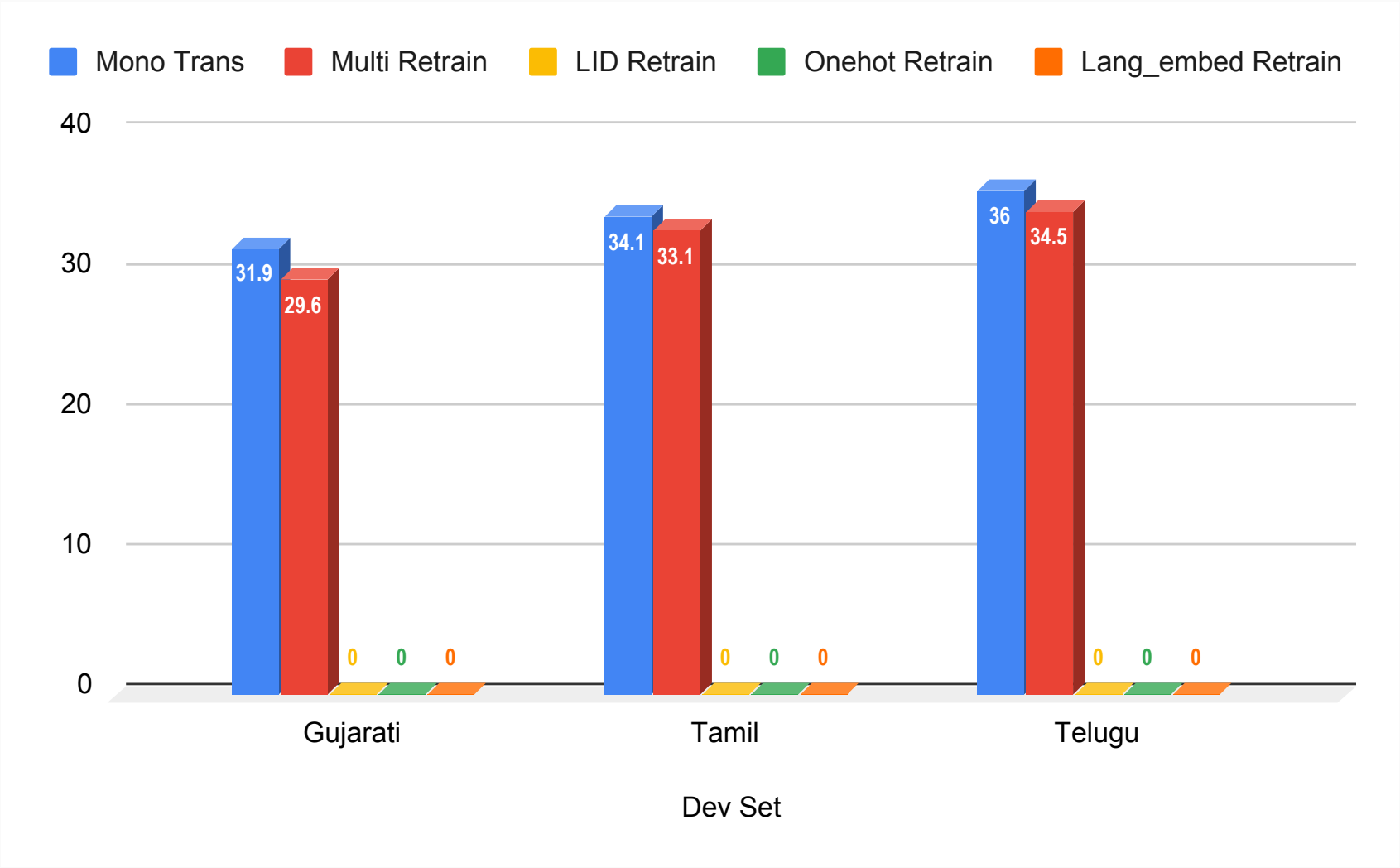
Fine tuning

Retrain well trained model using the target language

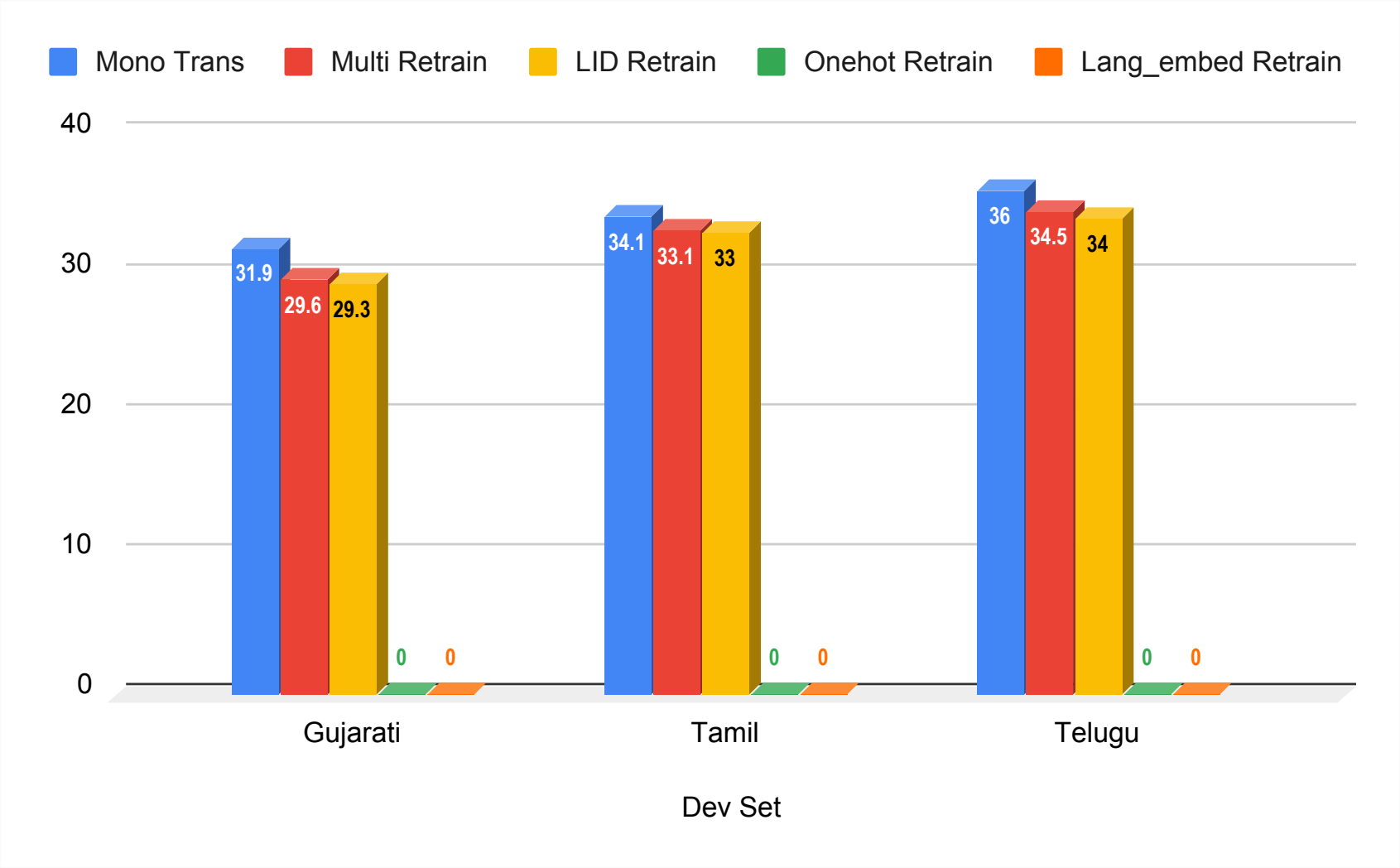


Results after retraining

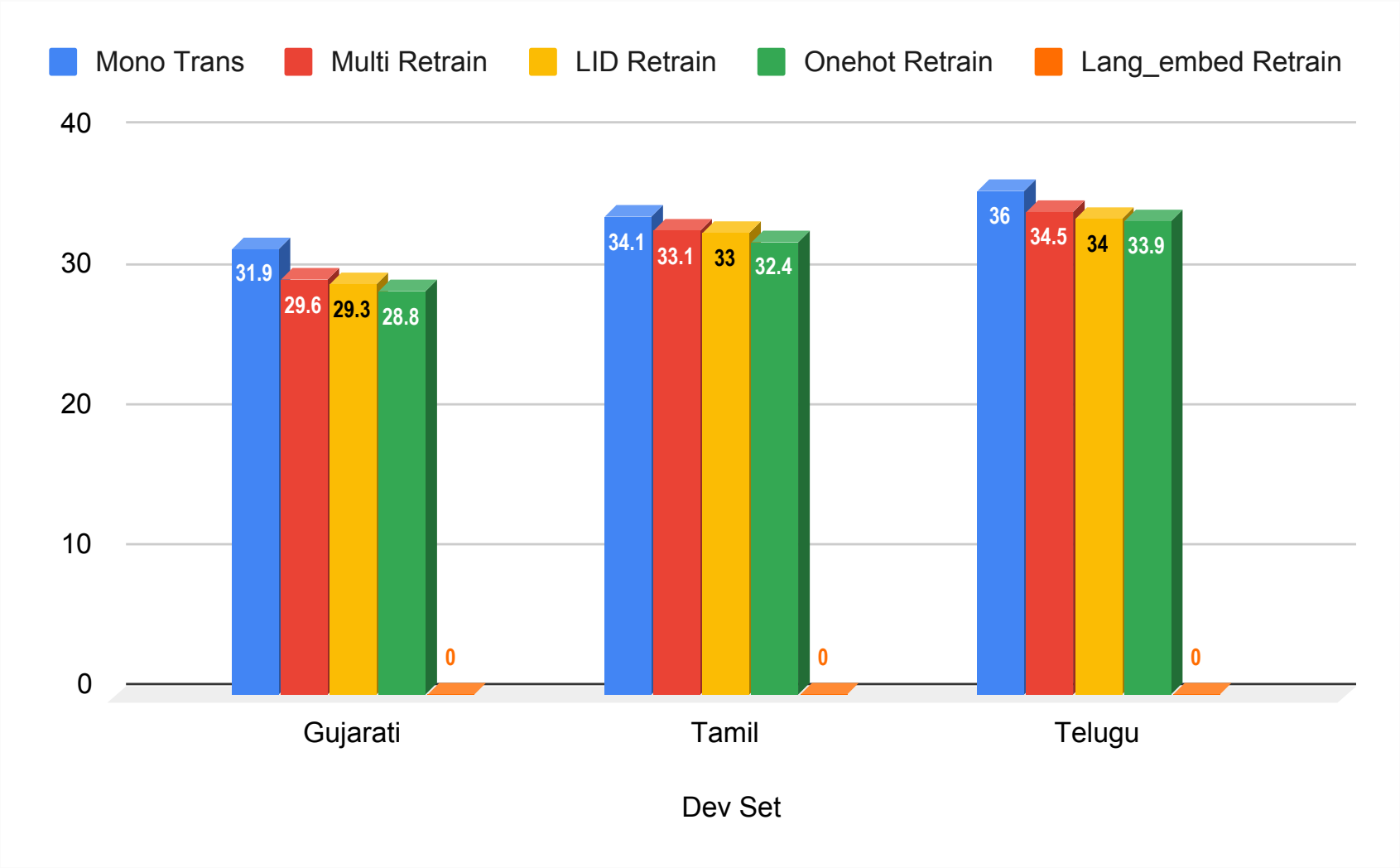
Results after retraining



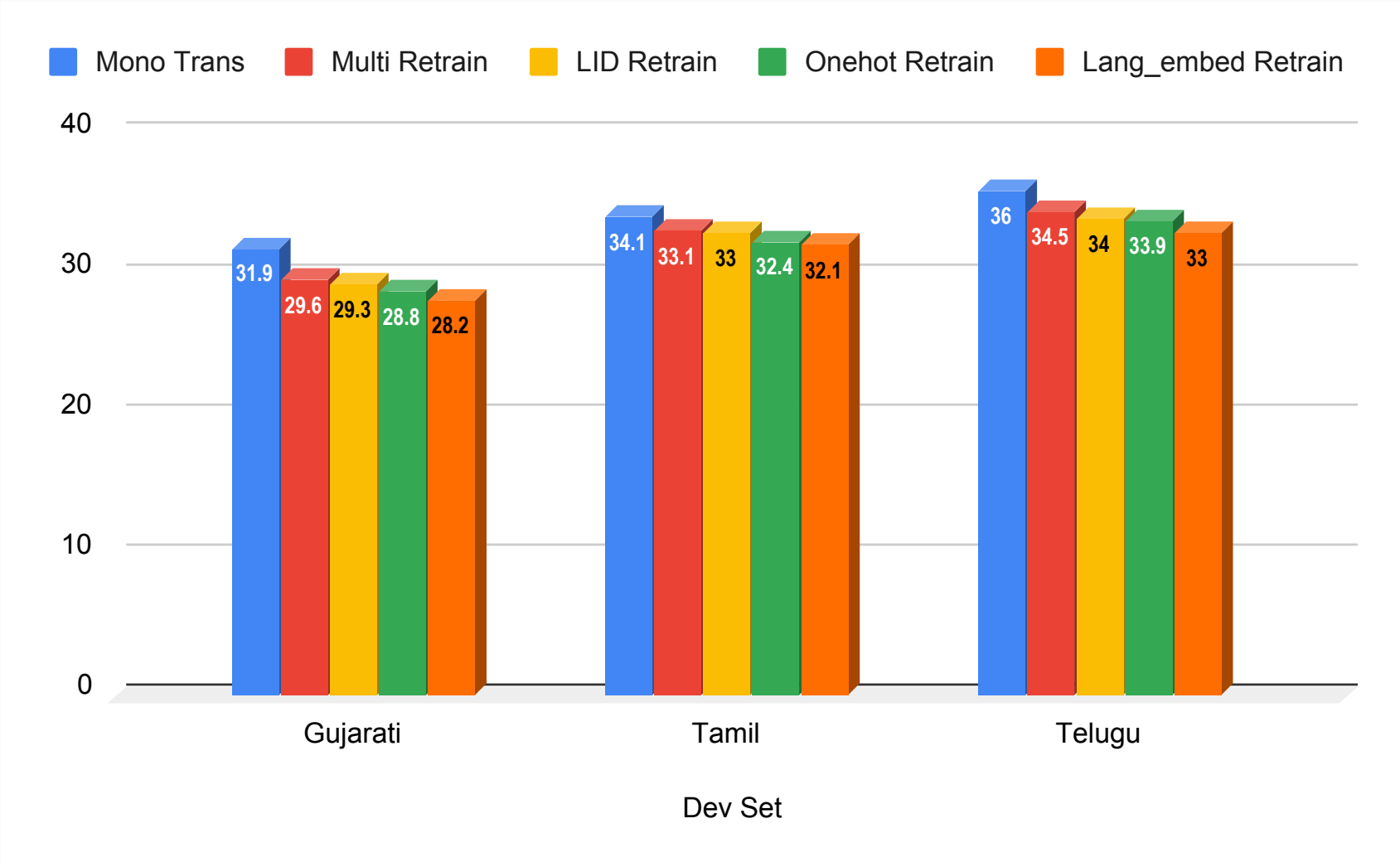
Results after retraining



Results after retraining

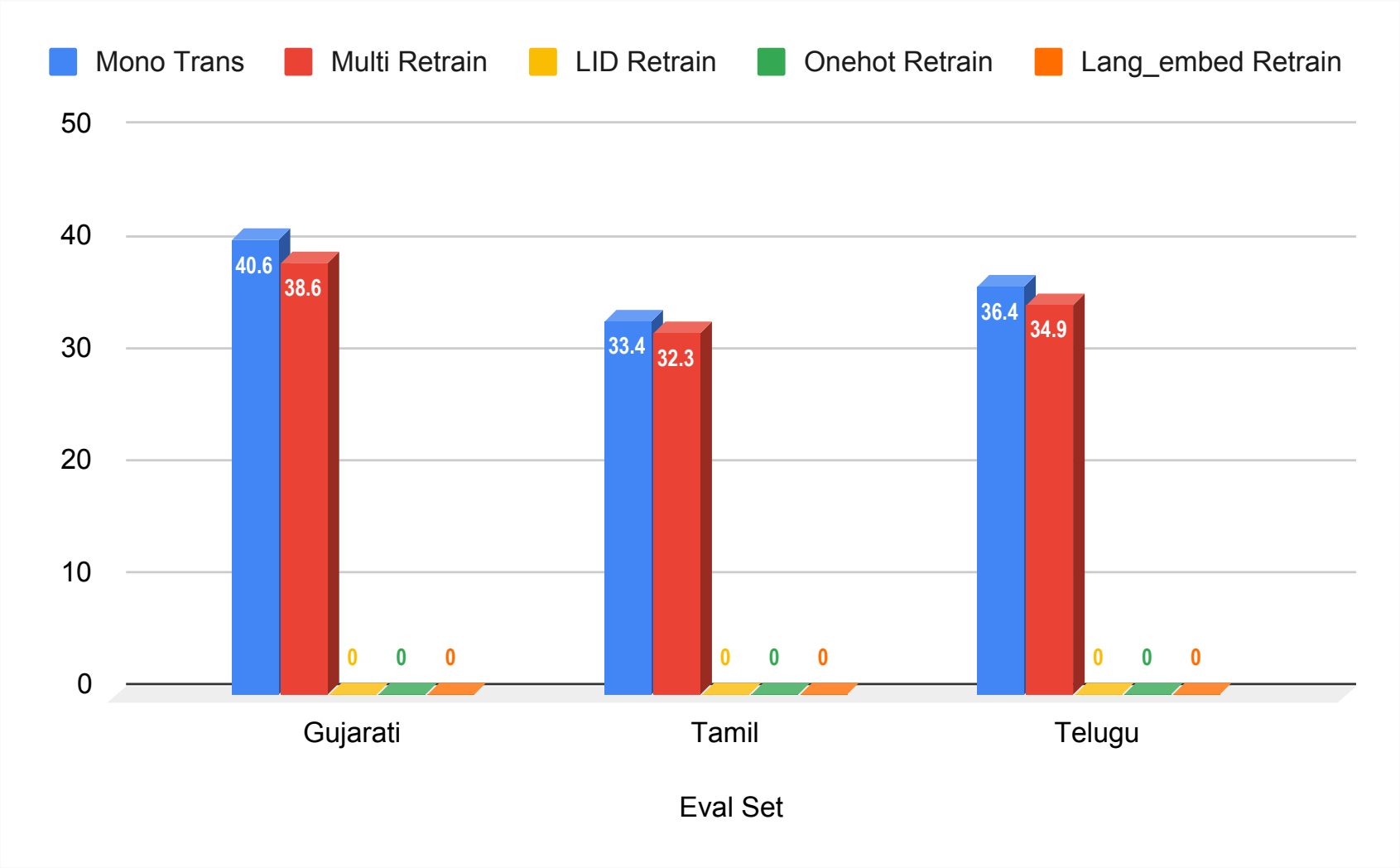


Results after retraining

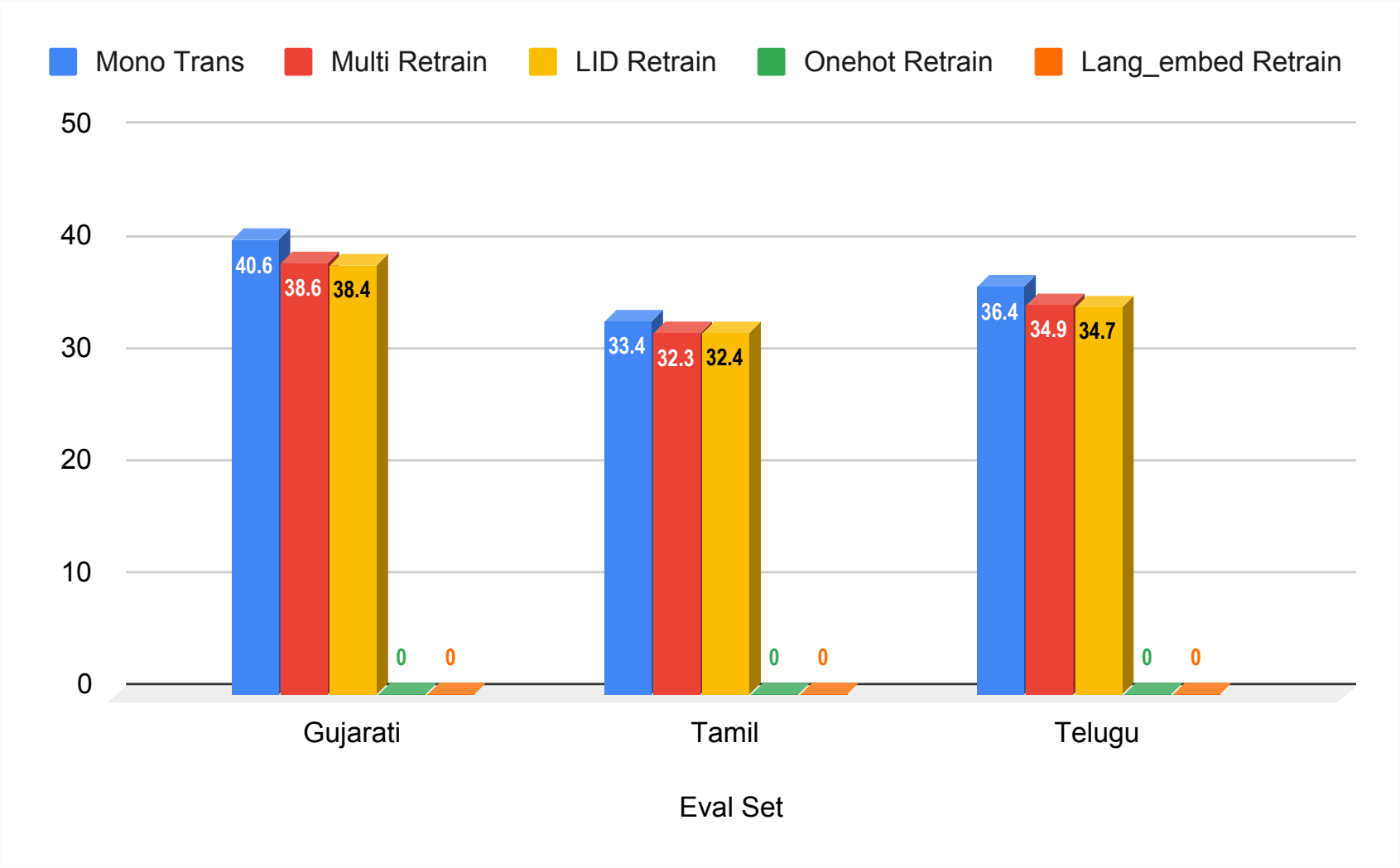


Results after retraining

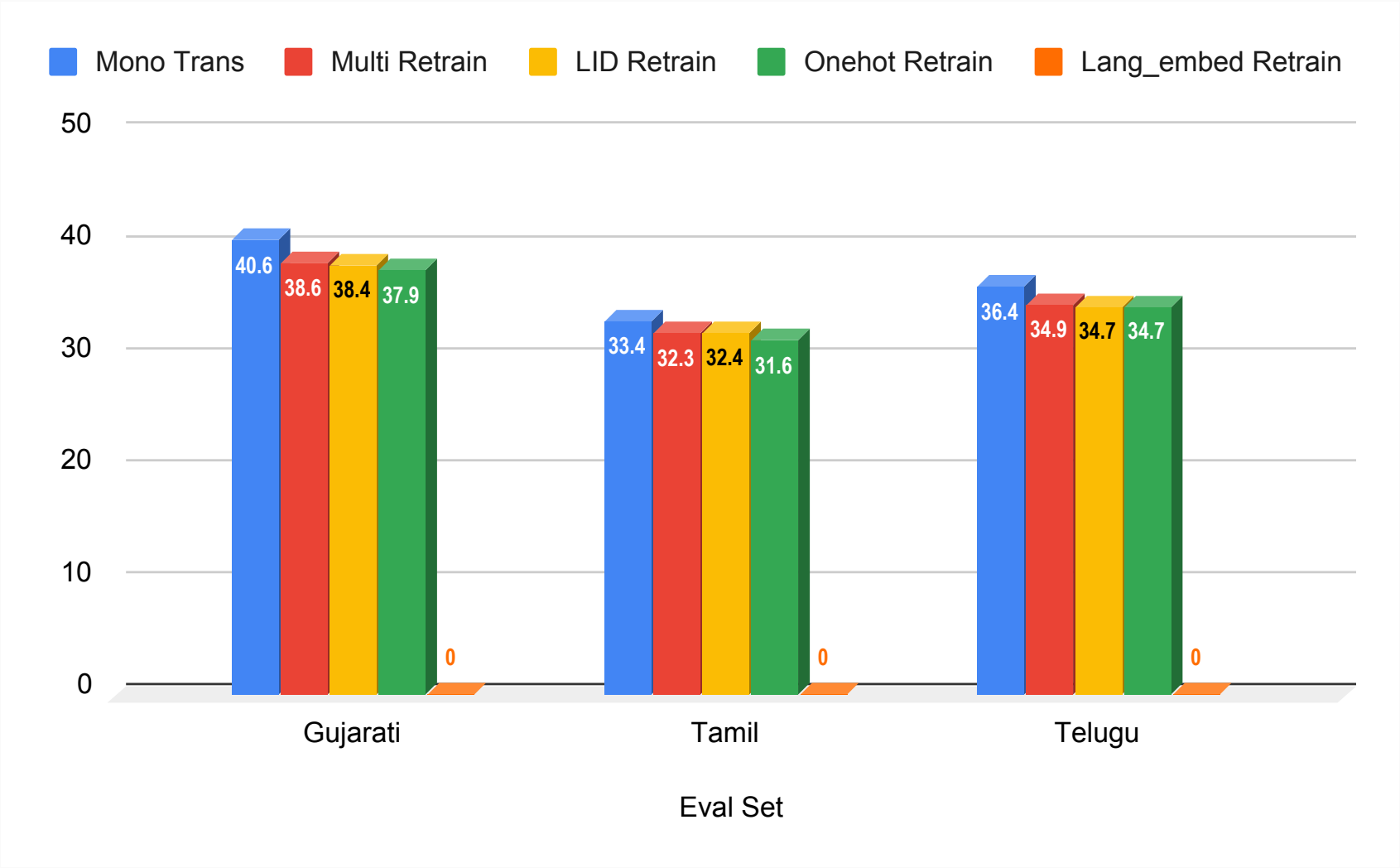
Results after retraining



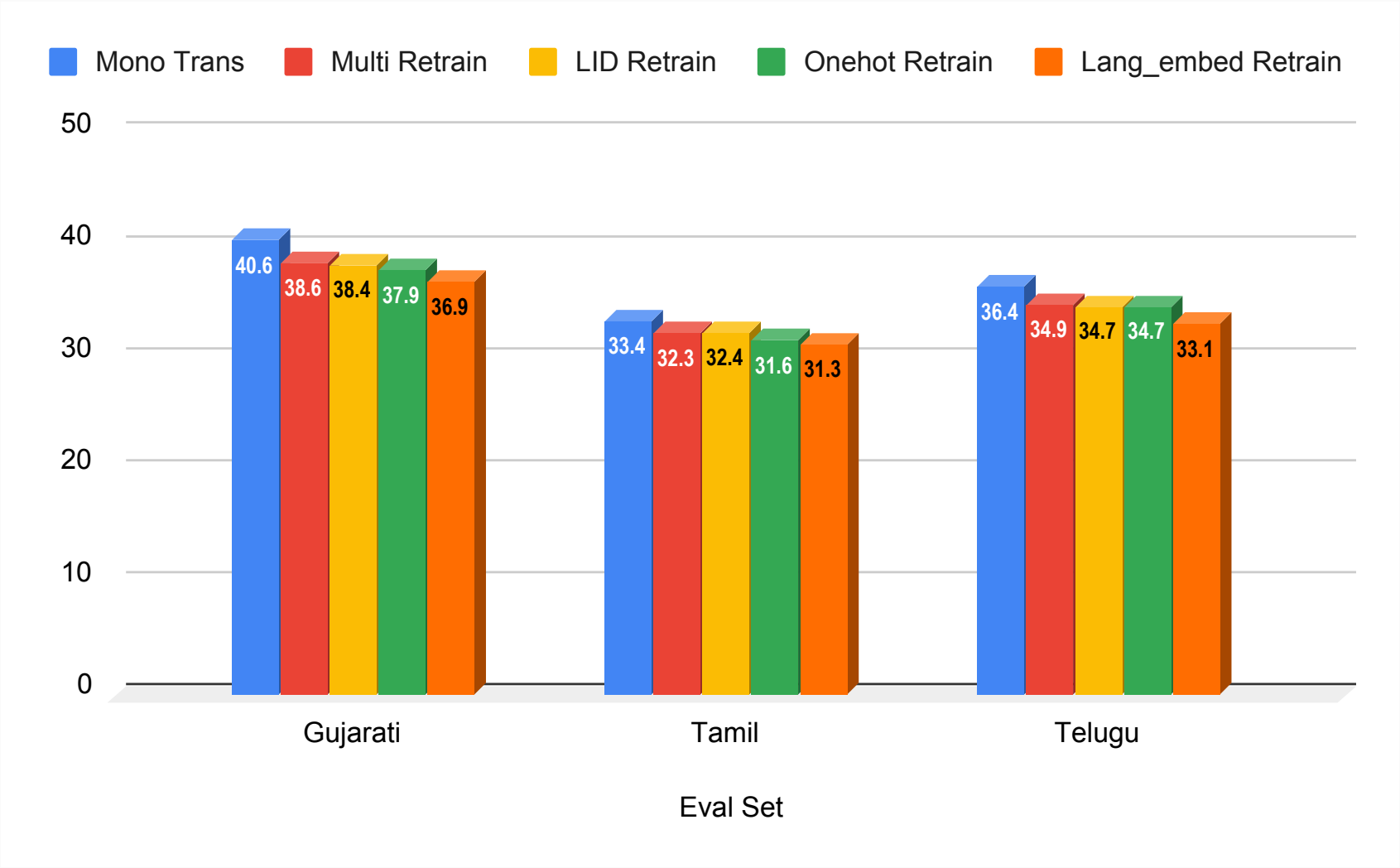
Results after retraining



Results after retraining



Results after retraining



Conclusion

Conclusion

- Making use of Language information while training/decoding improves model performance.
- Explored two ways of incorporating Language information
- Providing Language information at the Encoder by learning feature embeddings, gave the best performance.
- Fine tuning further improved the model performance.