INDIAN INSTITUTE OF TECHNOLOGY JODHPUR

Grammatical Error Correction for Indian Language

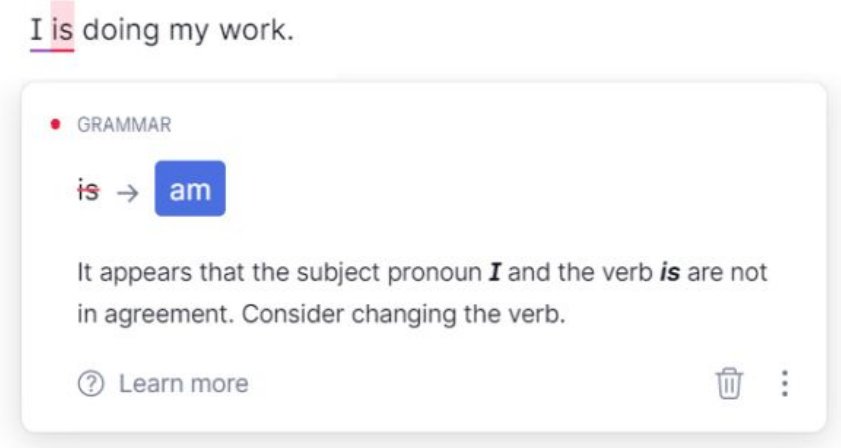
Presented by: Puja Gupta (M21MA004)
Supervisors: Dr. Gaurav Harit and Dr. Kirankumar Hiremath

Project Outline

- Grammatical Error Correction (GEC)
- Motivation
- Problem Formulation
- Phase Wise Implementation
- Experiments on English GEC
- Proposed method for Hindi GEC
- Dataset Used
- Results
- Conclusion
- Future Wok

What is GEC task in NLP?

- Grammatical Error Correction (GEC) is the task of error detection and correction in the text.
- In NLP, GEC can be formulated as a sequence-to-sequence task, where a model is trained using grammatically incorrect sentences as input and return a grammatically correct sentence.



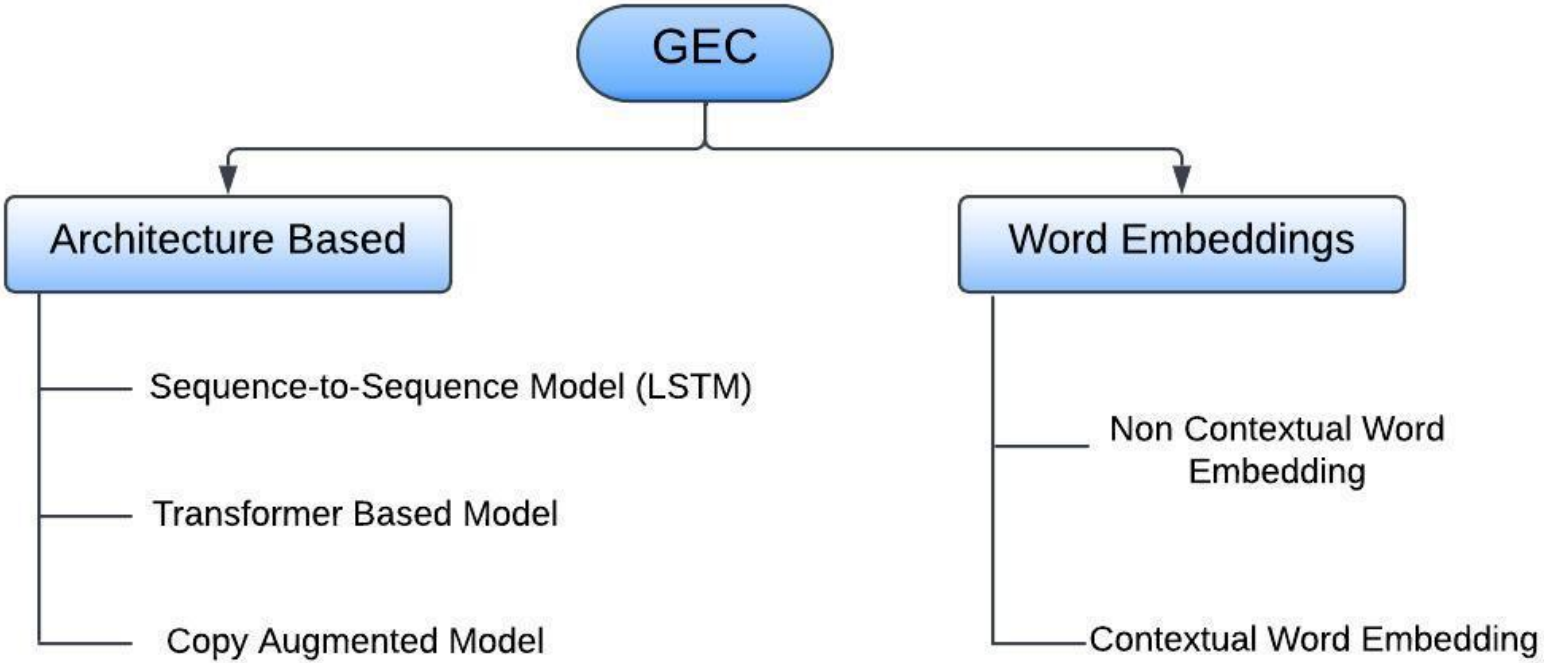
Motivation

- A lot of research and trained models are available for English Language.
- Minimal amount of work on GEC for Indian Languages.
- Need a well trained model for implementing GEC tasks for Indian languages which can help the writers.
- Proposed the work towards GEC for Hindi Language.

Input Sentence : मैं अपने एम.टेक प्रोजेक्ट पर काम कर रही है |

Correct Sentence : मैं अपने एम.टेक प्रोजेक्ट पर काम कर रही हूँ |

Problem Formulation



Phase 1 (Semester 3)

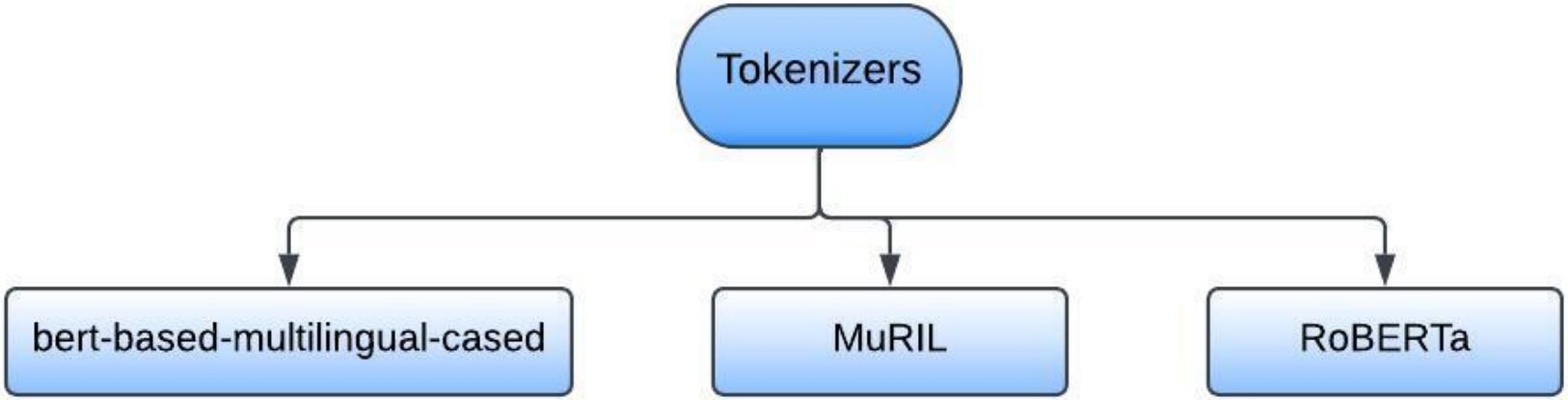
- Literature Review
- Sequence-to-Sequence Architecture (English & Hindi)
  - without Attention
  - with Attention
- Word Embeddings Used
  - Word2Vec
  - Glove
  - FastText

Phase 2 (Semester 4)

- Transformer-Based Architecture (English & Hindi)
  - Encoder-Decoder
  - T5ForConditionalGeneration
  - mT5ForConditionalGeneration
- Copy Augmented Transformer Model
- Comparison with present State-of-the-art Model

Tokenizers

- Breaks down text into smaller unit called tokens.
- Tokenization converts a text into a list of integers
- Embedding converts the list of integers into a list of vectors.
- Choice of tokenizer can have an impact on how the text data is processed and represented.
- Affect the accuracy and efficiency of the models trained on this data.



Text : "मैं अपने एमटेक प्रोजेक्ट पर काम कर रही हूँ।"

Tokens (*Bert-base-multilingual-cased*): ['म', '##ैं', 'अ', '##पने', 'ए', '##म', '##टे', '##क', 'प', '##्र', '##ोज', '##ेक्ट', 'प', '##र', 'क', '##ाम', 'क', '##र', 'र', '##ही', 'ह', '##ूँ', '।']

**Tokens (RoBERTa-hindi-guj-san*)*:['म', 'ैं', 'Ġअ', 'प', 'न', 'े', 'Ġएम', 'ट', 'े', 'क', 'Ġप', 'र', 'Ġक', 'ा', 'म', 'Ġक', 'र', 'Ġर', 'ह', 'ी', 'Ġह', 'ू', 'ँ', '।']

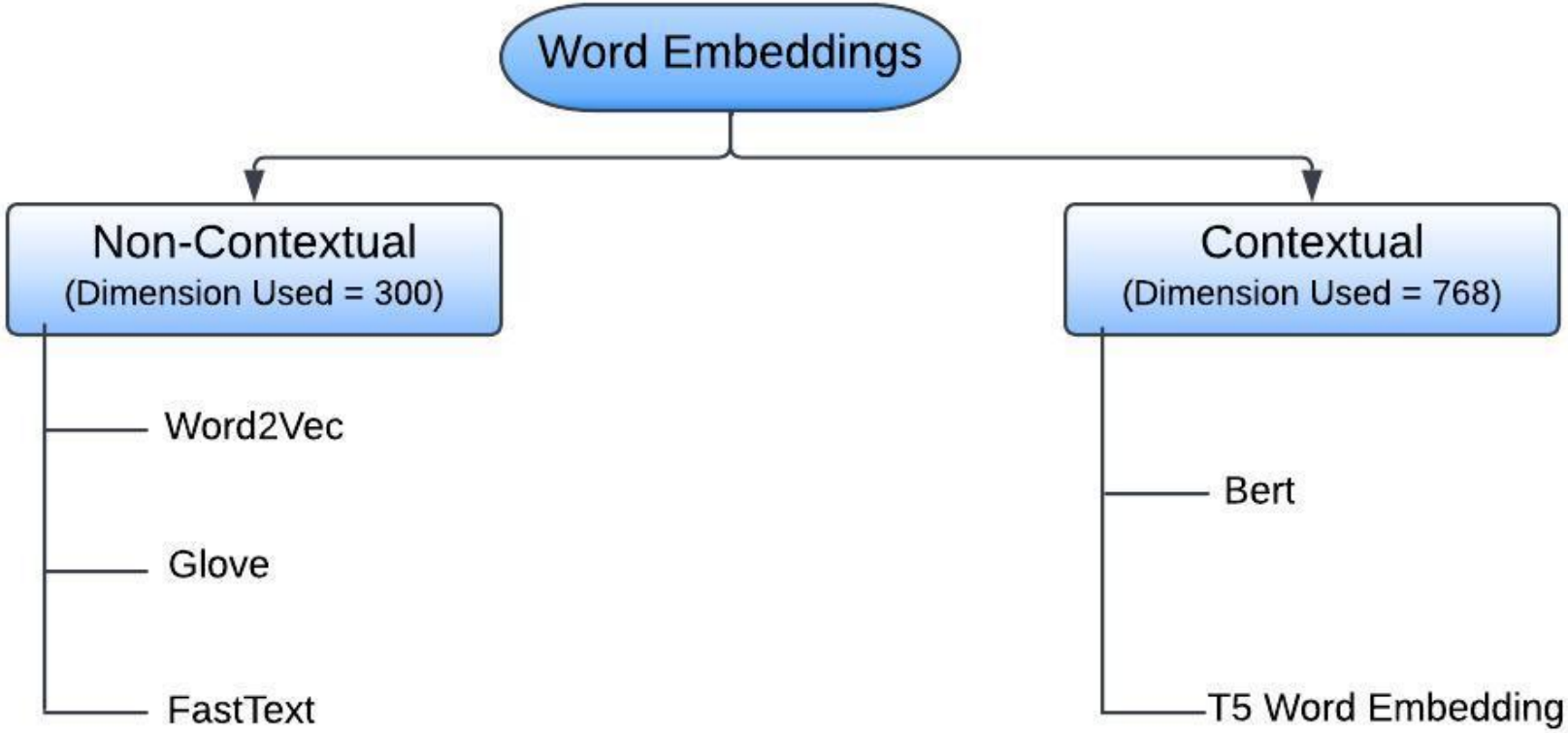
Tokens (*MuRIL)*: ['मैं', 'अपने', 'एमटेक', 'प्रोजेक्ट', 'पर', 'काम', 'कर', 'रही', 'हूँ', '।']

Tokenizer	Algorithm Used	Vocabulary Size	Tokenization
Bert-base-multilingual-cased	WordPiece	119547	Sub word level
MuRIL	WordPiece + Rule based method for Indian languages	197285	Word / Subword level
RoBERTa-hindi-guj-san	Byte Pair Encoding	30522	Sub word level

- Larger vocabulary allows to capture more fine-grained linguistic nuances and to handle rare or out-of-vocabulary words more effectively.

Word Embeddings

- Machines does not understand the textual data directly.
- Need to convert the textual data into some word vector representation and feed that to the model for processing.
- Word Embeddings enable to store textual information in a low dimensional vector.
- Words that occur in similar contexts tend to have similar embeddings.
- The vectors are learnt by models, through unsupervised learning.


Word Embeddings
Non-Contextual
(Dimension Used = 300)
Contextual
(Dimension Used = 768)
Word2Vec
Glove
FastText
Bert
T5 Word Embedding

Contextual Vs Non Contextual Word Embedding

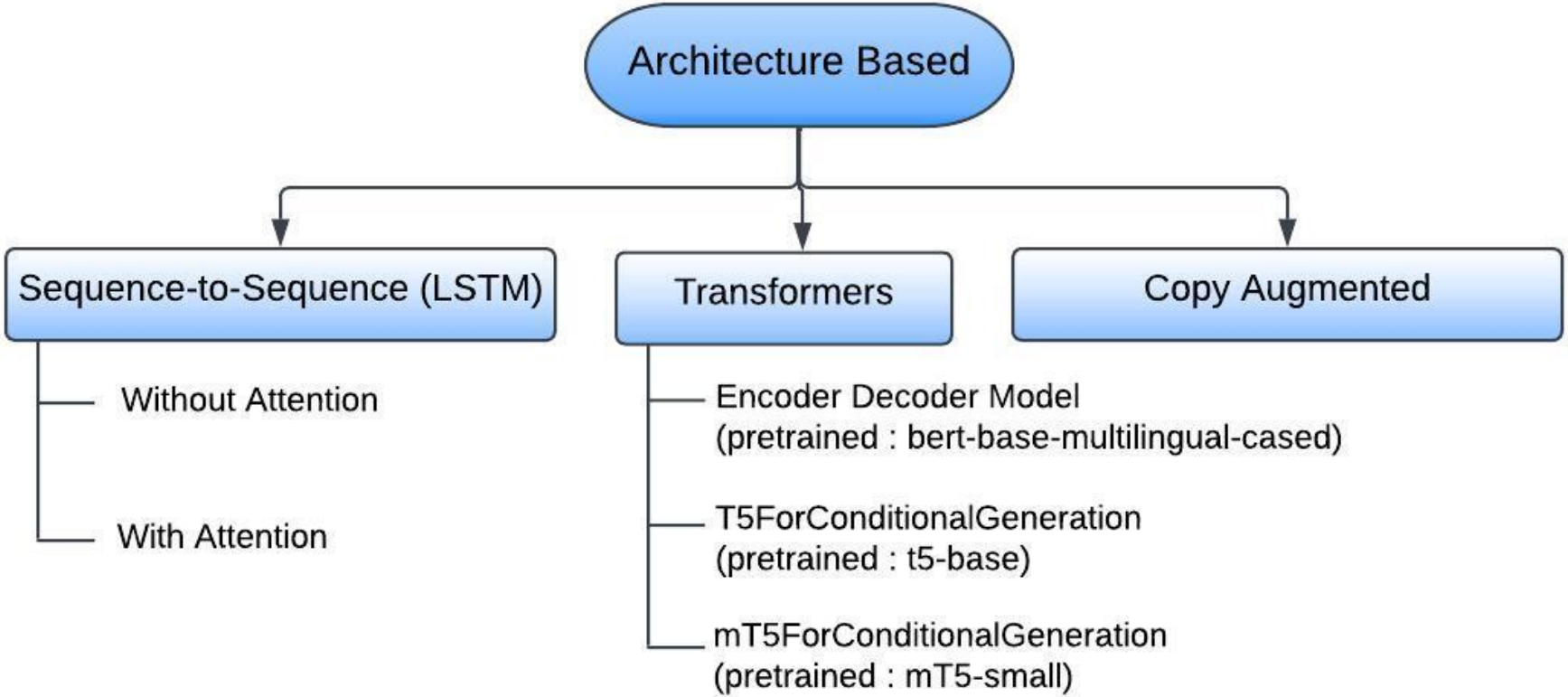
- Non-Contextual Word Embeddings does not capture the sense or context of a word used in a sentence.

- Contextual Word Embeddings are dynamic embeddings and consider the context in which a word has been used in a sentence.

For Example :

Sentence 1: मेरा **खाता** बैंक में है।

Sentence 2: राम आम **खाता** है।


Architecture Based
Sequence-to-Sequence (LSTM)
Transformers
Copy Augmented
Without Attention
With Attention
Encoder Decoder Model
(pretrained : bert-base-multilingual-cased)
T5ForConditionalGeneration
(pretrained : t5-base)
mT5ForConditionalGeneration
(pretrained : mT5-small)

Sequence-to-Sequence Model

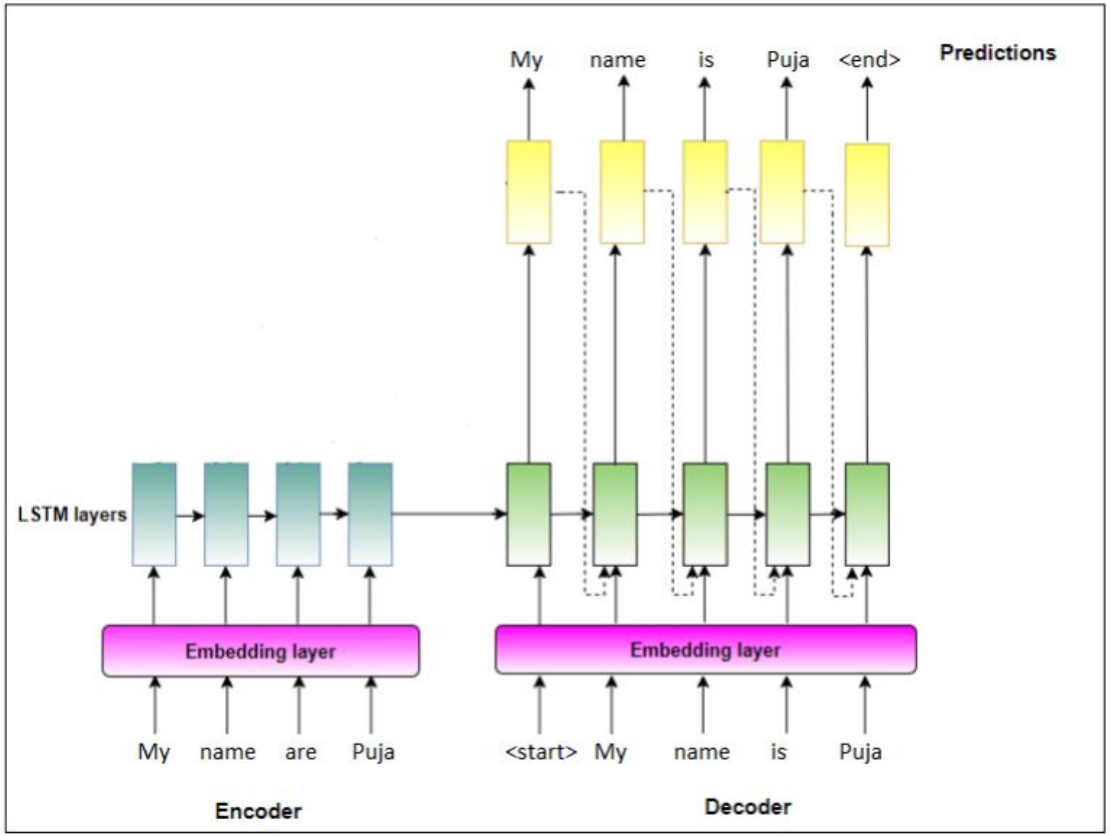
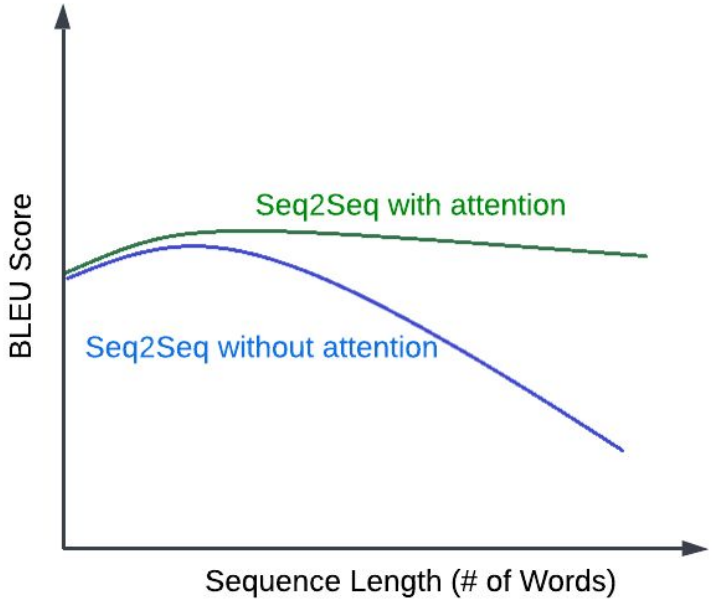


Figure 2.2: Seq2Seq Encoder Decoder model without Attention

Shortcomings of Seq-to-Seq Model (without attention)

- Compress the information about the i/p sentence into a fixed length vector “Embedding”.

- The longer the input sentence the greater the information loss leading to information bottleneck.



Sequence-to-Sequence (With Attention)

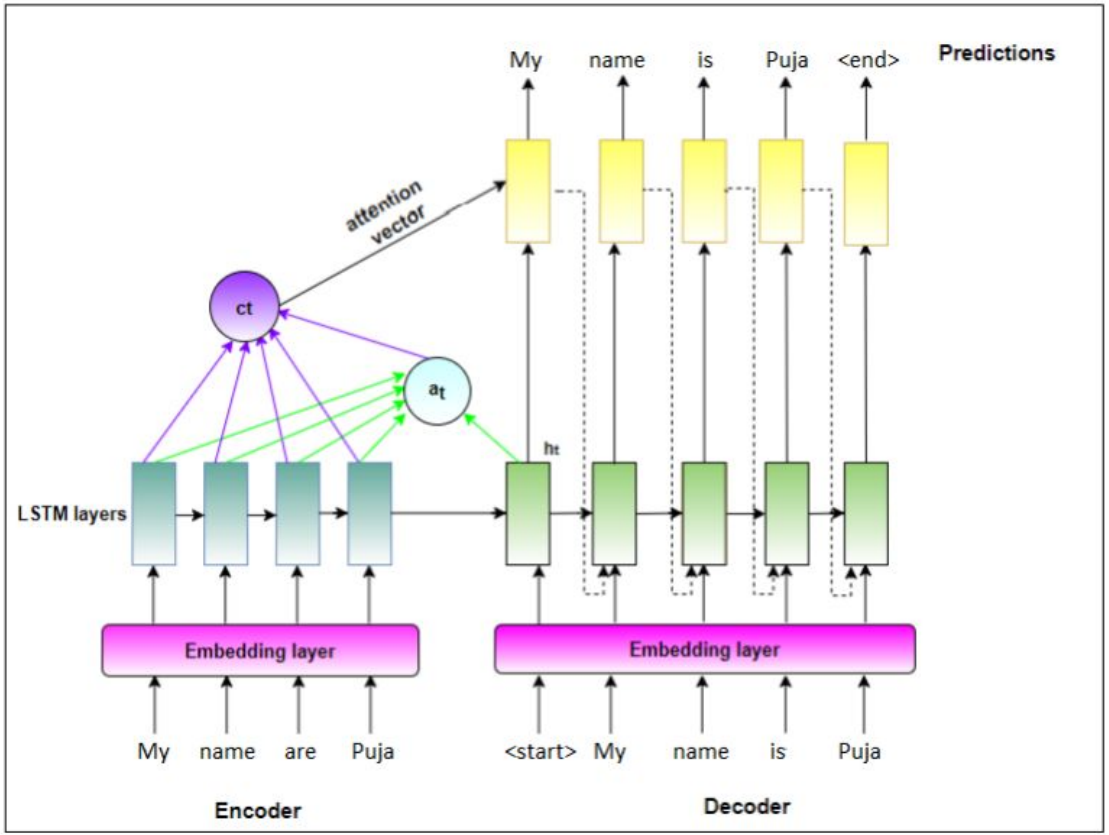
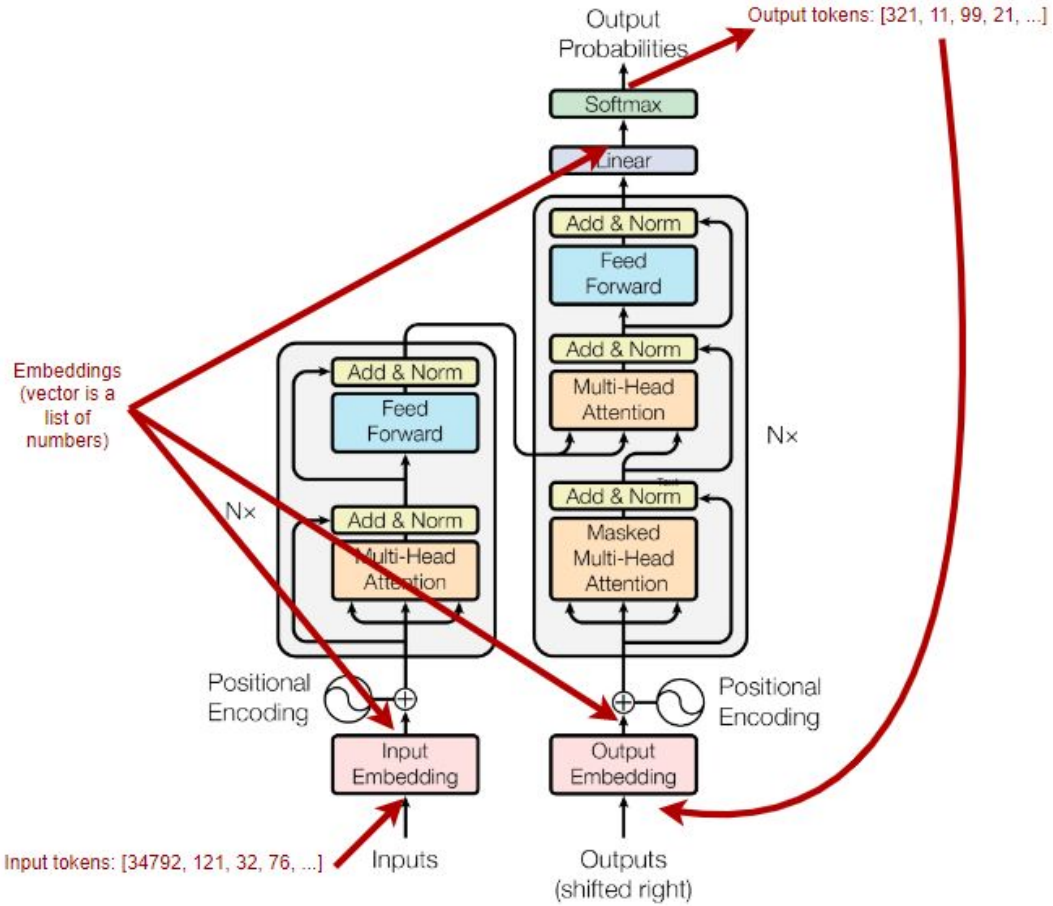


Figure 2.3: Seq2Seq Encoder Decoder model with Attention mechanism for GEC in English

Benefits of Attention

- Attention significantly improves Encoder Decoder based Seq2Seq model.
- With Attention, Seq2Seq model does not forget the source input.
- Attention mechanism keeps the context of input sentences to encoder with importance of a particular word.
- Downside: much more computation.

Transformer Model



Transformer Models

Difference in Configuration of used Transformer-based Models

Models	EncoderDecoder Model	T5ForConditionalGeneration	mT5ForConditionalGeneration
Pretrained Model Used	*bert-base-multilingual-cased*	t5-base	mT5-small
Pretrained task	Self supervised pre training (Masked lang Modelling & Next Sentence Predict)	Multitask mixture of Unsupervised & Supervised	Pretrained on 101 languages excluding any supervised training
Number of Encoder layers	12	12	8
Number of Decoder layers	12	12	8
Number of Attention layers	12	12	6
Hidden Units	768	768	-
Embedding Dimension	768	768	512
Vocab Size	119547	197285	250112
Parameters	179M	220M	300M
Tokenizers	bert-base-multilingual-cased	MuRIL,RoBERTa	MuRIL
Word Embedding	Contextual	Contextual	Contextual

Copy Augmented Transformer Architecture

- Base Architecture

  – FairSeqModel by Facebook AI Research

- Copying Mechanism

  Copy words directly from the source sentence.

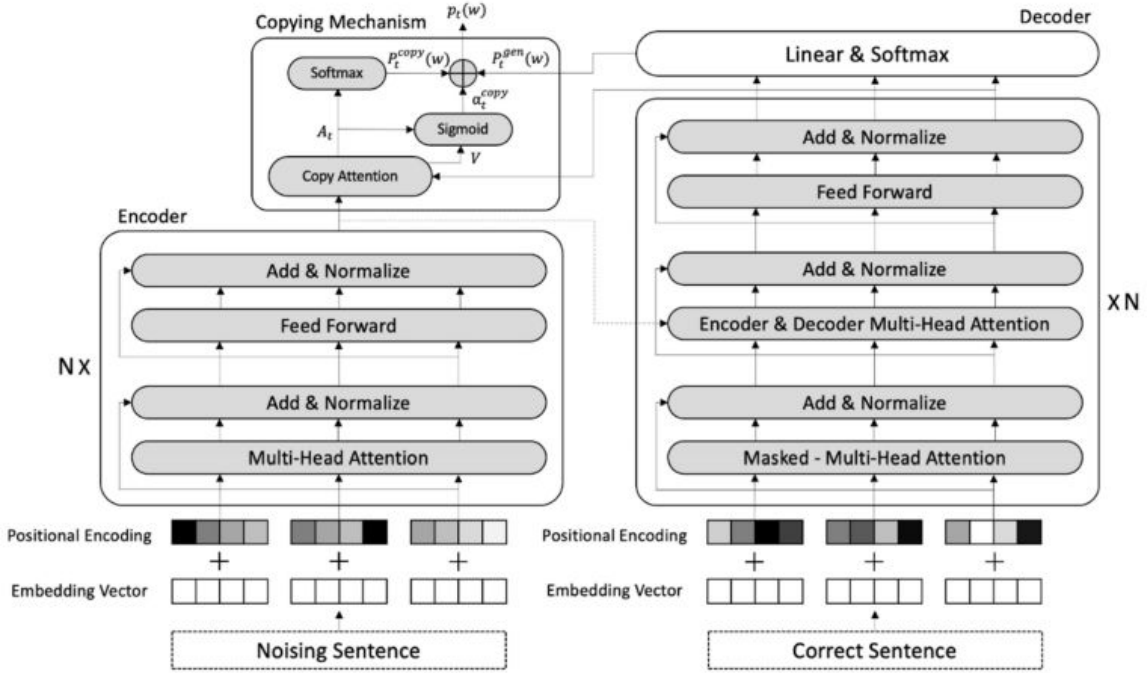


Figure 3. Model Architecture.

Copy Mechanism

$$p_t(w) = (1-\alpha_t^{copy}) * p_t^{gen}(w) + (\alpha_t^{copy}) * p_t^{copy}(w)$$

P_t : Final Probability distribution

P_t^{gen} : Generation Distribution

P_t^{copy}. : Copy Distribution

$\alpha_t^{copy} \in [0, 1]$: Balancing factor (Controls the balance between copying and generating at each time stamp t.

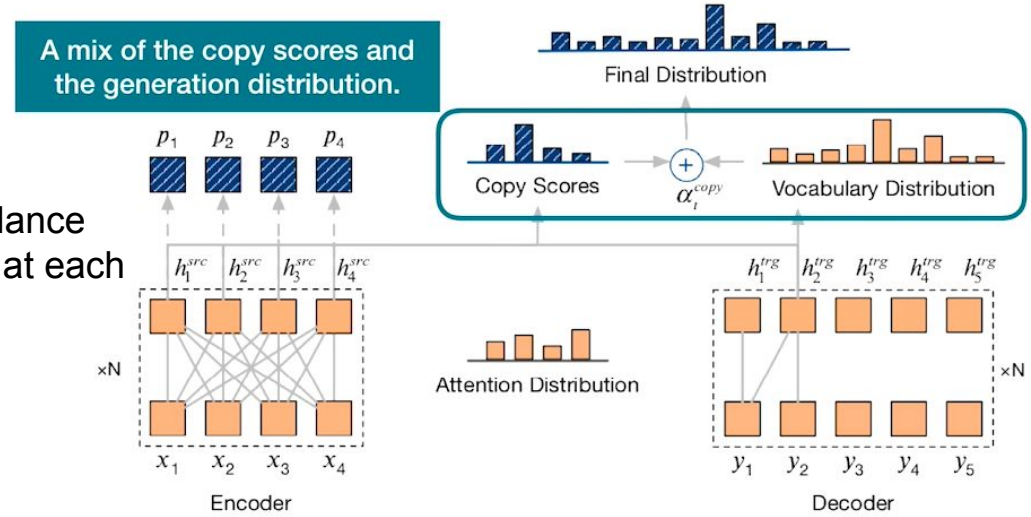


Figure 1: Copy-Augmented Architecture.

Input Sentence : ISRO are launching satellites.

Output Vocab: {are , launching , satellites, ….}

O/p with No copying mechanism : UNK is launching satellites.

Copy Augmentation : {are , launching , satellites, ….} + {ISRO , …}

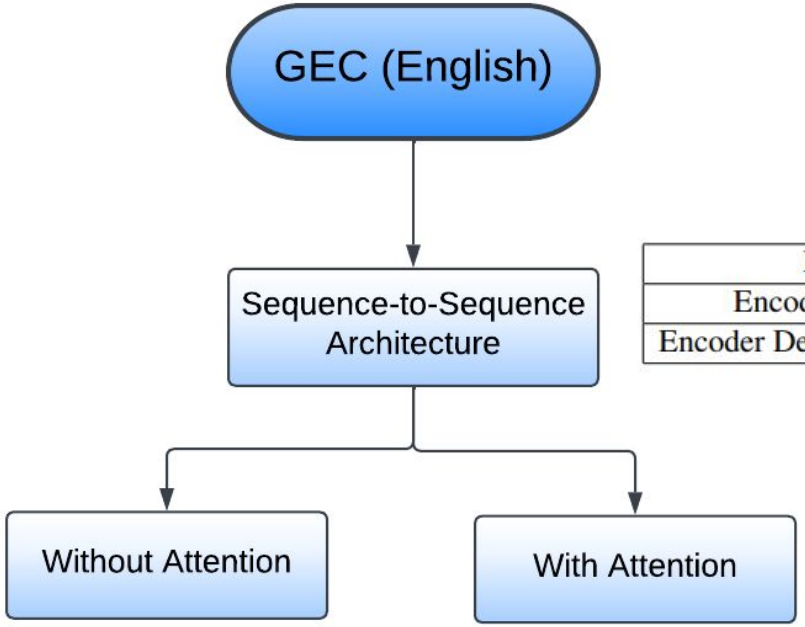
O/p with copy mechanism : ISRO **is** launching satellites.

Experiments on English Language

- Seq2Seq Encoder Decoder
  - without attention
  - with attention

- T5 transformer model

Results on English Language



LANG8 dataset
(80624 training, 20156 validation, 2000 test sentences)

Model	Embedding	Training Loss	Validation Loss	BLEU Score
Encoder Decoder	Trainable	0.5163	1.1652	0.1294
Encoder Decoder (Attention)	Trainable	0.2610	0.6519	0.4426

Table 5.1: Test results obtained for different models for English

Results on English Language

Model	Pre-trained	Tokenizer	Epoch	Training Loss	Validation Loss
T5ForConditionalGeneration	t5-base	t5-base	1	0.78800	0.655472

Table : Test results obtained for T5 transformer models for English
(45000 training sentences & 5000 validation sentences from the C4 dataset)

```
text = 'You is a kind person.'
print("Incorrect Sentence:", text)
predicted_s = correct_grammar(text, num_return_sequences=4)[1]
print("Predicted Sentence:",predicted_s)
print("_____")
```

```
Incorrect Sentence: Cat drinked milk.
Predicted Sentence: Cat drank milk.

_____
Incorrect Sentence: I doing my M.Tech Project.
Predicted Sentence: I'm doing my M.Tech Project.

_____
Incorrect Sentence: You is a kind person.
Predicted Sentence: You're a kind person.
```

Figure: GEC using T5 model for English

Proposed Method for Hindi GEC

Seq2Seq model with attention produced a decent result for English. These results motivated us to implement the same method for GEC in Hindi as well.

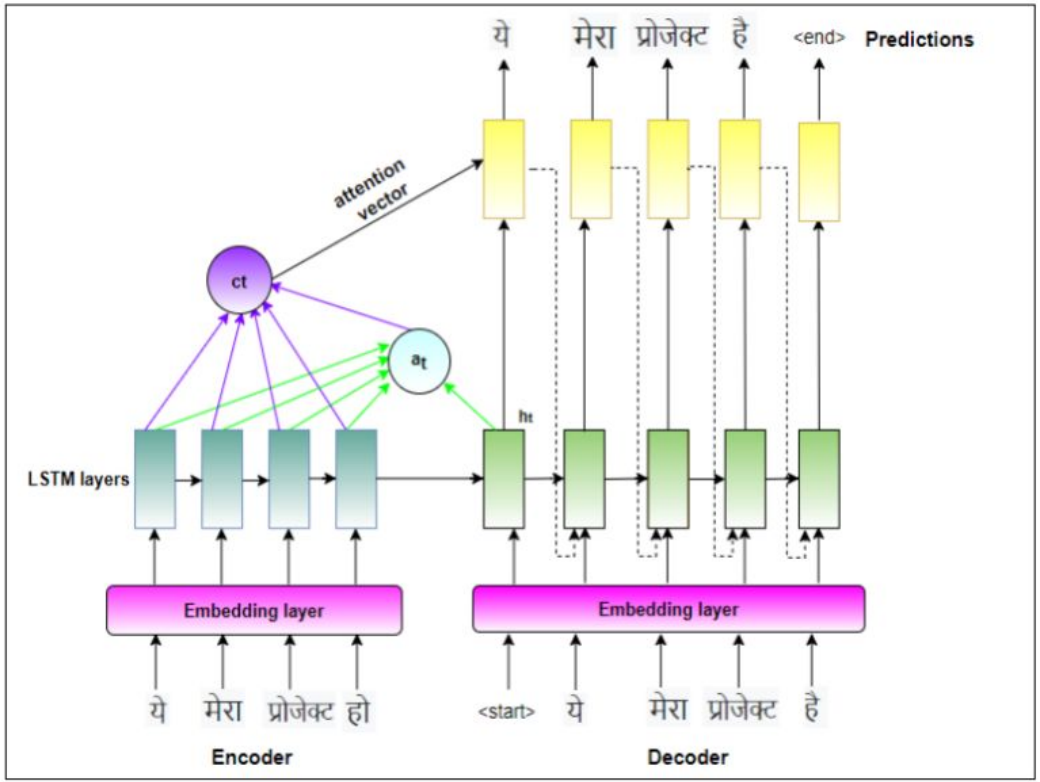


Figure 4.1: Seq2Seq Encoder Decoder model with Attention for GEC in Hindi

Proposed method for Hindi

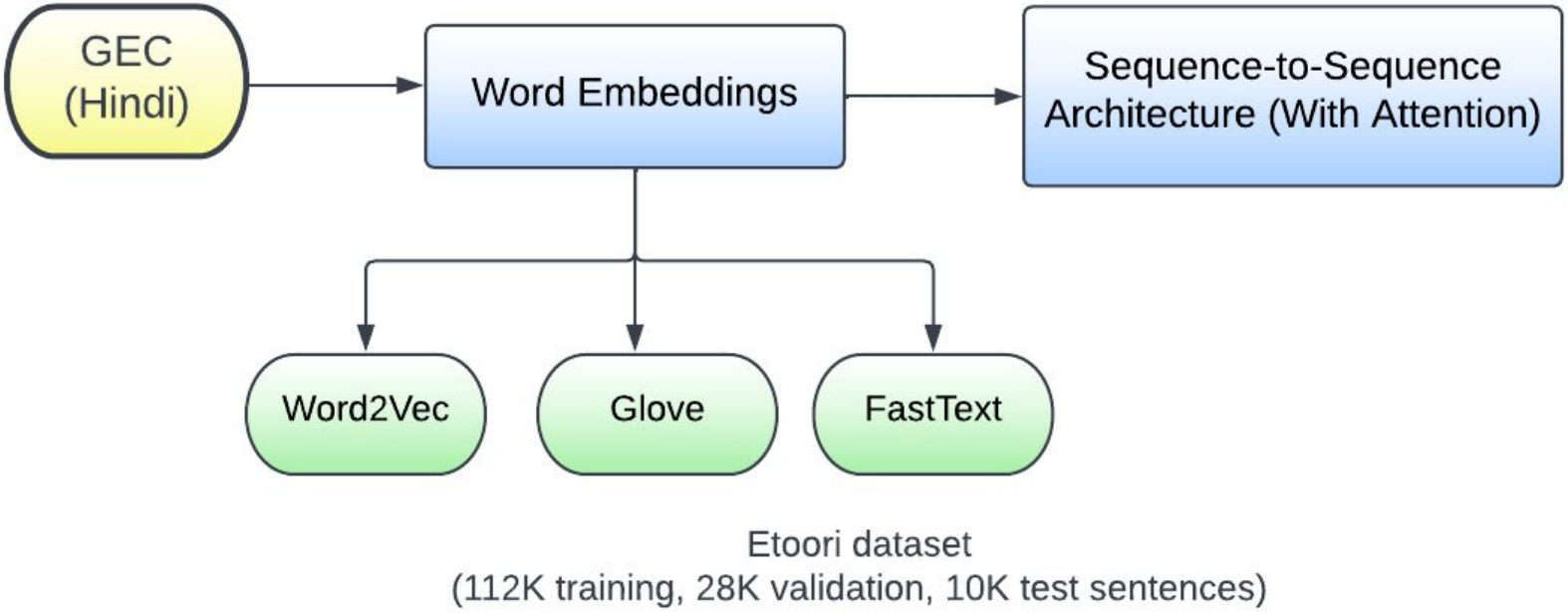


Table : GEC using Sequence-to-Sequence model with attention

Proposed method for Hindi

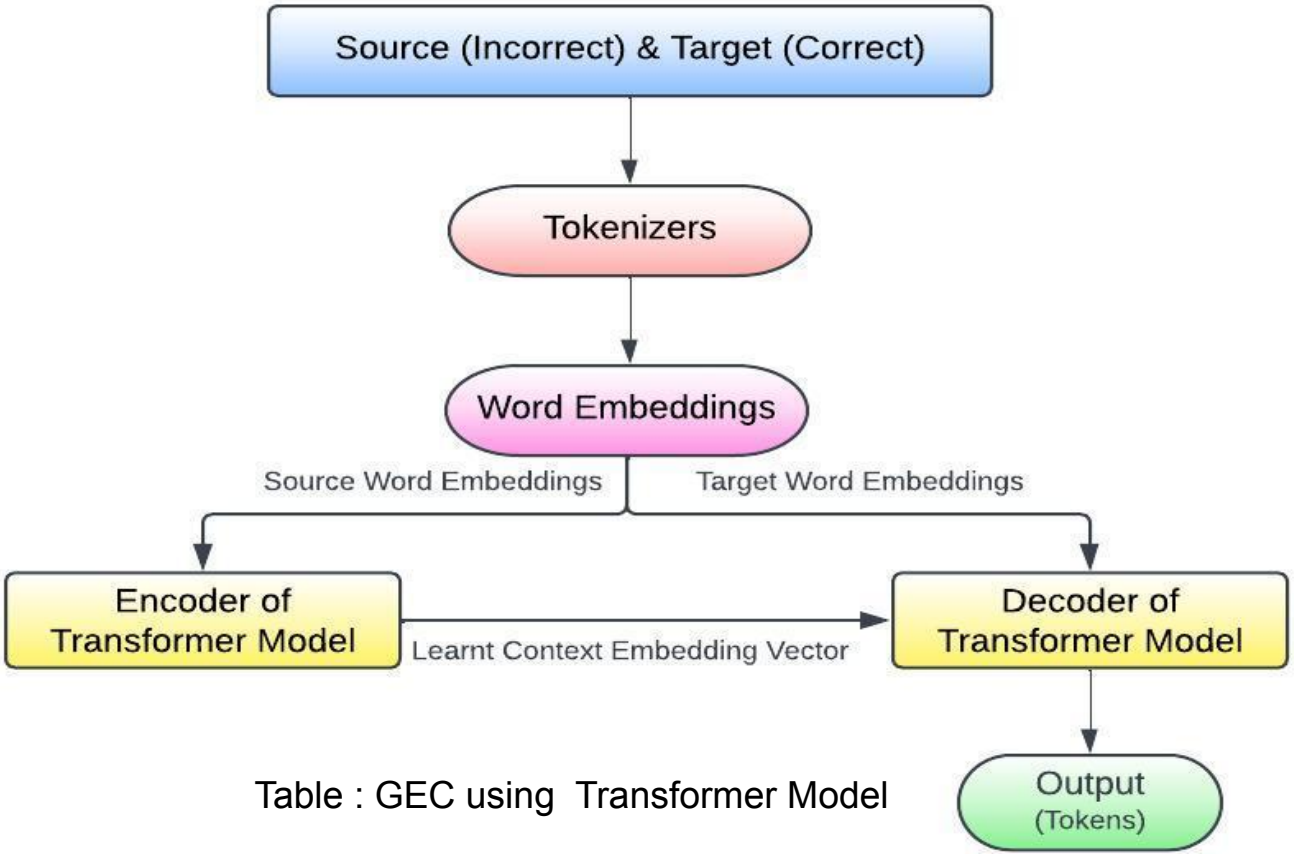


Table : GEC using Transformer Model

DataSet Used

Dataset	Training Sentences	Validation Sentences	Test Sentences
Etoori	112K	28K	10K
HiWikEd	2.6M		13K

enc_input	dec_input
परन्तु वे दोनों उन बातों को ज़्यादा समय तक अप ...	परन्तु वे दोनों उन बातों को ज़्यादा समय तक अपन...
देश में हिन्दी को विस्थापित कर का षड़यंत्र चल ...	देश में हिन्दी को विस्थापित करने का षड़यंत्र च...
तीन साल पहले कातिलाना हमले के प्रकरण में एफआर ...	तीन साल पहले कातिलाना हमले के प्रकरण में एफआर ...
रामायण रिविजिटेड अ टेल ऑफ लव एंड एडवेंचर नाम स...	रामायण रिविजिटेड अ टेल ऑफ लव एंड एडवेंचर नाम स...
तब तक के लिए हमें विराम ले की अनुमति दीजिए ।	तब तक के लिए हमें विराम लेने की अनुमति दीजिए ।

Models used for Hindi

Models	Seq-to-Seq (attention)	EncoderDecoder Model	T5ForConditionalGeneration	mT5ForConditionalGeneration	Copy Augmentation
Pretrained Model Used	-	*bert-base-multilingual-cased*	t5-base	mT5-small	-
Pretrained task	-	Self supervised pre training (Masked lang Modelling & Next Sentence Predict)	Multitask mixture of Unsupervised & Supervised*	Pretrained on 101 languages excluding any supervised training	-
Number of Encoder layers	1	12	12	8	6
Number of Decoder layers	1	12	12	8	6
Number of Attention layers	1	12	12	6	16
Copy Attention	-	-	-	-	1
Hidden Units	32	768	768	-	-
Embedding Dimension	300	768	768	512	512
Vocab Size	73829	119547	197285	250112	73829
Parameters	-	179M	220M	300M	-
Tokenizers	indic_tokenize	bert-base-multilingual-cased	MuRIL,RoBERTa	MuRIL	Tokenizes based on white spaces
Word Embedding	Non Contextual	Contextual	Contextual	Contextual	Contextual

Hyperparameters used to Train the model:

Models	Seq-to-Seq (attention)	EncoderDecoder Model	T5ForConditionalGeneration	mT5ForConditionalGeneration	Copy Augmentation
Epochs	50 with early stop	2	10	5	10
Loss	Sparse Cross Entropy	Cross Entropy	Cross Entropy	Cross Entropy	Cross Entropy
Optimizer	Adam	AdamW	AdamW	AdamW	Adam
Learning rate	1e-3	2e-5	2e-5	2e-5	1e-3
Batch Size	32	8	8	1	32
Weight decay	-	0.1	0.1	0.1	-
Dropout	-	0.1	0.1	0.1	0.2
fp16	False	True	True	False	False
Evaluation metric	Accuracy, BLEU Score, GLEU Score	Train/Valid Loss, GLEU Score	Train/Valid Loss, GLEU Score	Train/Valid Loss, GLEU Score	Train Loss, GLEU Score
cuDNN version	8.1.1	8.1.1	8.1.1	7.6.5	7.6.5
Cuda version	11.1	11.1	11.1	10.0	10.0

Result For GEC Task For Hindi - Seq2Seq

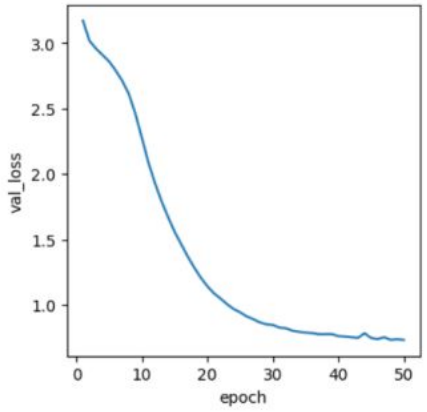
Model (Seq2Seq)	Validation Accuracy (%)	BLEU Score	Model Size (Bytes)	Inference Time(ms)
Encoder Decoder	69.55	0.30497	255761972	177
Encoder Decoder (Attention)	99.02	**0.76631**	331019284	186

Table 2: Encoder Decoder model with attention outperformed Encoder Decoder model without attention

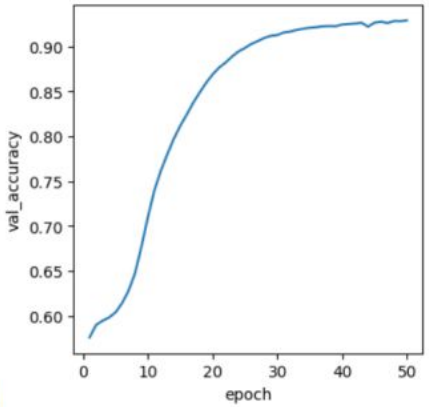
Result For GEC Task For Hindi - Seq2Seq

Model (Seq2Seq)	Embedding	Training Accuracy (%)	Validation Accuracy (%)	BLEU Score (Greedy Search)	GLEU Score
Encoder Decoder (Attention)	Word2Vec	98.08	95.44	0.7796	0.8050
Encoder Decoder (Attention)	Glove	98.86	96.15	0.82002	0.7821
Encoder Decoder (Attention)	FastText	99.30	96.55	0.8458	**0.8639**

Table 3: Results obtained for different word embeddings for Sequence-to-Sequence model with attention

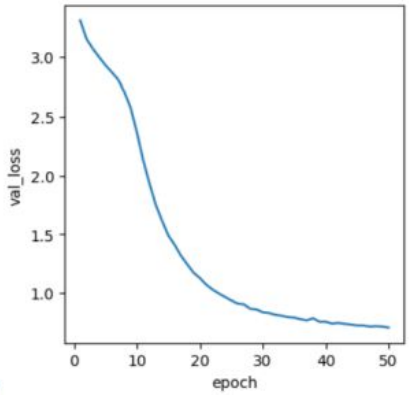


[a]

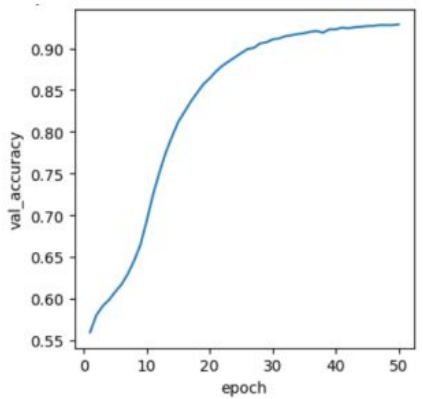


[b]

Figure 5.5: [a] Validation Loss Vs Epoch and [b] Validation Accuracy Vs Epoch for Encoder Decoder Architecture with Attention using Word2Vec Word Embedding

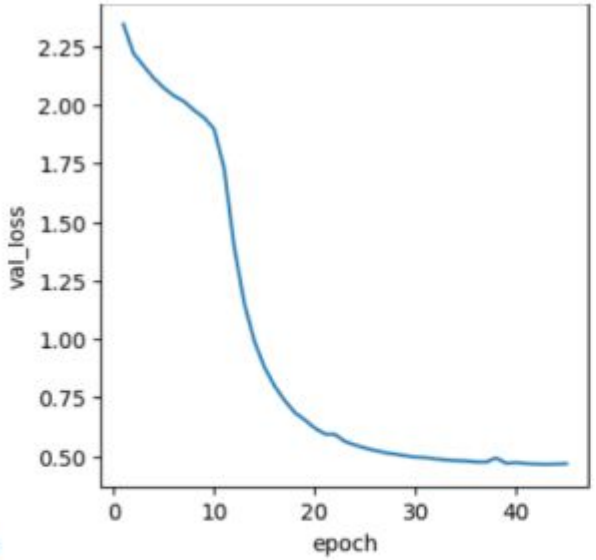


[a]

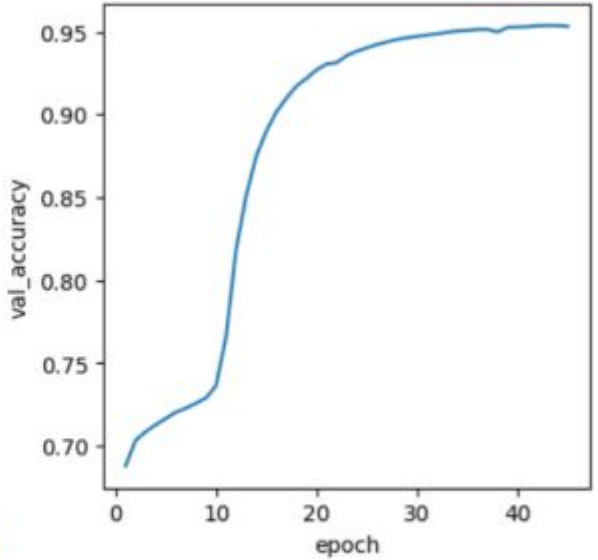


[b]

Figure 5.6: [a] Validation Loss Vs Epoch and [b] Validation Accuracy Vs Epoch for Encoder Decoder Architecture with Attention using Glove Word Embedding



[a]



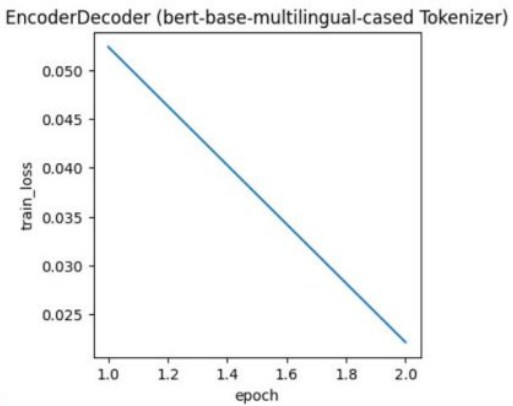
[b]

Figure 5.7: [a] Validation Loss Vs Epoch and [b] Validation Accuracy Vs Epoch for Encoder Decoder Architecture with Attention using Fasttext Word Embedding

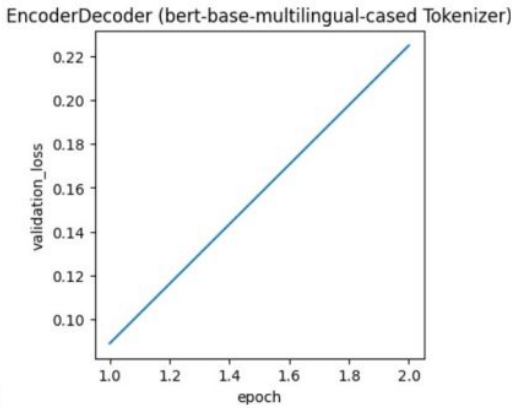
Result For GEC Task For Hindi - Transformers

Model (Transformers)	Pretrained Model	Tokenizer	Training Loss	Validation Loss	Vocabulary Size	GLEU Score
Encoder Decoder	bert-base-multilingual-cased	bert-base-multilingual-cased	0.32566	0.2248	119547	0.91757
T5ForConditional Generation	t5-base	RoBERTa-hindi-guj-san	0.08328	0.06815	30522	0.82015
T5ForConditional Generation	t5-base	MuRIL	0.43034	0.16969	197285	**0.92086**
mT5ForConditionalGeneration	mt5-small	MuRIL	0.11250	0.09533	250112	0.90232
Copy Augmented	-	White spaces	0.37821	0.24620	73829	0.8574

Table 3: Results obtained for different Transformer based models

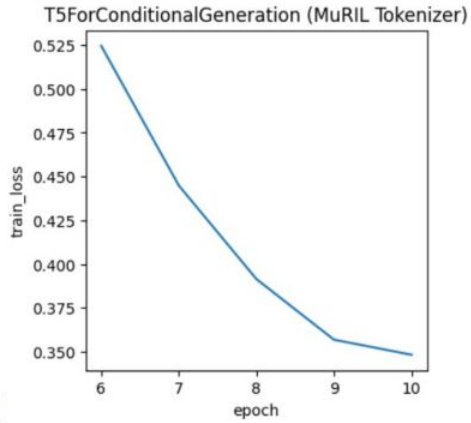


[a]

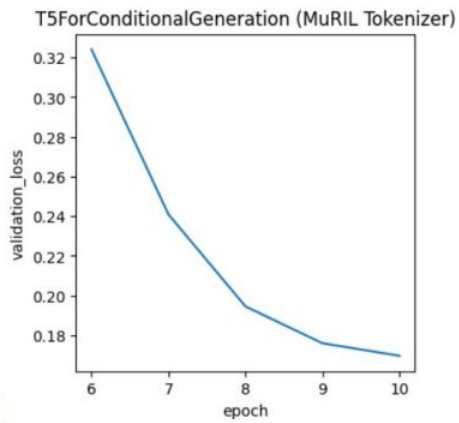


[b]

Figure 6.12: [a] Train Loss Vs Epoch and [b] Validation Loss Vs Epoch for Encoder-Decoder (pre-trained: bert-base-multilingual-cased each) transformer based Architecture with bert-base-multilingual-cased Tokenizer

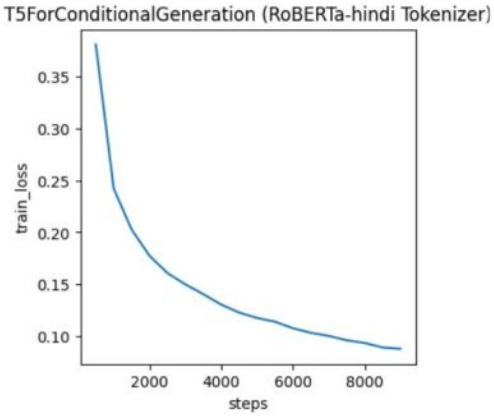


[a]

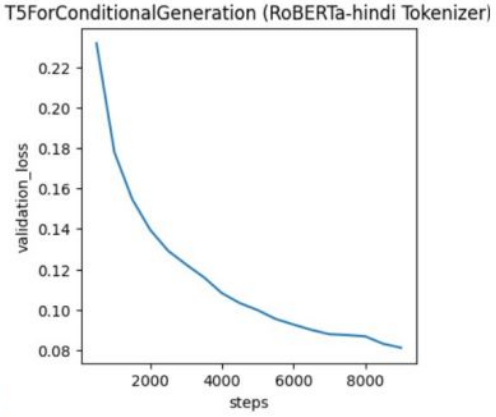


[b]

Figure 6.13: [a] Train Loss Vs Epoch and [b] Validation Loss Vs Epoch for T5ForConditionalGeneration (pre-trained: t5-base) transformer based Architecture with MuRIL Tokenizer

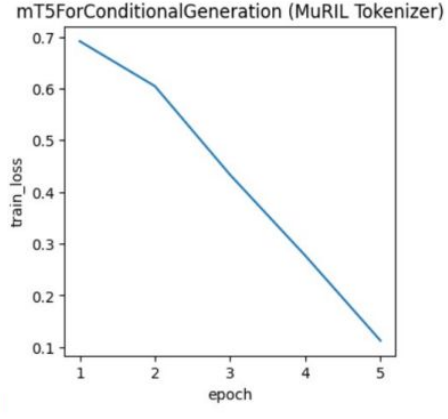


[a]

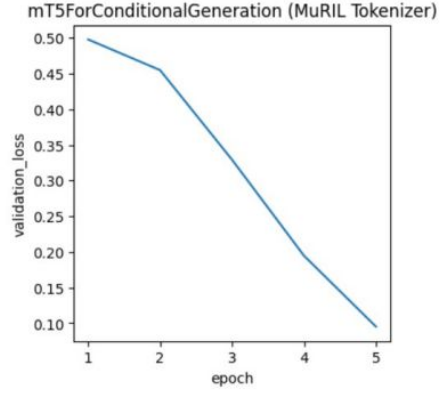


[b]

Figure 6.14: [a] Train Loss Vs Steps and [b] Validation Loss Vs Steps for T5ForConditionalGeneration (pre-trained: t5-base) transformer based Architecture with RoBERTa-hindi Tokenizer



[a]



[b]

Figure 6.15: [a] Train Loss Vs Epoch and [b] Validation Loss Vs Epoch for mT5ForConditionalGeneration (pre-trained: mt5-small) transformer based Architecture with MuRIL Tokenizer

Models Comparison

Model	Architecture	Embedding	Tokenizer	Model Size (MB)	Model Parameters	Inference Time (s)
Encoder Decoder (Attention)	Seq2Seq	Word2Vec	indic-tokenizer	195.5136	48870508	0.273
Encoder Decoder (Attention)	Seq2Seq	Glove	indic-tokenizer	195.5136	48870508	0.253
Encoder Decoder (Attention)	Seq2Seq	FastText	indic-tokenizer	195.5136	48870508	0.345
Encoder Decoder	Transformer	Contextual	bert-base-multilingual-cased	1465.603	384194811	1.16
T5ForConditionalGeneration	Transformer	Contextual	RoBERTa-hindi-guj-san	845.604	221670144	1.38
T5ForConditionalGeneration	Transformer	Contextual	MuRIL	1145.083	300176768	1.02
mT5ForConditionalGeneration	Transformer	Contextual	MuRIL	1334.168	349744128	1.47
Copy Augmented	Copy Augmentation	Contextual	White spaces	1116.105	293570124	1.23

Table 3 : Performance Analysis of different models based on Model Size, Model Parameters and Inference Time

Models Comparison

Model	GLEU Score (Etoori Test Dataset)	GLEU Score (HiWikEd test dataset)
Transf@	-	0.69
MLConv@	-	0.73
Copy Augmented*@	0.8574	0.80
Seq2Seq + Attention (Glove)*	0.78218	-
Seq2Seq + Attention (Word2Vec)*	0.80502	-
Seq2Seq + Attention (fastText)*	0.86394	-
Bert based Encoder Decoder (Transformer)*	0.91757	-
mT5ForConditionalGeneration (Transformer)*	0.90232	
Transformer(T5ForConditionalGeneration)* (Trained on 140K Etoori data + 2.6 M Artificially generated Hindi Test dataset)	0.92086	0.8107

Table 3 : Performance Analysis of different models based on GLEU Score.
* : Models implemented by me ;@ : Models implemented by Ankur et al.

Models	Epoch	ADP:INFL	PRON:INFL	ADJ:INFL	VERB:INFL	Full Dataset
Tranf[@]	1	0.62	0.67	0.57	0.79	0.69
Copy Augmented[@]	9	0.84	0.71	0.69	0.87	0.80
T5ForConditionalGeneration[*]	1	0.8082	**0.8095**	**0.7434**	0.8298	**0.8107**

[@] : Performance of Models Implemented by Ankur et al. on HiWikEd test dataset
[*] : Model Implemented by me (which is giving best performance among other trained model)

The T5ForConditionalGeneration model showed:

- Better results than baseline Tranf and Copy Augmented model for PRON:INFL and ADJ:INFL errors.
- Better results than baseline Tranf model for all types of errors and on Full Dataset.
- Better results than baseline Tranf and Copy Augmented model for full dataset.

GEC - Trained Models

Incorrect Sentence **Correct Sentence**	घर में **घुस** के बाद भी सुकून नहीं । घर में **घुसने** के बाद भी सुकून नहीं ।
Model	**Predicted Sentence**
Seq2Seq_attention (FastText)	घर में **घुसने** के बाद भी सुकून नहीं ।
encoder_decoder_bert-multilingual	घर में **घुसने** के बाद भी सुकून नहीं ।
T5_RoBERTa	घर में **घुसने** के बाद भी सुकून नहीं ।
T5_muRIL	घर में **घुसने** के बाद भी सुकून नहीं ।
mT5_small	घर में **घुसने** के बाद भी सुकून नहीं ।
Incorrect Sentence **Correct Sentence**	**मैंने** निर्णायक बन कर नहीं बैठता हूँ । **मैं** निर्णायक बन कर नहीं बैठता हूँ ।
Model	**Predicted Sentence**
Seq2Seq_attention (FastText)	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
encoder_decoder_bert-multilingual	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
T5_RoBERTa	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
T5_muRIL	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
mT5_small	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।

GEC - Trained Models

Incorrect Sentence **Correct Sentence**	**उसनें** बिपाशा ने यह स्वीकार किया है कि उन्हें कभी करीना को समझने का मौका ही नहीं मिला। **वहीं** बिपाशा ने यह स्वीकार किया है कि उन्हें कभी करीना को समझने का मौका ही नहीं मिला ।
Model	**Predicted Sentence**
Seq2Seq_attention (FastText)	**वहीं** बिपाशा ने यह स्वीकार किया है कि उन्हें कभी करीना को समझने का मौका ही नहीं मिला ।
encoder_decoder_bert-multilingual	**वहीं** बिपाशा ने यह स्वीकार किया है कि उन्हें कभी करीना को समझने का मौका ही नहीं मिला ।
T5_RoBERTa	**वहीं** बिपाशा ने यह स्वीकार किया है कि उन्हें कभी करीना को समझने का मौका ही नहीं मिला ।
T5_muRIL	**वहीं** बिपाशा ने यह स्वीकार किया है कि उन्हें कभी करीना को समझने का मौका ही नहीं मिला ।
mT5_small	**वहीं** बिपाशा ने यह स्वीकार किया है कि उन्हें कभी करीना को समझने का मौका ही नहीं मिला ।
Incorrect Sentence **Correct Sentence**	सीआरपीएफ **अप** ट्रेनिंग में लड़कियों को आत्मरक्षा की पूरी जानकारी दी है । सीआरपीएफ **ने अपने** ट्रेनिंग में लड़कियों को आत्मरक्षा की पूरी जानकारी दी है ।
Model	**Predicted Sentence**
Seq2Seq_attention (FastText)	आरपीएफ **ने अपने** ट्रेनिंग में लड़कियों को आत्मरक्षा की पूरी जानकारी दी है ।
encoder_decoder_bert-multilingual	आरपीएफ **ने अपने** ट्रेनिंग में लड़कियों को आत्मरक्षा की पूरी जानकारी दी है ।
T5_RoBERTa	आरपीएफ **ने अपने** ट्रेनिंग में लड़कियों को आत्मरक्षा की पूरी जानकारी दी है ।
T5_muRIL	आरपीएफ **ने अपने** ट्रेनिंग में लड़कियों को आत्मरक्षा की पूरी जानकारी दी है ।
mT5_small	आरपीएफ **ने अपने** ट्रेनिंग में लड़कियों को आत्मरक्षा की पूरी जानकारी दी है ।

GEC - Trained Models

Incorrect Sentence **Correct Sentence**	**मैंने** निर्णायक बन कर नहीं बैठता हूँ । **मैं** निर्णायक बन कर नहीं बैठता हूँ ।
Model	**Predicted Sentence**
Seq2Seq_attention (FastText)	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
encoder_decoder_bert-multilingual	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
T5_RoBERTa	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
T5_muRIL	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
mT5_small	**मैं** निर्णायक बन कर नहीं बैठता हूँ ।
Incorrect Sentence **Correct Sentence**	शहरी व ग्रामीण क्षेत्र में सीवरेज सिस्टम लागू कर का मुख्य उद्देश्य वातावरण को दूषित **हो** से बचाना है । शहरी व ग्रामीण क्षेत्र में सीवरेज सिस्टम लागू करने का मुख्य उद्देश्य वातावरण को दूषित **होने** से बचाना है ।
Model	**Predicted Sentence**
Seq2Seq_attention (FastText)	शहरी व ग्रामीण क्षेत्र में सीवरेज सिस्टम लागू करने का मुख्य उद्देश्य वातावरण को दूषित **होने** से बचाना है ।
encoder_decoder_bert-multilingual	शहरी व ग्रामीण क्षेत्र में सीवरेज सिस्टम लागू करने का मुख्य उद्देश्य वातावरण को दूषित **होने** से बचाना है ।
T5_RoBERTa	शहरी व ग्रामीण क्षेत्र में सीवरेज सिस्टम लागू करने का मुख्य उद्देश्य वातावरण को दूषित
T5_muRIL	शहरी व ग्रामीण क्षेत्र में सीवरेज सिस्टम लागू करने का मुख्य उद्देश्य वातावरण को दूषित **होने** से बचाना है ।
mT5_small	शहरी व ग्रामीण क्षेत्र में सीवरेज सिस्टम लागू करने का मुख्य उद्देश्य वातावरण को दूषित **होने** से बचाना है ।

Conclusion

- Detailed comparison of model performance using various architectures, tokenizers, and word embeddings.
- Sequence-to-Sequence model with attention out performed the one without attention with a huge margin.
- Among Seq2Seq model with non-contextual word embedding, best results obtained using FastText.
- All variations of the transformer model gave better results than the Seq2Seq model, with T5ForConditionalGeneration (using MuRIL Tokenizer), giving the best results.
- Choice of tokenizer and word embedding can significantly impact the model's performance.
- Size and number of parameters of transformer-based models are almost 6-7 times more than the Seq2Seq.
- Seq2Seq-based architecture's Inference time is much better than transformer-based architecture's.

Future Work

- A more inclusive dataset with real time scenarios and more than one error per sentence can be created.
- A simple and efficient sequence tagging approach as proposed by Omelianchuk et. al. in GECToR (https://aclanthology.org/2020.bea-1.16.pdf) can be used for Hindi as well.
- The process can be generalized for other low-resource Indian languages for GEC tasks and made open source.

Thank You

Reference Slides

Encoder Decoder Transformer

- Encoder part : *pretrained bert-base-multilingual-cased*
- Decoder part : *pretrained bert-base-multilingual-cased*

The model configuration:

- Number of layers: 12
- Hidden size: 768
- Number of attention heads: 12
- Vocabulary size: 119,547
- Maximum sequence length: 512

```
loading configuration file config.json from cache at /DATA/gupta92/.
5e83dbed325647a63e7e1f5de19f0382ba/config.json
Model config BertConfig {
  "_name_or_path": "bert-base-multilingual-cased",
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "position_embedding_type": "absolute",
  "transformers_version": "4.22.2",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 119547
}
```

T5ForConditionalGeneration

Initialization: ***pretrained t5-base***

The configuration of t5-base model:

- Number of encoder layers: 12
- Number of decoder layers: 12
- Embedding dimension: 768
- Hidden states: 768
- Number of attention heads: 12
- Vocab size :197285

```
loading configuration file t5_gec_hindi_muRIL_2/config.json
Model config T5Config {
  "_name_or_path": "t5-base",
  "architectures": [
    "T5ForConditionalGeneration"
  ],
  "d_ff": 3072,
  "d_kv": 64,
  "d_model": 768,
  "decoder_start_token_id": 0,
  "dense_act_fn": "relu",
  "dropout_rate": 0.1,
  "eos_token_id": 1,
  "feed_forward_proj": "relu",
  "initializer_factor": 1.0,
  "is_encoder_decoder": true,
  "is_gated_act": false,
  "layer_norm_epsilon": 1e-06,
  "model_type": "t5",
  "n_positions": 512,
  "num_decoder_layers": 12,
  "num_heads": 12,
  "num_layers": 12,
  "output_past": true,
  "pad_token_id": 0,
  "relative_attention_max_distance": 128,
  "relative_attention_num_buckets": 32,
```

mT5ForConditionalGeneration

Initialization: ***pretrained mT5-small***

- Multilingual variant of T5 model (101 languages)

The configuration of mT5-small model:

- Number of encoder layers: 8
- Number of decoder layers: 8
- Embedding dimension: 512
- Number of attention heads: 6
- Vocab size : 250112


```
Model config MT5Config {
  "_name_or_path": "mT5_Etoori_1",
  "architectures": [
    "MT5ForConditionalGeneration"
  ],
  "d_ff": 1024,
  "d_kv": 64,
  "d_model": 512,
  "decoder_start_token_id": 0,
  "dense_act_fn": "gelu_new",
  "dropout_rate": 0.1,
  "eos_token_id": 1,
  "feed_forward_proj": "gated-gelu",
  "initializer_factor": 1.0,
  "is_encoder_decoder": true,
  "is_gated_act": true,
  "layer_norm_epsilon": 1e-06,
  "model_type": "mt5",
  "num_decoder_layers": 8,
  "num_heads": 6,
  "num_layers": 8,
  "pad_token_id": 0,
  "relative_attention_max_distance": 128,
  "relative_attention_num_buckets": 32,
  "tie_word_embeddings": false,
  "tokenizer_class": "T5Tokenizer",
  "torch_dtype": "float32",
  "transformers_version": "4.22.2",
  "use_cache": true,
  "vocab_size": 250112
}
```