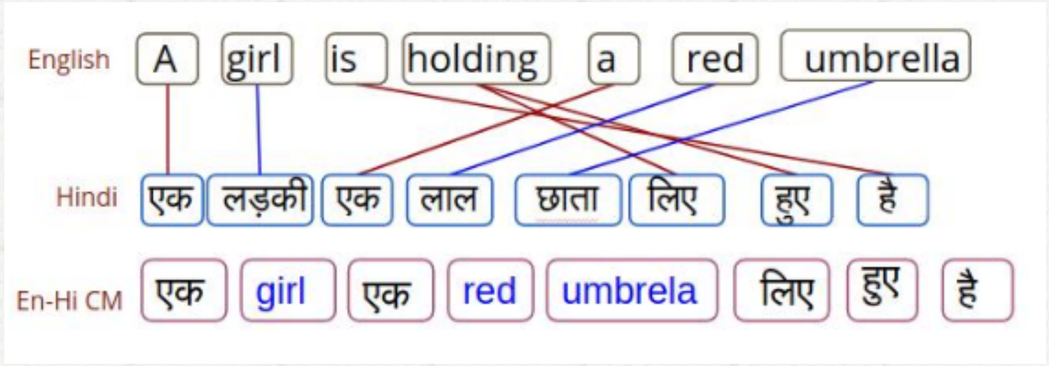
Project Presentation

Presented by Team 29

Code-Mixing

Code-mixing refers to the blending of two or more languages within a sentence or discourse, a phenomenon increasingly observed in multilingual societies and across digital communication platforms.
An example of code-mixing widely recognized in the Indian subcontinent is Hinglish, a blend of Hindi and English.



Problem Statement

We intend to devise a system capable of generating sentences that accurately represent this hindi and english fusion, using input data consisting of English sentences.

Dataset

We have used the HinGE Dataset. HinGE has Hinglish sentences generated by humans as well as two rule-based algorithms corresponding to the parallel Hindi-English sentences.

The dataset includes the following columns:

- English, Hindi: The parallel source sentences from the IITB English-Hindi parallel corpus.
- Human-generated Hinglish: A list of Hinglish sentences generated by the human annotators.
- Other columns include WAC and PAC rating of the words generated by WAC and PAC algo. respectively.

Baseline Models

Model	en-es			en-de			en-fr			en-hi		
	B	**R**	**M**	**B**	**R**	**M**	**B**	**R**	**M**	**B**	**R**	**M**
Seq2Seq	16.42	36.03	24.23	19.19	36.19	24.87	19.28	38.54	26.41	15.49	35.29	23.72
Attentive-Seq2Seq	17.21	36.83	25.41	20.12	37.14	25.64	20.12	39.30	27.54	16.55	36.25	24.97
Pointer Generator	18.98	37.81	26.13	21.45	38.22	26.14	21.41	40.42	28.76	17.62	37.32	25.61

These models serve as baselines due to their effectiveness in sequence-to-sequence tasks, with Seq2Seq providing a basic framework, Attentive Seq2Seq improving attention mechanisms, and Pointer Generator addressing vocabulary limitations by enabling direct copying from the input. They collectively offer a solid foundation for codemixed language generation tasks.

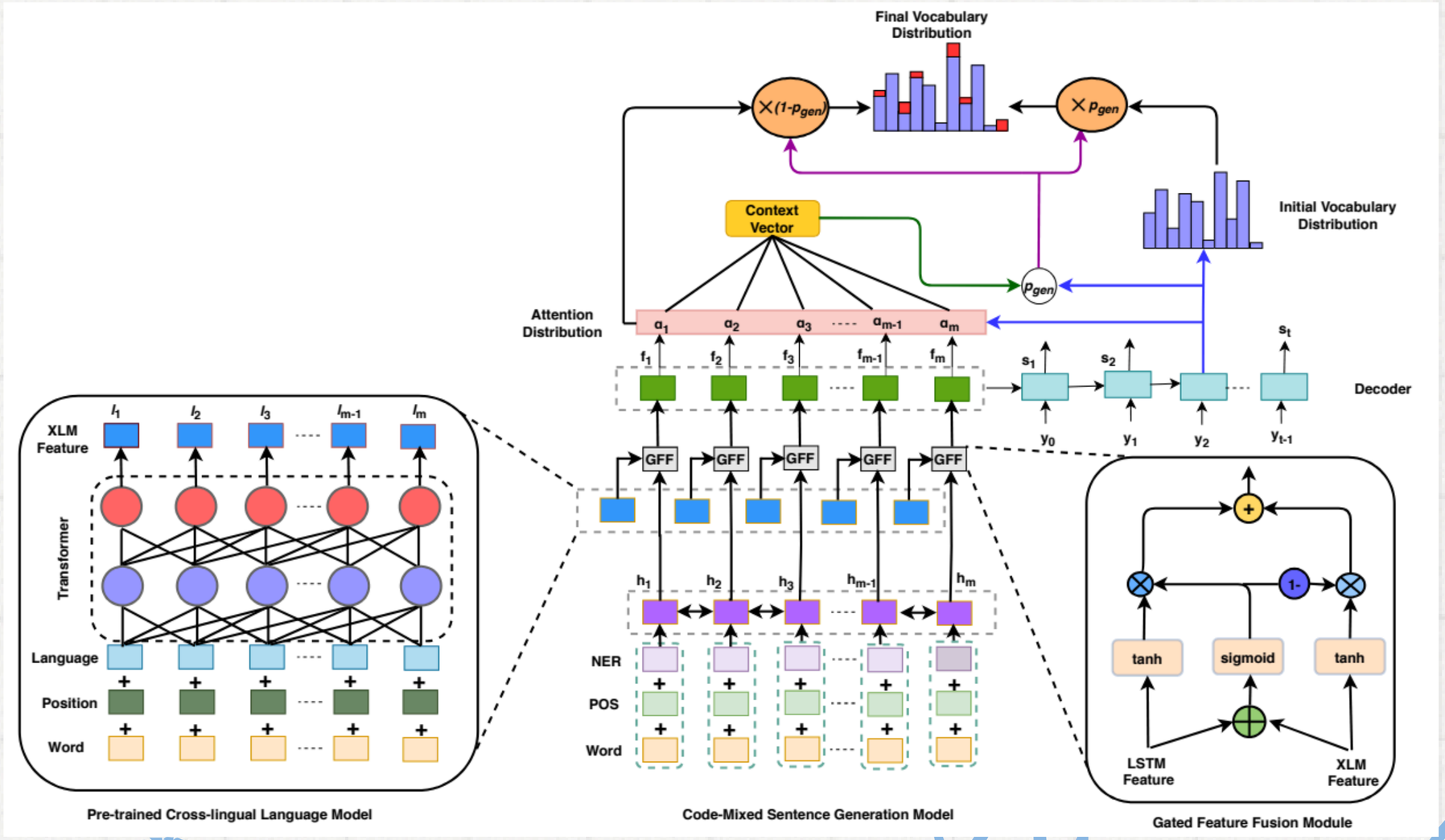
Literature Review

- **Semi-supervised Approach to Generate the Code-Mixed Text using Pre-trained Encoder and Transfer Learning by Deepak Gupta, Asif Ekbal, Pushpak Bhattacharyya:** Proposes a model leveraging pre-trained language models and neural architecture search techniques for generating code-mixed Hinglish dialogues, emphasizing context and linguistic constraints.

- **A Comprehensive Understanding of Code-mixed Language Semantics using Hierarchical Transformer by Ayan Sengupta*, Tharun Suresh*, Md Shad Akhtar, and Tanmoy Chakraborty:** Advances the understanding of code-mixed language semantics using a hierarchical transformer model, providing insights into semantic coherence and enhancing text generation.

- **Marathi-English Code-mixed Text Generation and L3Cube-HingCorpus and HingBERT:** Contributes specialized datasets and models like HingCorpus and HingBERT for code-mixed language processing, marking crucial progress in resource development.

Implementation



Libraries Used: The code uses libraries such as PyTorch , spaCy for natural language processing tasks, BERT tokenizer, XLM-RoBERT and pickle.

Data Preparation: A dataset (TextDataset) is created with English sentences, tokenized using BERT tokenizer, and annotated with part-of-speech tags and named entity recognition.

Modules:

Attention: Defines the attention mechanism used in the LSTM-Attention module.
LSTM-Attention: Implements the LSTM-based sequence-to-sequence model with attention.
TextDataset: Custom dataset class for loading and processing text data.
TextEncoder: Encodes text inputs into embeddings using an LSTM-based encoder.
XLMEncoder: Encodes text inputs using a pre-trained XLM-RoBERTa model.
GatedFeatureFusion: Integrates the outputs of TextEncoder and XLMEncoder using a gating mechanism.

Metrics for Evaluation

For evaluation of our model we used BLEU and METEOR