Joint Transformer/RNN Architecture for Gesture Typing in Indic Languages

Emil Biju, Anirudh Sriram, Mitesh M. Khapra, Pratyush Kumar

Indian Institute of Technology Madras

Abstract

- Gesture typing is a fast and convenient method for providing textual input to touch based keyboards
- **Problem:** To perform gesture typing on Indic language keyboards by performing gesture input decoding, transliteration and spelling correction.
- We propose a CTC-Transformer based path decoder which, unlike prior approaches, does not assume co-character independence during decoding.
- We also introduce an ELMo-network based spelling correction model that uses spelling-aware character embeddings for comparing words.

Implementation Types

- **English-to-Indic decoding**

The input to this model is from an English keyboard. This swipe input is decoded using the CTC Gesture path Decoder,, transliterated and spell-corrected to obtain the output in the target Indic language.

- **Indic-to-Indic decoding**

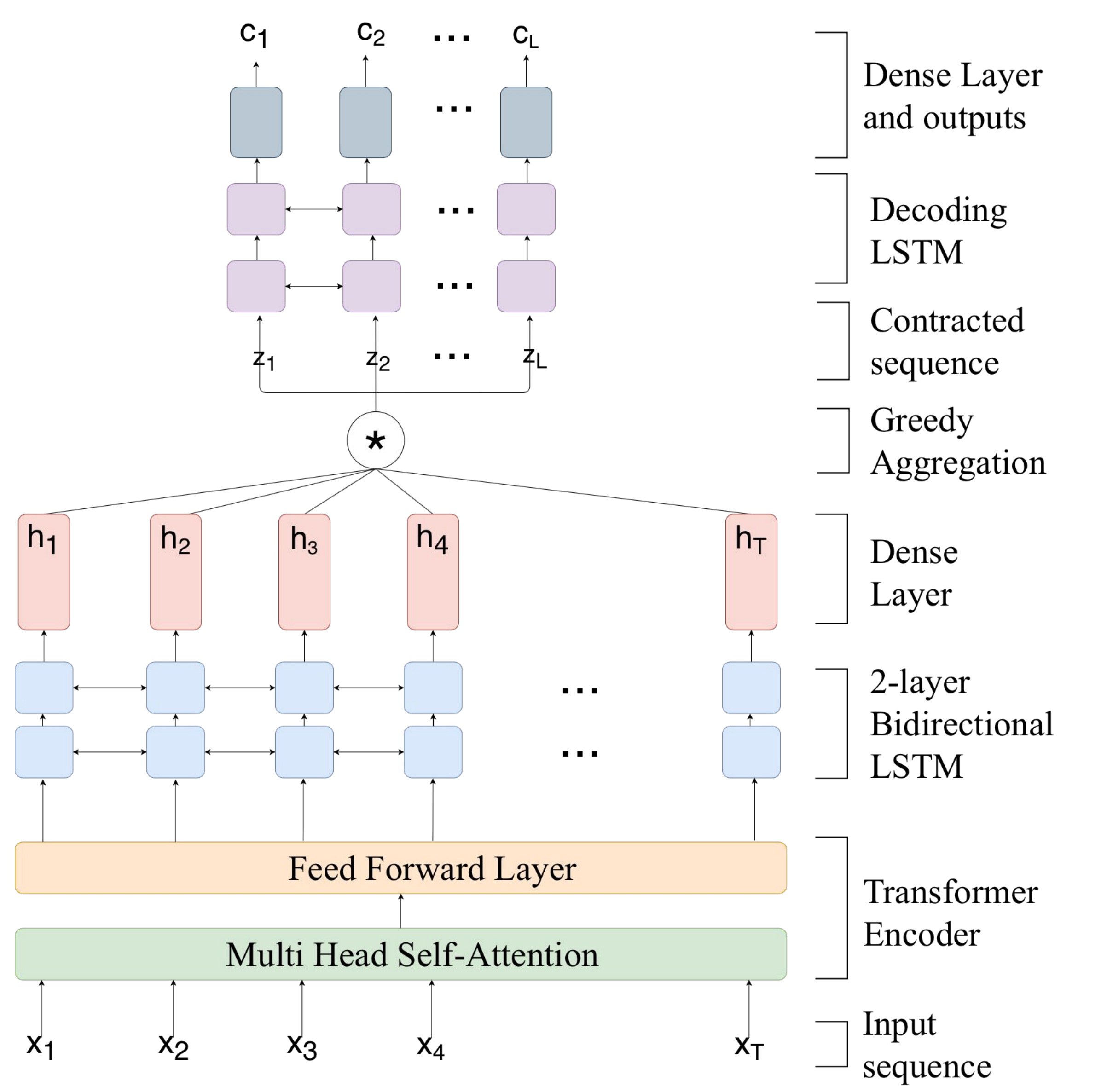
Here, the input is provided to an Indic character keyboard and the output is generated in the same Indic language. Thus, the Transliteration Generation module is removed from the pipeline for this case.

Gesture Data Generation

- The gesture paths are synthetically generated using the principle of jerk minimization which states that human motor control tends to choose the path of minimum jerk. We create 2 datasets for this work
- The first dataset contains 193,658 Indic words along with corresponding coordinate sequences of the gesture path.
- The second dataset contains 104,412 Indic-English transliteration scraped from Wikidata along with generated path coordinate sequences.
- Each sampled point contains the (x,y) coordinate locations, position derivatives and the character on the keyboard at that point.

CTC Gesture Path Decoder

Converts path coordinates into English/Indic characters sequences. Consists of a Transformer encoder, followed by Bidirectional LSTM network to produce an encoded representation. CTC loss function is used for optimization. Greedy aggregation is used to combine consecutive occurrences of the same character which are then passed to a character language model.



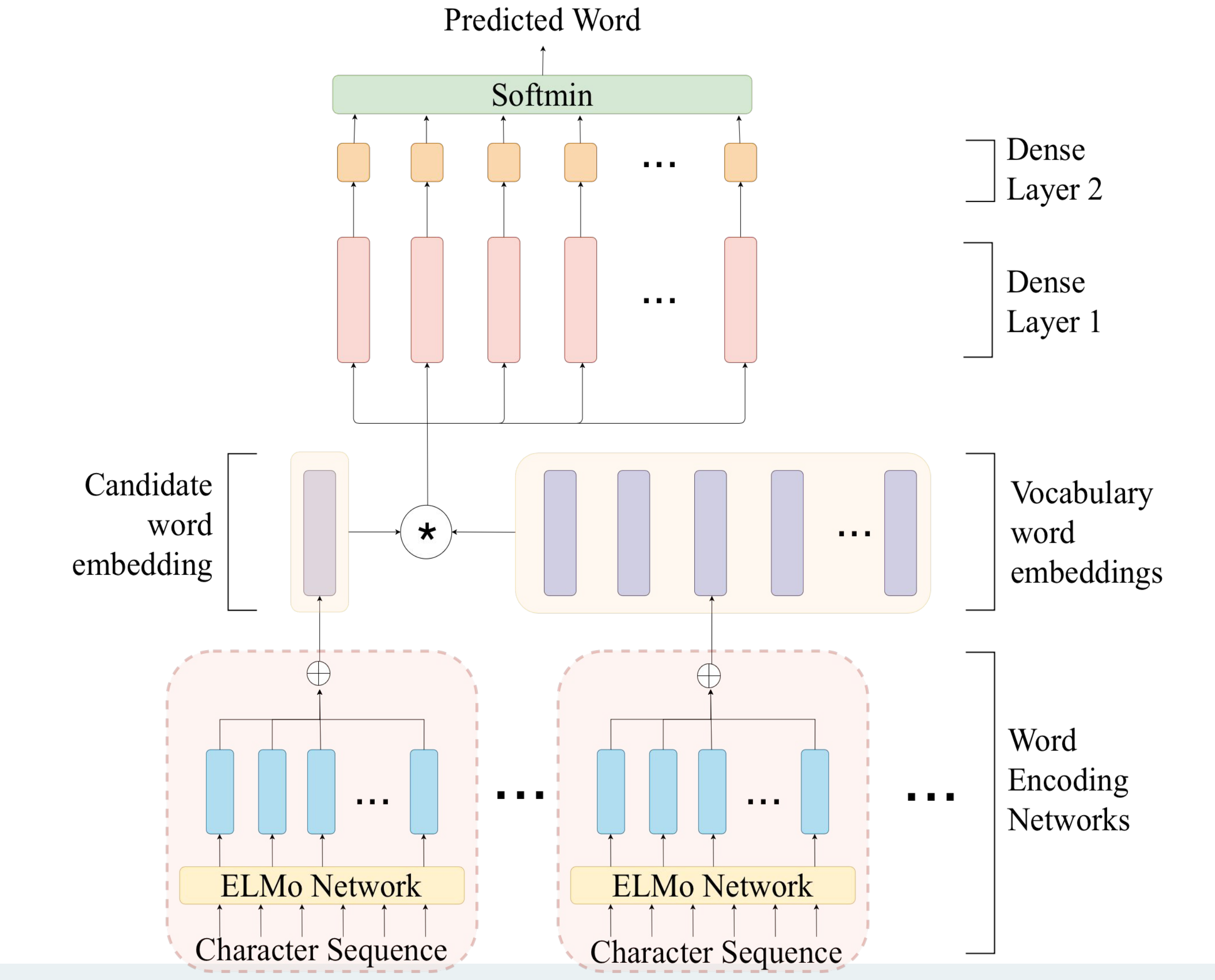
Experiments

- The correlation between word length and accuracy of decoding.
- We analyse the speed profile of the minimum jerk trajectory to explain the improtance of velocity derivative information and the occurrence of errors in path decoding.
- The variation in fraction of words belonging to each character type that are incorrectly predicted by the model.
- The variation in the angle between points on the keyboard in the words which were incorrectly predicted.
- The phonetic similarity between words which were incorrectly predicted.

Transliteration Generation and Spelling Correction

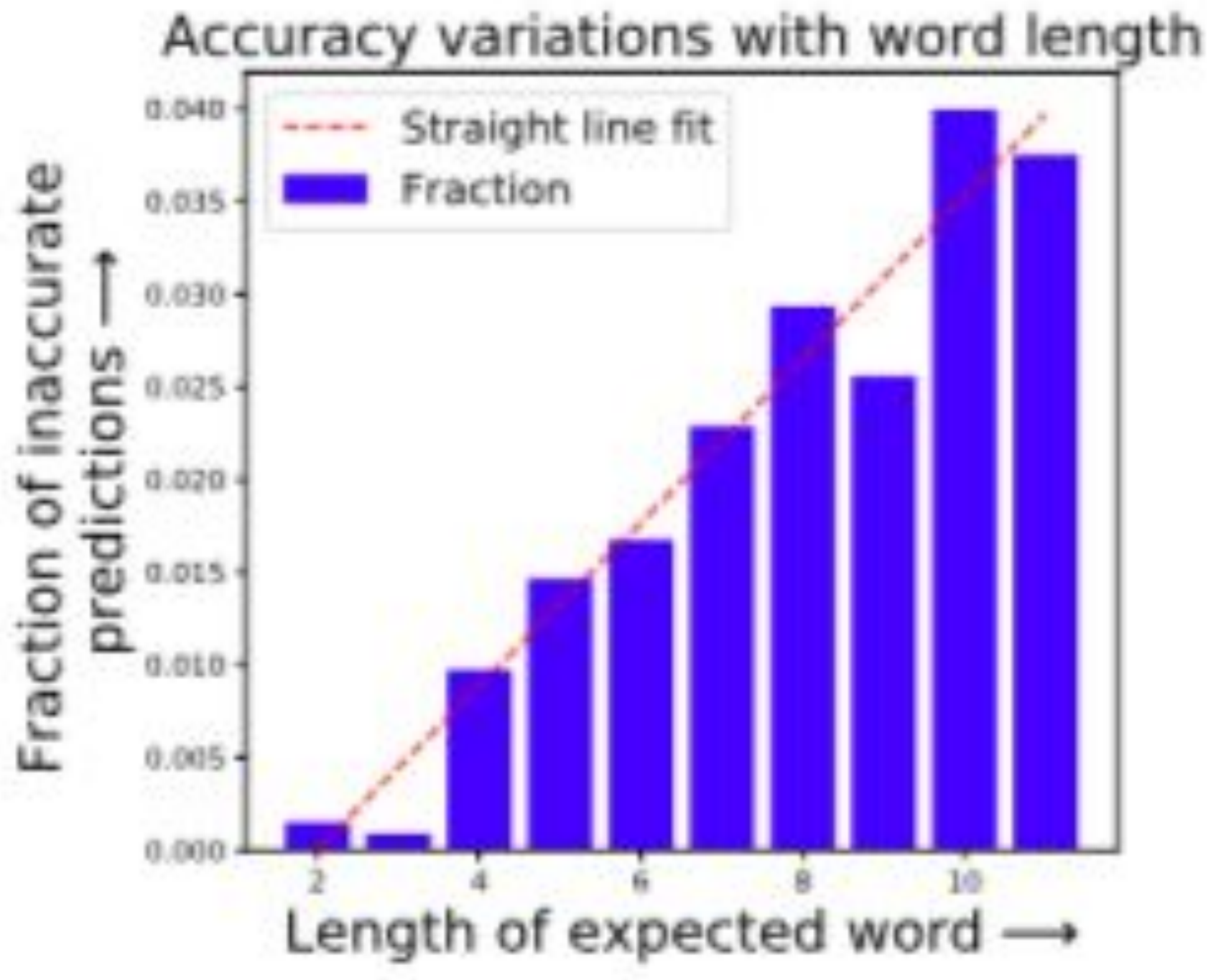
Generation: This model is used to transliterate the English character sequence generated by the Gesture Path Decoder into the Indic language. It uses a unidirectional GRU encoder with Bahdanau attention and Beam Search decoding with k=3.

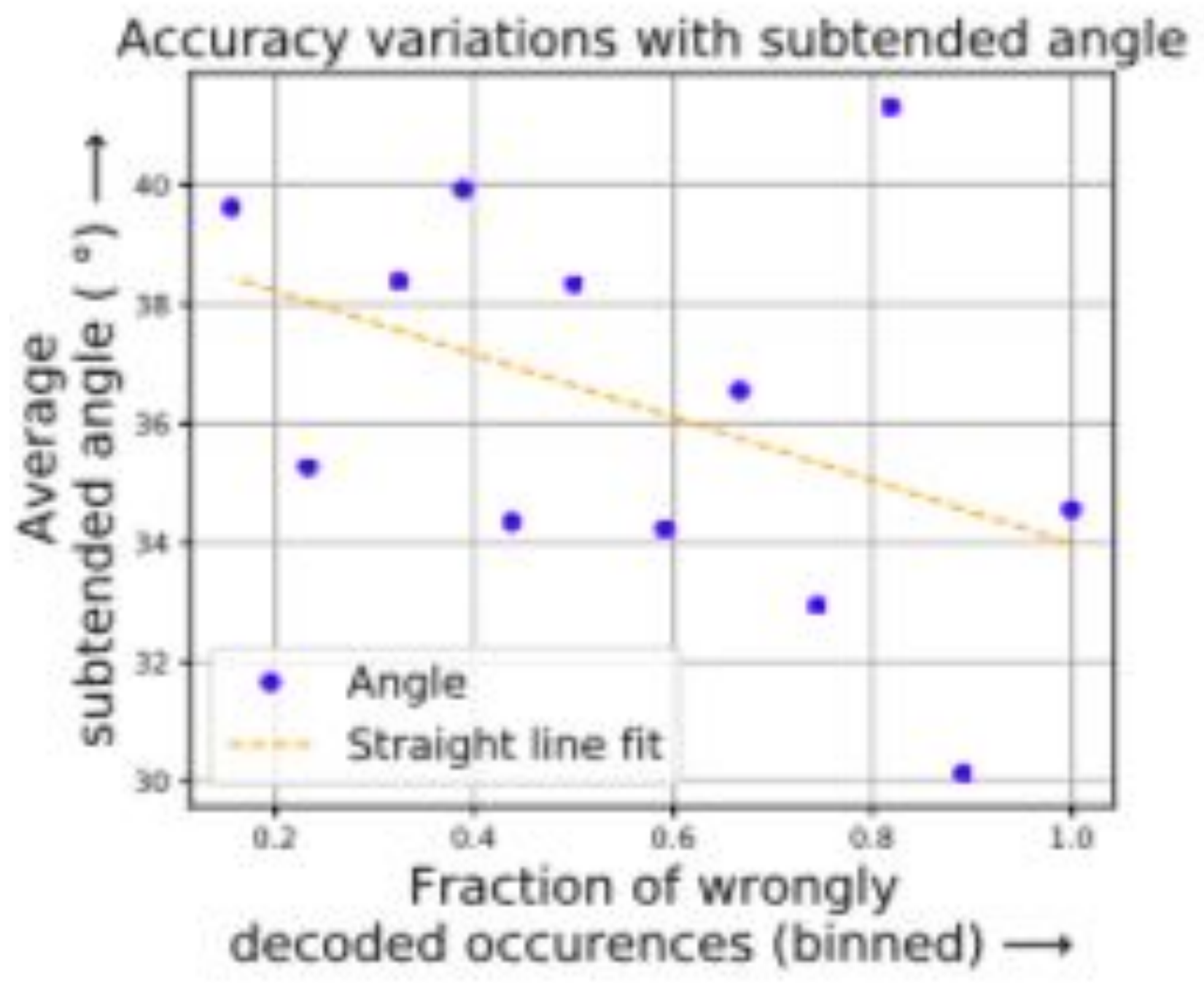
Correction: We use the ELMo network for generating character embeddings, which are then are summed together to obtain the word encoding. The resulting embeddings would depend on other characters in the word and their relative positioning, thus making it suitable for spelling comparison. The word encoding from the vocabulary with the shortest Euclidean distance to the candidate word is chosen.

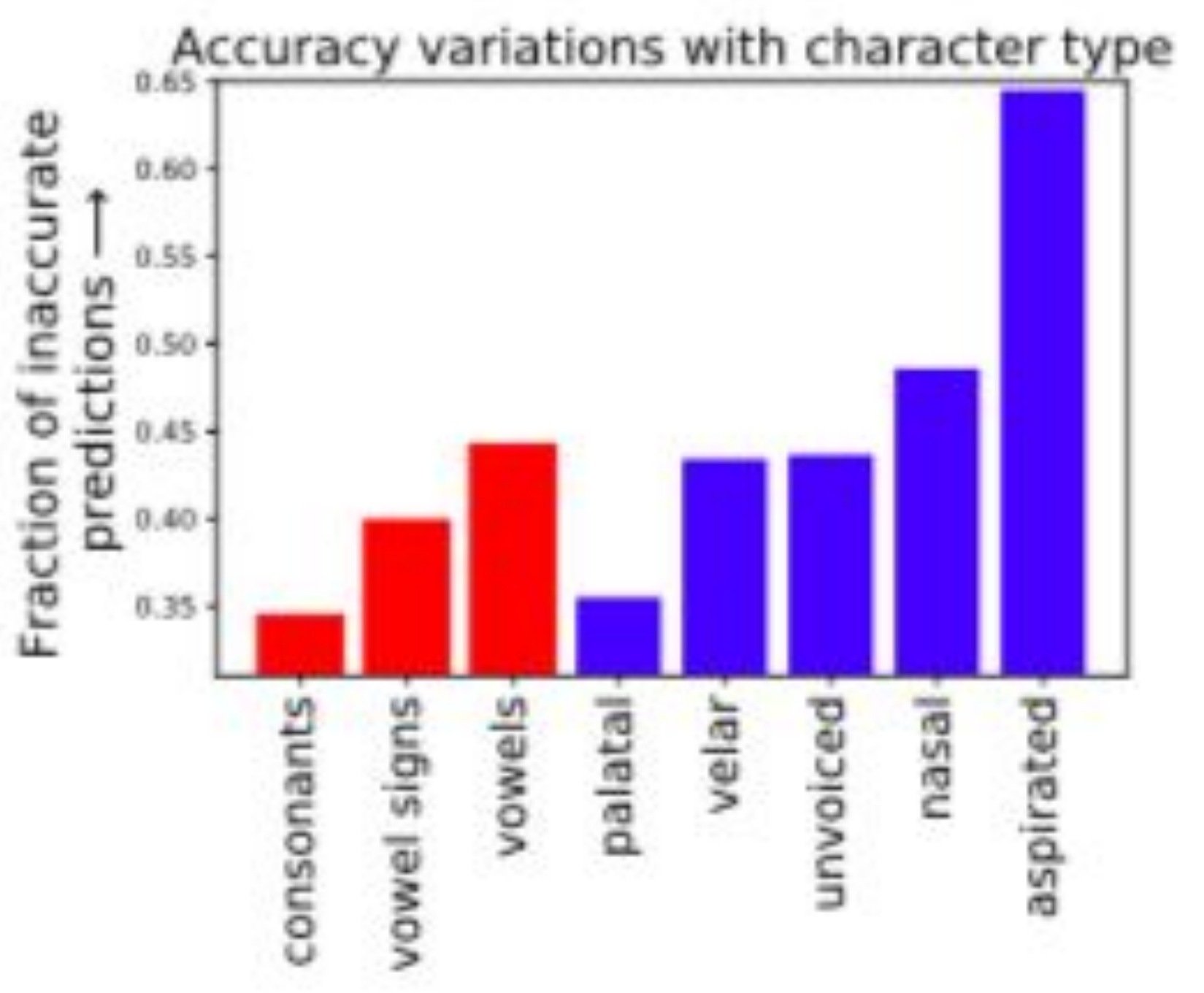


Results

- We analyzed our model's performance on both tasks across 7 different Indic languages, namely Hindi, Bengali, Gujarati, Tamil, Telugu, Kannada and Malayalam.
- The CTC decoder accuracy varies between 98.02% to 98.78% in the English-to-English decoding task. Similarly, the accuracy varies between 56.98% to 75.06% in the Indic-to-Indic decoding task.
- The overall accuracy in case of Indic-to-Indic decoding case varies between 86.75% in Hindi to 95.53% in case of Malayalam.The accuracy in case of English-to-Indic decoding varies between 70% in Malayalam to 89.12% in case of Hindi.







Conclusion

In this paper, we have demonstrated how a CTC-based path decoder and an ELMo-based spelling correction module can enable Indic gesture typing. Going further, we wish to extend our work to support visually impaired users and diversify our dataset to a larger set of languages.

https://iitmnlp.github.io/indic-swipe