OPTIMIZING TRANSFORMER MODELS FOR ENGLISH2HINDI TRANSLATION: A SUPERVISED FINE-TUNING ANALYSIS

A Thesis Submitted
In Partial Fulfillment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY
in
Data Science

by
Anmol Chhetri
(2K22/DSC/02)

Under the Supervision of
Mr. Rahul
Assistant Professor, Department of Software Engineering
Delhi Technological University

Department of Software Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

May, 2024

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I Anmol Chhetri hereby certify that the work which is being presented in the thesis entitled "Optimizing Transformer Models for English2Hindi Translation: A Supervised Fine-tuning Analysis" in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Data Science, submitted in the Department of Software Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from 2022 to 2024 under the supervision of Mr. Rahul.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

27/5/24

Signature of Supervisor(s)

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR(s)

Certified that **Anmol Chhetri** (2K22/DSC/02) has carried out his search work presented in this thesis entitled **"Optimizing Transformer Models for English2Hindi Translation: A Supervised Fine-tuning Analysis"** for the award of **Master of Technology** from Department of Software Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

27/5/24

Mr. Rahul
Supervisor
Assistant Professor
Department of Software Engineering
Delhi Technological University

Place: New Delhi
Date: 27/05/24.

Optimizing Transformer Models for English2hindi Translation: A Supervised Fine-Tuning Analysis

Anmol Chhetri

ABSTRACT

Machine Translation is an essential task in natural language processing. It breaks down language barriers, enabling effective communication and collaboration across diverse linguistic and cultural backgrounds. Sequence-to-Sequence models are used for solving various downstream tasks like machine translation, text summarization, Question Answering, Speech recognition, etc. However, machine translation has been a challenging task for researchers. This encouraged researchers to shift from SMT to NMT. NMT is a way of solving a translation task using neural networks like transformers.

Due to its parallel computation capability, it was also used in other applications such as computer vision, audio processing, etc. But researchers stated many challenges with the model such as structural constraints between input and output text, computational complexity, and path length between long-range dependencies. Several other versions of transformers were introduced to address these issues. Modification of transformer architecture can be either in the positional encoding or in the attention mechanism. However, a systematic review with mathematical understanding is still not present.

Transformer-based architectures have been built to perform human-like translation through rigorous training on large corpus data. Hence, Transformers are now considered a benchmark for translation tasks. There are also various pre-trained models that have shown their potential on selective languages, but limited models that solve English to Hindi translation due to the unavailability of large parallel corpus and also due to Hindi language exhibiting complex sentence structures compared to English.

This paper fills this gap by fine-tuning four pre-trained models on the IITB English-Hindi dataset, namely OPUS-MT, M2M100, mBART-50, and MADLAD-400. In this study, the aim is to compare the quality of translated text among these models through a metric called BLEU. It was observed that OPUS-MT and M2M100 produced high-quality Hindi translated text with BLEU of 89.11 and 86.83 respectively. These results were found better as compared to the 44.34 BLEU point of the SOTA model on the IITB dataset. At last, this paper also reviews and analyses two types of X-Formers mainly pre-training and training.

Keywords—*NLP, Neural Machine Translation, Transformer, Fine-tuning, BLEU.*

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

ACKNOWLEDGEMENT

I take the opportunity to express my sincere gratitude to **Mr. Rahul**, Department of Software Engineering, Delhi Technological University, Delhi, for providing valuable guidance and constant encouragement throughout the work. His knowledge and expertise helped me at every stage of the preparation of this research plan.

I am extremely thankful to **Prof. Ruchika Malhotra**, Head of the Department, Department of Software Engineering, Delhi Technological University, Delhi & other faculty members of the Department of Software Engineering, Delhi Technological University, Delhi, for the motivation and inspiration. I am also thankful to all non-teaching staff at DTU, who have helped me directly or indirectly in the completion of this research plan.

Anmol Chhetri
2K22/DSC/02
M.Tech Scholar (Full Time)
Department of Software Engineering
Delhi Technological University

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SMT	Statistical machine translation
NMT	Neural machine translation
WMT	Workshop on Statistical Machine Translation
IITB	Indian Institute of Technology Bombay
CFILT	Computation for Indian Language Technology
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
GPT	Generative Pre-trained Transformer
NLP	Natural language processing
FFNN	Feed-Forward Neural Network
NLP	Natural Language Processing
BLEU	BiLingual Evaluation Understudy
LLM	Large Language Model
SOTA	State-of-the-Art
HLD	High-Level Design
GPU	Graphics Processing Unit
GQA	Group-Query Attention
MQA	Multi-Query Attention
EN-DE	English-to-German
EN-FR	English-to-French

CHAPTER 1

INTRODUCTION

This chapter presents an introduction to transformer architecture for machine translation tasks and describes the history of transformers with the necessity for English-to-Hindi translation. This chapter consists of background, problem statement, motivation, contribution, thesis organization and finally concluded with a summary.

1.1 OVERVIEW

Machine translation has been an exciting task mainly due to the diverse forms of languages that exist all around the globe. Specifically In India, there are 22 registered languages according to the constitution of India with Hindi being the official language [1]. There are currently two techniques that help us in translation i.e., SMT and NMT. In SMT, translations are derived through probability distribution which is generated after processing a large amount of bilingual text [2]. SMT algorithm creates a mapping between the words, phrases and sentences in one language to another language. Additionally, SMT was not able to capture long-range dependency due to which the context of translated text was not clear and produced low accuracy results. NMT algorithms were introduced to resolve these issues by using a neural network in the architecture [3]. The choice of dataset for a particular downstream task is also very critical. Therefore, various translation datasets can be found online via the Hugging Face website. The WMT2014 dataset is one of the most common datasets for translation tasks which includes subsets like the Czech-English (cs-en) dataset, German-English (de-en) dataset, French-English (fr-en) dataset, Hindi-English (hi-en) dataset, and, Russian-English (ru-en) dataset. Due to the large parallel corpus of de-en and fr-en datasets, they are used very frequently for several research studies generally for supervised tasks [4], [5]. However, it was observed that datasets with limited parallel corpus are used for zero-shot translation to create an inductive bias on unseen data [3], [6]. During this research, it was also observed that there is not much research on en-hi tasks. Therefore, this thesis focuses on the IITB CFILT dataset which is one of the first large corpora for en-hi translation tasks containing large parallel English-Hindi and monolingual Hindi corpora.

Traditional methods such as RNN and CNN were used for solving translation tasks. However, both had a complex architecture due to which training time was too long [7], [8]. Therefore, Transformer [9] has rapidly gained popularity in machine translation tasks due to its parallel computing nature and ability to

produce high-quality translated sentences. Transformer architecture was built on a very simple architecture of attention mechanism which focuses on how the tokens in a sequence communicate with each other. Transformer became very popular after the release of ChatGPT [3], [10]. Vanilla Transformer usually follows the architecture of encoder-decoder blocks. Where the first version of Transformer is usually referred to as vanilla. This architecture provides flexibility for solving various NLP tasks other than translation like text summarization [11], [12]. Therefore, transformers based are now considered as a baseline for translation tasks. Hence, the below section provides a brief mathematical understanding of Transformer architecture.

1.2 BACKGROUND

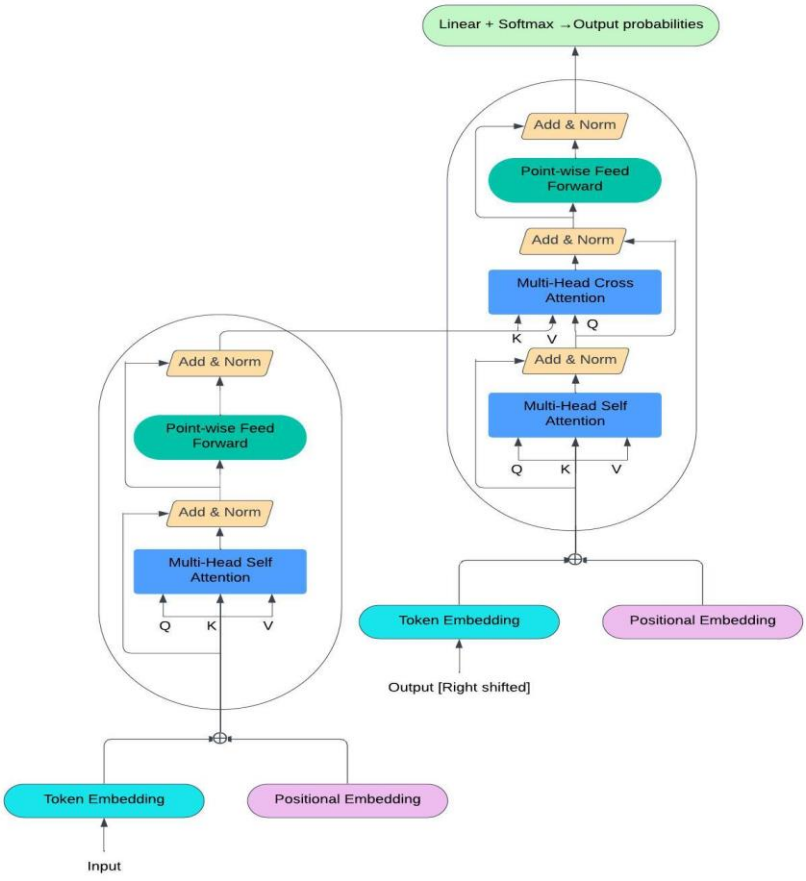


Fig. 1.1 Transformer Architecture

This section discusses all the components of transformer architecture as seen in Fig. 1.1. The vanilla transformer was applied for the WMT2014 EN-DE task and the WMT2014 EN-FR task [9]. Transformer consists of two blocks i.e., encoder and decoder. The encoder helps us to find the contextual meaning hidden within the sequence, whereas the decoder performs autoregressive generation using the teacher

forcing method. An encoder consists of a multi-head self-attention and point-wise FFNN shown in the left block of Fig. 1.1. In contrast, the right block is a decoder that consists of 3 sub-layers mainly multi-head self-attention, followed by cross attention from an encoder and a FFNN. A residual connection is also established around each layer within the encoder and decoder followed by an addition and layer normalization. Better communication between the tokens in a sequence requires a higher complexity of such a model due to which multiple encoder blocks were stacked together to produce a non-masking output, after which this output was given to multiple stacked decoder blocks. Table 1.1 summarizes the configuration used in [9] and the BLEU score of the model for both tasks. Where $embd$ represents the dimension of each token after the embedding layer, N denotes the number of blocks stacked together, H is the number of heads in parallel, $Train\ steps$ is the number of steps taken by the model to reach an optimal result and, $drop_{res}$ represents the dropout ratio used on each residual connection.

Table 1.1 Configuration of transformer

Model	BLEU		Configurations				
	EN-DE	EN-FR	$embd$	N	H	$Train\ steps$	$drop_{res}$
Base Transformer	27.3	38.1	512	6	8	$1M$	0.1
Big Transformer	28.4	41.8	1024	6	16	$3M$	0.3

1.2.1 Embedding Layer

The embedding layer can be referred to as a representation matrix where each row corresponds to each token information. The embedding layer is used to encode the token and its position in a vector form. Each token is represented by two vectors, a token vector which stores the information of that token, and a positional vector which stores information about the position of that token within a sequence. Tokens can either be word-level or character-level. However, character-level tokens are more flexible to use for NLP-based tasks [13].

1.2.1.1 Token Embedding

Token embedding is the process of converting each token to a vector form of size $embd$. A batch of input sequences with size (B, T) will have a dimension of $(B, T, embd)$ after token embedding, where B is the batch size and, T is the time dimension which denotes a sequence length. Both encoder and decoder have their token embedding layer sharing the same embedding weight matrix of size $(T, embd)$.

1.2.1.2 Positional Embedding

Positional embedding is the process of converting the position of the token to a vector form of the same size $embd$ for the convenience of summing it with token embedding. Positional encoding can be formulated as a learnable parameter [14] or, could be a pre-defined mathematical function [9].

$$POE(p, 2 \cdot i) = \sin\left(p/1000^{(2 \cdot i/embd)}\right) \quad (1.1)$$

$$POE(p, 2 \cdot i + 1) = \cos\left(p/1000^{(2 \cdot i/embd)}\right) \quad (1.2)$$

Fig. 1.2 shows how [9] uses a fixed sinusoid function for each dimension of the positional vector of a token. Eqn 1.1 and Eqn 1.2 show the mathematical formula used in pre-defined Sinusoid position embedding of sin and cos components respectively.

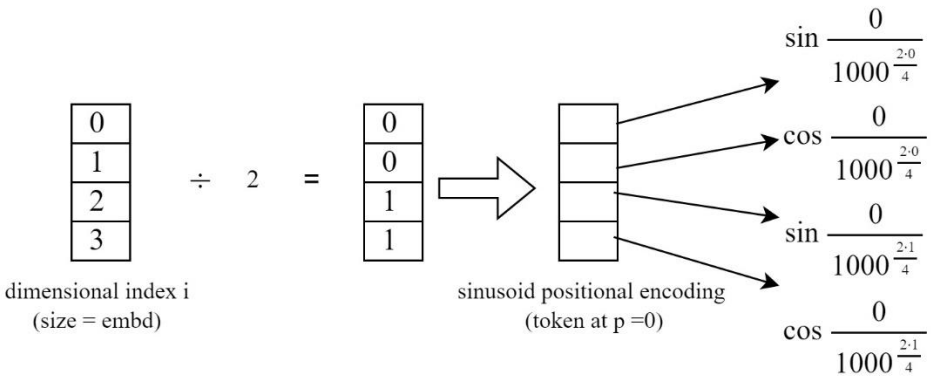


Fig. 1.2 Working of sinusoid embedding

Where p is the relative position of tokens in a sequence and i is the dimensional index of a particular token which ranges from $[0, embd)$. The mathematical working of Eqn 1.1 and Eqn 1.2 can be seen in Fig. 1.2. There are three main reasons for which Sinusoid positional embedding was preferred over a learnable encoding as described in Table 1.2.

Table 1.2 Reasons for using sinusoid embedding

Reason	Explanation
Periodicity	The sinusoidal function is periodic in nature and is crutial for capturing the sequential nature of tokens.
Constrained Values	The sine and cosine function ranges from –1 to 1. Hence, it doesn't introduce overly large or small values that could degrade the model's stability.
Easy to Extrapolate for Long sequences	Sinusoidal functions are smooth which is helpful in capturing contextual information for long sequences.

1.2.2 Attention Mechanism

Attention is the key part of the transformer model. This mechanism will establish a communication protocol among tokens. The attention mechanism ensures parallel computation, which proves to be less time-consuming than the existing models [7], [15]. There are two types of attention implementation, single-head attention and multi-head attention. Eqn 1.3 is the mathematical formula for the single-head attention mechanism whereas, Eqn 1.4 is used to find multi-head attention in which $head_i$ represents $SHA(Q_i, K_i, V_i)$. Here, Q and K are query and key vectors respectively in d_{qk} dimension and, V is a value vector in the d_v dimension such that each token in a sequence has a Q, K, V vector associated with it. Transformers uses the concatenation of Eqn 1.3 to achieve parallel computation used by Eqn 1.4.

$$SHA = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_{qk}}}\right) \cdot V \tag{1.3}$$

$$MHA = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_{qk}}}\right) \cdot V \tag{1.4}$$

Fig. 1.3(left) represents the single-head computation where the dot product between Q and K is known as a compatibility matrix. Masking is enabled for the decoder allowing it only to communicate either with itself or its previous tokens. Whereas, it is disabled for the encoder which denotes that each token can attend and communicate with each other token in the sequence. Also, the significance of softmax is to average out all the communicable tokens information. It can be seen in Fig. 1.3(right) that multi-head computes all the Q, K, V vectors parallelly for each

head which is then concatenated, followed by a linear projection due to skip connections.

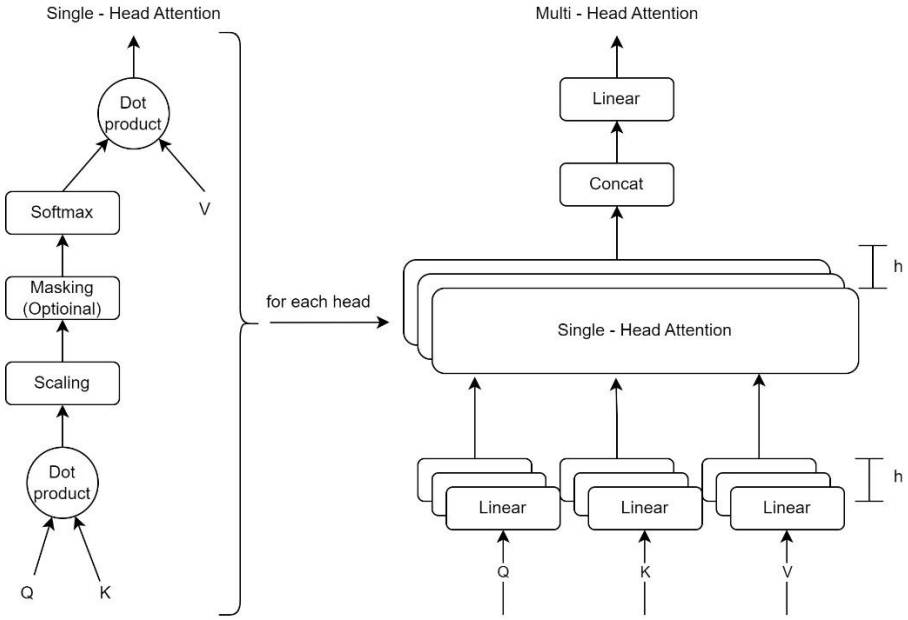


Fig. 1.3 Working of Attention Mechanism

The self-attention technique signifies that the query, key, and value vector are all taken from the same source. As seen in Fig. 1.1, the decoder (right block) and encoder (left block) both have a self-attention layer. In Encoder the Q, K, V are taken from the embedding of the input token sequence. However, the Decoder's first attention block takes a Q, K, and V from embedding the output token sequence.

1.2.3 Point-wise FFNN Layer

Transformer is all about communication followed by computation. Communication is done through an attention mechanism but, there must be something that helps the tokens to learn what they have communicated. [9] introduced densely connected Feed-forward neural net(FFNN) shown in Fig. 1.4. It was observed that the $hidden\ layer \in R^{4 \cdot embd}$, whereas $input\ layer \in R^{embd}$ and, output layer follows the input dimension due to compatibility with residual connection.

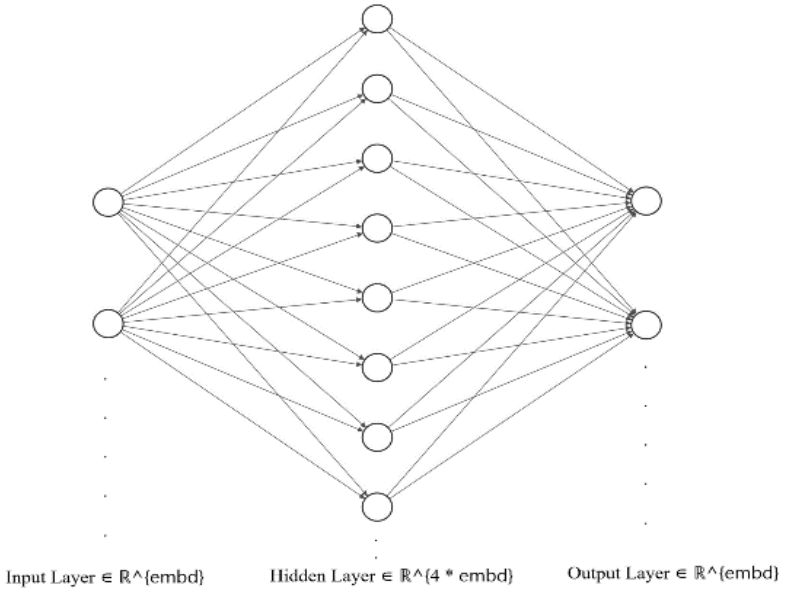


Fig. 1.4 Feed forward neural network in Transformer

A $RELU$ activation was used in the hidden layer for non-linearity. This network was used in both the encoder and decoder after the attention sublayer shown in Fig. 1.1 Since after the attention layer each token in a sequence has a vector that stores the information of its communication with every other token. Therefore, it is important to know that this network was operational token wise.

1.2.4 Layer Normalization

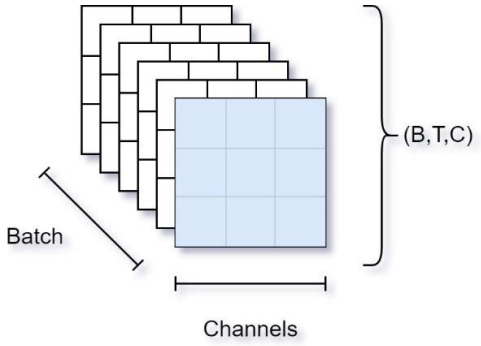


Fig. 1.5 Layer Norm in Transformer

According to Fig. 1.5 Layer normalization is the normalization of vectors along channel dimension i.e., $embd$. It was employed around each sublayer of encoder and decoder after the residual addition for training stability shown in Fig. 1.1. In the translation task we are gathering the attention vectors for each token

within a particular sequence and importantly there is no relation between other sequences which is the reason for applying a layer normalization rather than batch.

1.3 PROBLEM STATEMENT

Machine translation is an essential task in NLP. SMT and NMT are the basic two ways of performing translation. SMT is a traditional approach of mapping between words of one language to another and is also currently used by several researchers [2], [3]. However, SMT struggles to capture long-range dependencies, resulting in translations with unclear context and reduced accuracy.

On the other hand, Translation models require substantial amounts of text data to achieve high-quality results. The choice of dataset for a particular downstream task is also very critical. Due to the large parallel corpus of WMT2014 de-en and WMT2014 fr-en datasets, they are used very frequently for several research studies generally for supervised tasks [4], [5]. However, It was however observed that there is not much research on en-hi task due to limited dataset of English to Hindi translation. Proof of it is, it was observed that datasets with limited parallel corpus are used for zero-shot translation to create an inductive bias on unseen data [3], [6].

The most difficult challenge was to train large language model to train in a limited computational resource for translation task. This condition can affect the ability of transformer model to achieve SOTA results and it becomes more difficult when there are limited datasets like English-Hindi data.

These three problems were the baseline for this research. This thesis will work to solve mentioned problem and will explore the limits of pre-trained transformer-based model for English-to-Hindi machine translation. All these problems were also used to think and construct research questions which were eventually solved for successful completion of this research. The main object is to observe if further training of the trained model is effective for translation task with the challenge that GPUs resources were limited in power and number.

1.4 MOTIVATION

Transformer-based pre-trained language models have revolutionized the field of Natural Language Processing (NLP). Notable examples include BERT [16], ALBERT [16], RoBERTa [17], DistilBERT [18], GPT [3], T5 [19], BART [20], Switch Transformer [21]. These models have demonstrated remarkable success in

capturing universal language representations through extensive training on vast quantities of unlabeled text data. Fine-tuning these pre-trained models allows them to excel in specific downstream tasks like machine translation, even if they were initially trained for multi-task problems [19] or, other related tasks. It was also concluded that if the model is pre-trained on a machine translation task then the BLEU score will increase on further training [22]. In recent years, there has been a growing focus on developing models for translation tasks involving Indian languages. Key developments include:

1. **IndicTrans2** [23]: This model is created by AI4Bharat and it is a multilingual model which is designed to handle translation tasks across 22 scheduled Indian languages. It uses a large parallel corpus of 230 million bitext pairs extensively for Indic languages.
2. **IndicBART** [24]: This model supports 11 Indian languages, including Marathi, Hindi, and Punjabi. It has the architecture similar to mBART [25] and it has a six layer encoder-decoder transformer architecture.
3. **IndicBERT** [26]: Also developed by AI4Bharat, IndicBERT is a fine-tuned version of ALBERT [17]. It is known for its efficiency and reduced model size to solve multi-task NLP problems.

Therefore, observing all the above innovations in machine translation to the Hindi language the motivation behind this thesis is to:

1. **Check Fine-Tuning Limits**: Analyzed the potential of transformer-based pre-trained model on machine translation task through the fine-tuning process.
2. **Improve Translation Quality**: The motive was to increase the BLEU score for pre-trained model after further training on unseen data and validate the results with SOTA models.
3. **Contribute to NLP domain**: Since 2017 Transformer based encoder-decoder models have been popular among machine translation tasks. Hence the motivation was to contribute to this field with the robustness of a rarely used dataset.

1.5 CONTRIBUTION

1.5.1 Contribution 1: Taxonomic Analysis of X-formers

In the first phase of the research, a detailed taxonomic analysis of various modified versions of transformers was performed which was termed as X-formers. These modifications were introduced to address the inherent limitations of the

original transformer architecture, particularly in the context of machine translation tasks. The primary constraints addressed include:

1. **Generalizability**: The structural constraints between input and output text which effects the model ability to generalize across diverse text sequences.
2. **Long-range Dependency**: The models was not able to capture contextual information in long text sequences.
3. **Computational Complexity**: The traditional time complexity associated with the self-attention layer took quadratic time complexity which became a reason for high training time.

This research involved analyzing significant modifications proposed in various studies, focusing on their relevance to machine translation. The X-formers were classified based on their specific layer modifications within the architecture into pre-training variants and training variants. This provides a comprehensive mathematical reference to understand the latest transformer variants and aims to inspire researchers to develop hybrid X-formers that address multiple challenges concurrently, ultimately enhancing BLEU scores in translation tasks.

1.5.2 Contribution 2: Experimental Analysis for English-Hindi Translation

In the subsequent phase, experimental analysis was conducted to evaluate the performance of various pre-trained transformer-based models for the English-to-Hindi translation task. Despite the success of pre-trained models in several languages, there has been a notable lack of models specifically tailored for English to Hindi translation. Therefore, four pre-trained models were fine-tuned i.e., OPUS-MT, M2M100, mBART-50, and MADLAD-400 by using the IITB CFILT English-Hindi dataset.

The aim was to compare the translation quality of these models using the BLEU metric. None of the models had been pre-trained on the IITB CFILT dataset which acts as an inductive bias during fine-tuning. Hence, fine-tuning on the IITB dataset ensures the robustness of these models with unseen Hindi vocabulary and assess the quality of the dataset itself. Training was conducted on high-performance GPUs like A100 and V100, with model states and loss logs saved on the Hugging Face Private Hub.

The results indicated that OPUS-MT was the most effective model. mBART-large-50 also showed strong, making it a viable alternative for robust Hindi translation. This study not only highlights the potential of fine-tuning large pre-trained models for specific NLP tasks but also underscores the challenges posed by

limited computational resources. The findings contribute to ongoing efforts to improve machine translation models, particularly for less-resourced language pairs such as English to Hindi.

1.6 THESIS ORGANIZATION

This thesis is structured into seven chapters, each focusing on a different aspect of optimizing transformer models for English-to-Hindi translation through supervised fine-tuning. The chapters are organized to provide a coherent and logical progression from introduction to conclusion.

1. **Chapter 1: Introduction**

This chapter introduces the thesis by discussing the history and necessity of translation, the evolution of transformer models, and the relevance of this research. It includes the problem statement, motivation, and main contributions of the study. The chapter concludes with an overview of the thesis structure.

2. **Chapter 2: Literature Survey**

This chapter reviews existing literature related to translation and transformer models. It includes a summary table of 20 significant papers, highlighting their methodologies and findings, and identifies research gaps that this thesis aims to address.

3. **Chapter 3: Research Objectives**

This chapter defines the research objectives and questions to support the research. It provides a clear overview of the aims and a summary of the specific objective of the research.

4. **Chapter 4: Methodology**

This chapter describe the research methodology which includes data collection, data preprocessing, model training, and model evaluation. A proposed high-level design was constructed for readers better understanding. At last, the experimental setup and Python libraries/modules are also mentioned.

5. **Chapter 5: Results and Discussion**

This chapter presents the experimental results which includes result from model training with a comparison with existing English-to-Hindi translation models. Lastly, it mentions a discussion of the results to point out the limitations and key points of model performance.

6. **Chapter 6: Conclusion, Future Scope, and Social Impact**

This final chapter summarizes all the research findings in this study, then it discusses the future scope of this study, and at last it states the social impact of translation tasks in real world.

7. **References**

The section lists down all references cited in the thesis which were used for successful completion of experimental analysis and it also helps to support the credibility of this study.

1.7 SUMMARY

The first section of this chapter discusses a short overview on the history of transformer explaining the different ways translation task can be solved i.e., SMT and NMT. This section also points out the limitation of RNN based traditional approach for solving translation task. The second section describes in-depth working of a transformer architecture and includes topic like Embedding layer, Attention mechanism, point wise feed-forward neural network, and layer normalization. Next section includes problem statement where three major problems were identified like difficulty in capturing long range difficulty, limited large parallel corpus, and limited computational resources. Motivation was also included where few of the models developed by AI4Bharat was mentioned whose purpose was same as this thesis. A contribution section was added to showcase my research done so far that includes a taxonomic analysis and an experimental analysis on pre-trained transformer-based model for English-to-Hindi machine translation. At last, a thesis organization tells us the aim and scope of each chapter.

CHAPTER 2

LITERATURE SURVEY

This chapter demonstrates a literature survey that was performed explaining the related works carried out on transformer model to enhance their performance in terms of the time complexity it had on self-attention layers as well as the BLEU score. At Last, an accumulation of the research gaps was shown addressing the limitations among all the related works.

2.1 OVERVIEW

According to the research performed two major review was performed. One was totally focused on the variant of transformer which was build on the motive of fixing the limitations of vanilla transformers. The second review was based on the various pre-trained transformer-based model which performed fine-tuning with various goal and task. In short, first review of papers includes type of X-formers and second review was regarding the type of pre-trained transformer-based models particularly for translation tasks. The complete process of review started after studying the base paper of transformer [9]. The roadmap of review process can be seen in Fig. 2.1.

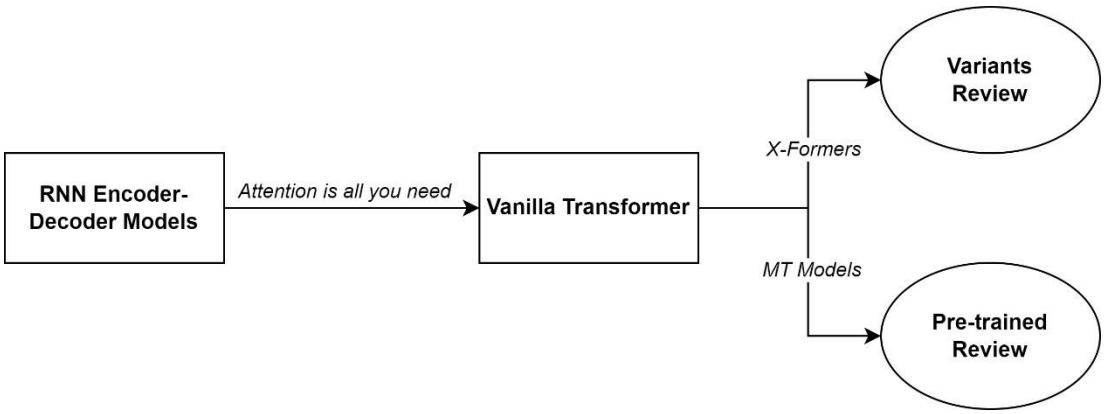


Fig. 2.1 Roadmap of survey process

In the early days RNN based models were very popular and used frequently for encode-decoder tasks like machine translation. However, the major problem they had was of capturing the long-term dependency between tokens among a sequence. Several experiments were then held to further enhance the vanilla

architecture by either modifying the pre-training encoding section [14], [27] or the training phased encoder-decoder blocks [15]. The term used to describe the variants of the vanilla transformer is known as X-formers. Although transformers are highly in demand for NLP-related tasks and are also used with images, videos, etc. X-formers were introduced to improve the constraints of vanilla architecture: (1) Generalizability – due to structural constraints between the input and output text, (2) Long-range dependency – transformers struggle to capture contextual information in a large text sequence. (3) Computational Complexity – parallel computation of query, key, and value vectors in transformer leads to increasing time complexity. This review aims to provide a comprehensive understanding of the variants of transformer which addresses the above-mentioned issues. However, this chapter only mentions some of the important variants that target at least one of the mentioned issues in the translation domain. Therefore, it is methodical to classify the X-formers based on the division of the current architecture of the transformer mentioned in Fig. 1.1 i.e., pre-training variants and training variants.

2.2 REVIEW OF RELATED WORKS

Transformer finds it difficult to process long sequences due to $O(n^2)$ time complexity of the attention mechanism [9], [28]. However, the CNN and RNN-based model has $O(n)$ time complexity per layer. Therefore, Architecture like Longformer [14], introduced the dilated sliding window concept which tries to mimic the technique of dilated CNN but also kept their receptive field large to capture the context of tokens at a longer distance. Note that Longformer produced time considering the portion of the original matrix(sparsity) through sliding window protocol. We could rather use the low-rank matrix property of the compatibility matrix [28] to reduce the size of it. Linformers [28] concluded attention weight matrix can be approximated well by a few vectors with the largest singular values taking only $O(n)$ time and space complexity. Another efficient way of doing this is to only search for values close to query q_i in $Q \cdot K^T$. Reformers [29] propose another method to reduce the compatibility matrix just like [28] but, reformers use Locality sensitive hashing (LSH) which finds collision with maximum probability. However, Reformers reduced the time complexity to $O(n \log n)$ but still achieved results close to vanilla [9]. Meanwhile, rather than focusing on absolute position encoding [9], a rotation matrix can be used to efficiently encode relative position (RoPE) in which, the original Q and V vectors are rotated by some angle θ [27]. The BLEU score of Roformer was identical to the vanilla approach [27], as it targets model generalizability rather than long-range dependency. Generally, pre-trained transformer models generate state-of-the-art results by being trained on a large corpus of data [14], [27], [30]. However, learning the sequential order of tokens on small text data with error-free translation is difficult. [31] applied a fast-gradient method on multi-head attention to approximate the gradient computation in a faster and less accurate way. Absolute positional embedding is inefficient because of the fixed range of sin and, cos. Therefore, it is preferred to use relative positional encoding to better capture long-range dependencies between tokens [27], [31].

Furthermore, the pre-trained encoder-decoder transformer shows good performance on various NLP tasks such as text summarization [11], [14], classification, and translation.

Table 2.1 Summary of relevant X-Formers

X-Former	**Time Complexity**	**Modification Layer**	**Metrics used**
Transformer with Untied Positional Encoding [32]	$O(n^2)$	Self-Attention Layer	QNLI, QQP, SST, CoLA, MRPC, RTE, STS
RoFormer [27]	$O(n)$	Embedding Layer	BLEU
Transformer Fast method-Relative Positional Embedding [31]	$O(n)$	Embedding Layer	BLEU
Reinforcement Learning-Positional Encoding [7]	$O(n^2)$	Positional Encoding Layer	BLEU
Longformer [14]	$O(n)$	Self-Attention Layer	BLEU
Fast Gradient Method-Multi head Attention [31]	$O(n^2)$	Self-Attention Layer	BLEU
Zero Masked-Self Attention[7]	$O(n)$	Self-Attention Layer	BLEU
Multi-Query Attention [33]	$O(n)$	Self-Attention Layer	BLEU
LORAMOE [34]	$O(n)$	Feed-Forward Neural Network Layer	BLEU, Accuracy, F1-Score
Group Query Attention [15]	$O(n)$	Self-Attention Layer	BLEU
Reformer [29]	$O(n \log n)$	Self-Attention Layer	BLEU
Linformer [28]	$O(n)$	Self-Attention Layer	BLEU
Multi-split Reversible Transformer [35]	$O(n)$	Self-Attention Layer	BLEU
REDER [36]	$O(n)$	Self-Attention Layer	BLEU

Most of the study focuses on introducing new mathematical techniques to either improve the time and space complexity of original self-attention or improve the positional embedding. However, a variant of the original multi-head attention architecture was proposed where each head still uses multiple query vectors but only utilizes a single key and value vector for each head [15], [33]. As a result, MQA shows degradation of BLEU score compared to [9] due to loss of information. However, [15] proposed GQA which produces results like MQA by assigning a subset of Queries to a single key and value vector. A full pre-activation skip connection produces good results by handling different data distribution at the start itself [37]. Therefore, Enhanced transformers prefer pre-activation skip connection by performing layer normalization at the beginning. It is also unnecessary to communicate with the current token, as it only increases computational costs. Instead, convert the principal diagonal of the softmaxed compatibility matrix to zero [7].

No study focused on the FFNN layer as it only computed the communicated results. Also, during fine-tuning using too much data can sometimes make the model forget important information it learned before. [34] introduced multiple parallel experts called LoRA connecting with an adapter in the FFNN layer to freeze the main part of the model during the training. $O(n^2)$ memory requirement also poses a threat when using a deeper transformer model. This can be solved by reconstructing some of the activations instead of storing them [35]. The advancement in transformers has gone so far that the original multi-head attention with FFNN can be used with a small reversible design [36] to translate from English to Hindi even after training the transformer with the Hindi-to-English dataset. At Last, All the X-formers discussed so far have shown an improvement in either time complexity of self-attention or due to the modification in a particular layer. The summary of all the discussed X-formers are also shown in Table 2.1.

Recent studies have shown a growing interest in fine-tuning pre-trained Transformer based LLMs. Nevertheless, there are several challenges and considerations in this field. There are three states of model i.e., before pretraining, after pre-training, and after fine-tuning. Before and after pre-training, the parameters of the pre-trained model change significantly. However, during fine-tuning, the model parameters don’t change that much [38]. BLEU is the most widely used metric for translation tasks however [22] also used cross entropy for evaluation. The transformer model consists of encoder and decoder blocks which take tokens as input. A token is the simplest unit of text that can be a character, word, subword or subsequence in a text sequence. Training a transformer model on character-level text data requires setting up a deep architecture with billions of parameters otherwise it would lead to training instability and slow convergence [39]. And since the transformer requires $O(n^2)$ time during self-attention. Therefore, it was proposed in [39] that a good alternative is to first train a transformer on higher-level tokenized

data i.e., subword, and then fine-tune it on character-level data with some degradation in results. Such a deep architecture can decrease model efficiency. Hence, a residual connection can boost neural network-based model performance [40]. One of the bottlenecks of the Transformer is the time complexity of the attention mechanism. This motivated researchers to come up with different approaches. [28] Proved that the compatibility matrix obtained in the self-attention process is a low–rank matrix and this observation was considered a baseline to propose an efficient self-attention mechanism with $O(n)$ time complexity. Unlike the Linformer model in [28] which was trained on text data with $O(n)$ time, [41] used a trained transformer on $O(n^2)$ time but introduced an RNN version of self-attention for fine-tuning stage. This new selfattention was able to be approximated with randomized feature maps. Finally, it was concluded that the training cost was less as compared to training the RNN version from scratch.

Table 2.2 Summary of pre-trained model for translation

Model	**Size**	**Multilingual**	**Base Model**	**Evaluation Metric**
OPUS-MT	76×10^6	No	Marian NMT	BLEU, spBLEU, chrF, chrf++, COMET
M2M100	484×10^6	Yes	Transformer	BLEU, SacreBLEU
mBART-50	611×10^6	Yes	Transformer	BLEU
MADLAD-400	3×10^9	Yes	T5	SacreBLEU, chrF

According to [42], a Large pre-trained language model (PLM) like variants of BERT, GPT-based model, and transformer does unsupervised training on a large corpus of text data. It was found that the transfer learning approach of fine-tuning on a smaller, labelled dataset has been shown to achieve SOTA performance on many NLP tasks including translation tasks. These findings were observed and this research found four major PLM trained on large corpus [43], [44], [45], [46]. The summary of all the pre-trained model reviewed in this research can be seen in Table 2.2. These models are specifically developed for machine translation task.

2.3 RESEARCH GAPS

Completion of above review process gave us some observation. One such observation is that BLEU score is very common and preferred evaluation metric for translation model. BLEU score was calculated on the test dataset usually after a fixed number of training steps. BLEU score ranges from 0 to 1. However, 100 is multiplied by it which scales the range from 0 to 100. A high BLEU score represents that the translated text from the model is nearly similar to the original translated text and, a low score denotes a bad translation from a model. Another observation was that all the related proposed model had an aim of reducing time and space complexity of self-attention mechanism and especially reducing the operation of compatibility matrix.

Also, there were few gaps which were also observed such as usage of English-Hindi datasets. Limited use of pre-trained transformers for machine translation task. The transformer was introduced in place of RNN and CNN based encode-decoder architecture on the fact that it will be a more efficient way in capturing long range dependencies between tokens in a sequence. But, It was observed in almost every paper that the disadvantages of X-Formers is that they were not able to capture long documents. However, only one X-Former called Longformer was able to accomplish this.

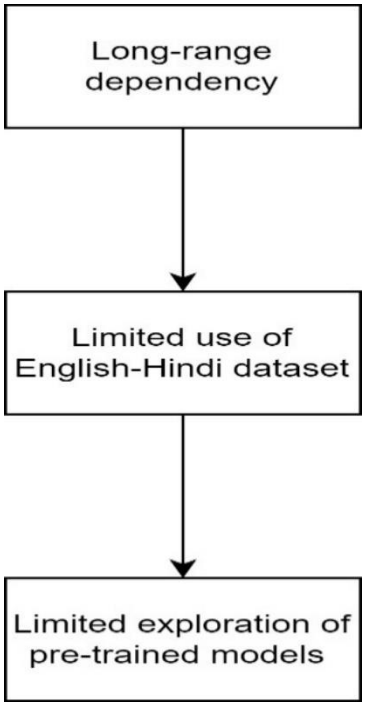


Fig. 2.2 Flow of identification of research gaps

The step-by-step identification of research gaps were shown in the Fig. 2.2. All the reviewed study were totally focused on translation say x language to English which was due to the global effectiveness of English language. There were almost 3 to 4 models which were performing English to Hindi translation. All of which were develop and created by A14Bharat start-up incubated in IIT Madras. This research therefore became a research gap for these findings and eventually was one of the motivations for exploring the limits of pre-trained transformer models on a large English-Hindi dataset developed by CFILT at IIT Bombay. Thereafter, a search criterion was created for identifying the relevant pre-trained model for an experimental analysis. However very few of them were relevant enough due to the reasons mentioned in problem statements. The search criterion used are mentioned below:

1. Among all the Seq2Seq NLP, only machine translation tasks were taken under consideration.

2. Only encoder-decoder transformer-based models were studied for this paper.

3. This paper only focuses on models that is not pre-trained on IITB-English-Hindi dataset.

2.4 SUMMARY

This chapter summarizes all the latest modifications of transformer and the improvement over vanilla transformer such as Linformer, Reformer, RoFormer, Longformer, TUPE, GQA, MQA, etc. The major X-formers were showing improvement on the time-complexity of self-attention mechanism and specially where query and key vector are concatenated using dot matrix product. Longformer was an exceptional X-former. It was able to communicate with long-range sequences. Since Transformer were introduce in place of RNN due to long range dependency limitation of RNN. But the quadratic complexity in both space and time were getting the training time slow for transformers. Hence It is now clearly known that the self-attention quadratic complexity is the only major problem with transformers and therefore, this literature review was very useful for this research and the future scope where a Hybrid transformer model is to be built from scratch with the aim of lie time complexity. Another set of review was focused on the latest pre-trained model. All the models were selected in a way that they were Suitable for a Machine translation task. It was observed that the model size was an effect and hence, Varying model sized were selected for better and quality research. At last a search criterion was mentioned to select four models for further training to capture better translation quality through BLEU score.

CHAPTER 3

RESEARCH OBJECTIVES

This chapter states the main aim for conducting the research by solving the framed research questions. This chapter was created for solving the research gaps found in the relevant related works stated in above chapter. Furthermore, the research questions will be answered in below chapters using an experimental analysis.

3.1 OVERVIEW

Several transformer-based architectures have been built to perform human-like translation through rigorous training on large corpus data. Various pre-trained models have shown their potential in selective languages, but limited models that solve English to Hindi translation due to the unavailability of a large parallel corpus. Therefore, this research focuses on the IITB English- Hindi dataset which is one of the first large corpora for en-hi translation tasks containing large parallel English-Hindi and monolingual Hindi corpora. This research fills all the research gaps by fine-tuning four pre-trained models on the IITB English-Hindi dataset, namely OPUS-MT, M2M100, mBART-50, and MADLAD-400.

Fine-tuning the model on the IITB dataset helps us to understand the robustness of these well-known models on unseen Hindi vocab and it will also help us to know the extent of quality of this dataset. To further validate the scores, this research intends to compare the results with some existing translation models trained on the IITB dataset. The research question will also address two existing models such as “Transformer with Untied Positional Encoding” (TUPE) and “Sliding window self-attention” to understand the working for capturing long range dependency through some mathematical understandings.

3.2 RESEARCH QUESTIONS

The following research questions will be addressed based on the literature review and research gaps identified:

1. Do pre-trained transformer-based model perform well for translation task?
2. Why are English-Hindi Dataset being less used for experimental analysis?
3. How can transformers effectively process long sequences?

3.3 SUMMARY

Based on the research gaps mentioned in previous chapter, three research questions were framed. All the research questions are carefully designed and solved in next chapter. The first research question was solved by developing a pipeline to further train four pretrained model. Second research question will be solved by using IITB CFILT dataset in which major reasons will be discussed for less usage of English-Hindi dataset with detailed description of CFILT data. The last question will be solved by understanding the mathematical concept for capturing long sequences in later chapters.

CHAPTER 4

METHODOLOGY

This chapter explains the methodology used for an experimental analysis to fine-tune four pre-trained models. This chapter showcases a detail work of the proposed work such as collection of data, data pre-processing, model training, and model evaluation. The setup used during this work is also explained for detailed view of the work.

4.1 OVERVIEW

This section will describe each model used in this work. All the model were selected through extensive research in Hugging face public hub. A model pipeline was created to further store the results in a private hub. This paper focuses on accessing the power of four models of machine translation. These models are OPUS-MT-en-hi, M2M100, mBART-large-50, and MADLAD- 400.

OPUS-MT [46] is a transformer-based model with nearly 76 million parameters. It is trained on open parallel corpus projects and utilizes the Marian framework for efficient training. This model is known for its ability to translate between various languages. Many-to-Many 100 (M2M100) [43] is a multilingual language model (MLM) that follows transformer architecture with nearly 484 million parameters. This PLM is developed by Facebook AI to translate sentences among 100 different languages without relying on English as an intermediate language. mBART-large-50 [45] is a version of the Multilingual BART (mBART) model. It is trained on a large multilingual corpus and can be translated into 50 different languages. The large version has around 611 million parameters and, has the capacity for capturing complex text patterns. MADLAD-400-3B-MT [44] is a large multilingual model for machine translation tasks and is based on T5 architecture [19]. It was developed by Google with training on around 1 trillion tokens and, can translate into over 400 different languages.

These models require a large number of computational resources with huge training costs. Therefore, according to [42], the proposed method used the concept of transfer learning to fine-tune mentioned models. Finally, the HLD of the proposed methodology can be seen in Fig. 4.1.

4.2 PROPOSED WORK

Hugging face Hub was considered as a server for the proposed work IN which pulling of model and dataset took place from public hub and, after further training of the pre-trained models these models were pushed back to a private hub. Python programming was used to implement the model pipeline shown in Fig. 4.1. The algorithm for this approach contains four major parts i.e., Loader function, tokenization of text, trainer function, and evaluation function. Further section will be based on these parts of HLD.

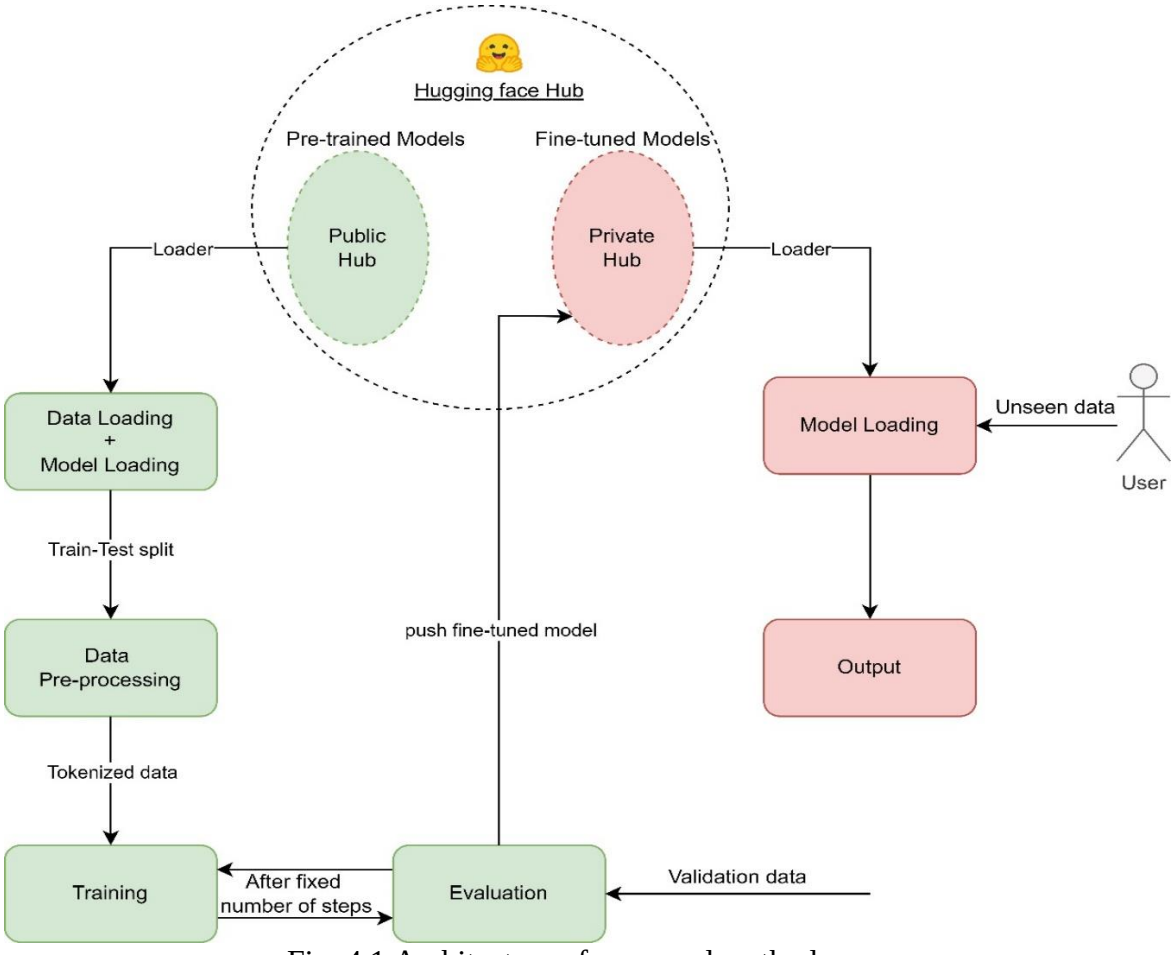


Fig. 4.1 Architecture of proposed method

To increase the generalizability of the result, the selected model contains sizes from very small to very large which can be seen in Table 2.2. Where, the en-hi version of the OPUS-MT model, M2M-100, a large version of mBART-50 has parameters in million and, MADLAD-400 has its parameters in billion. However, OPUS-MT is based on the Marian framework written in C++ language and, it was considered for this research due to its usage and its size.

4.2.1 Data collection

The hugging face dataset hub contains several datasets related to the NLP task. Therefore, this study used Hugging Face to ingest one of the most frequently used datasets known as IITB-english-hindi. This data was developed by the Computation for Indian Language Technology (CFILT) at the Department of Computer Science and Engineering, IIT Bombay [47]. This dataset contains a large parallel corpus of English and Hindi. This dataset contains a large parallel corpus of English and Hindi. The size of the dataset is nearly 190 MB with 3 splits in it i.e. train, test and, validation. According to the latest updation in the hub, it contains a total of 1.66 million train data, 2507 test data and, 520 validation data.

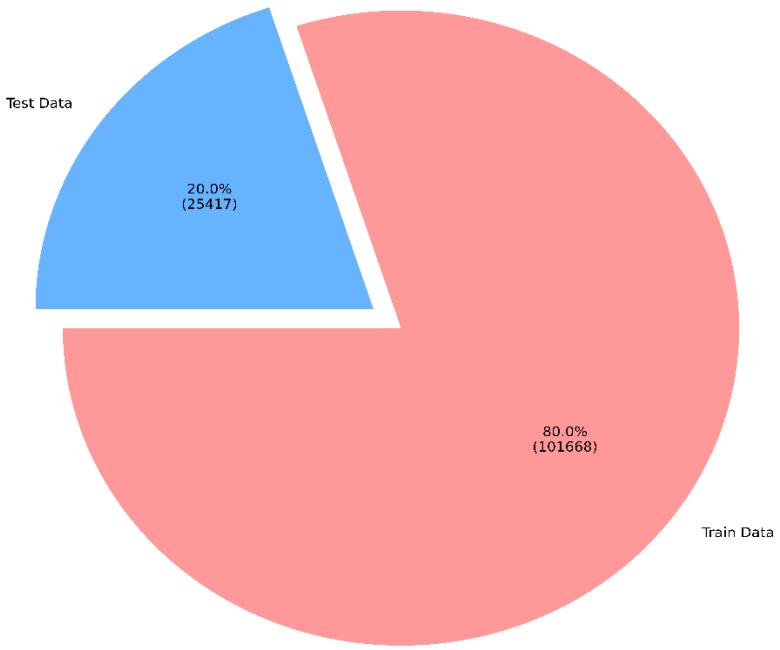


Fig. 4.2 Dataset composition

The number of rows in test data i.e., 2507 is too low for proper evaluation of the model under study. Also, this experiment was performed only to explore the potential of different-size models. Due to the above reasons, A subset of a total of 1,27,085 parallel sentences were extracted from train data. Fig. 4.2 represents the splitting of train and test data. Here, test data is considered as validation data.

4.2.2 Data pre-processing

The data pre-processing step was divided into two phase which is EDA and feature engineering. The data pre-processing was a simple procedure according to this use case since all the dataset was captured from Hugging face and it was already having not null values and no duplicate rows. Although there is no scope of capturing the outliers in this dataset as it contains text data. However, the sequence length can be captured using univariate EDA.

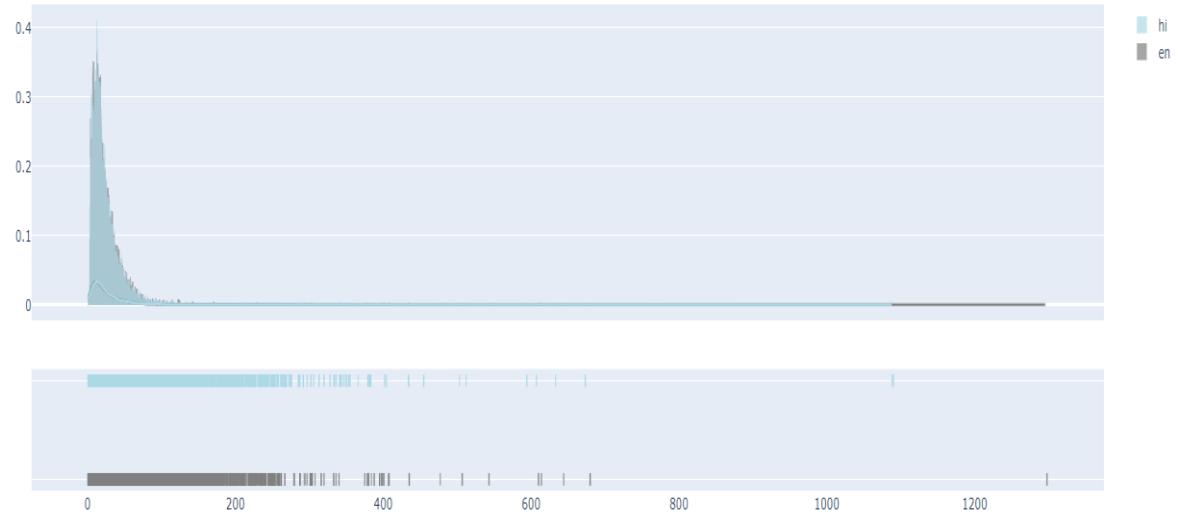


Fig. 4.3 Probability distribution of train sequence length

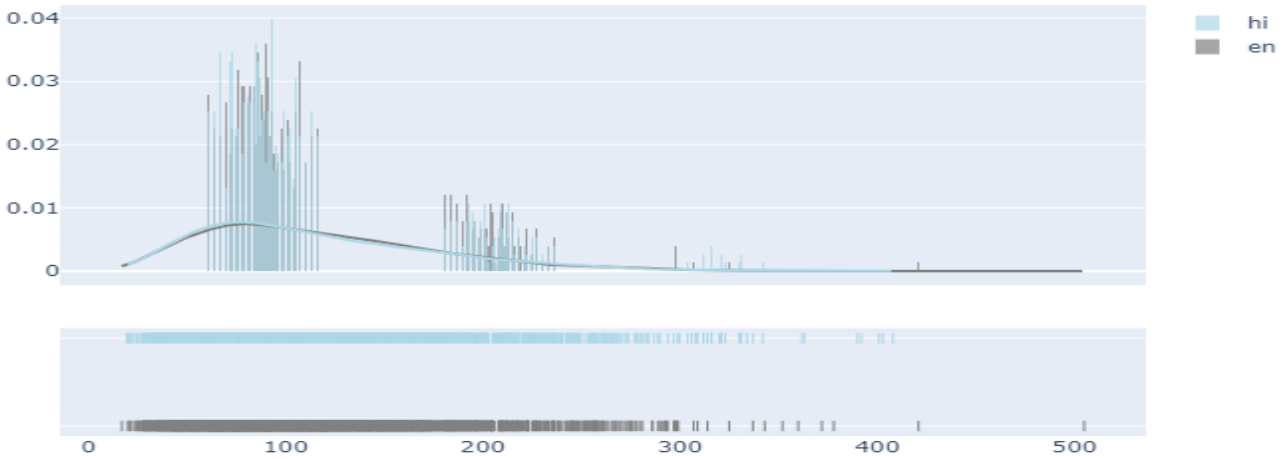


Fig. 4.4 Probability distribution of test sequence length

Fig. 4.3 and Fig. 4.4 shows distribution curve of the sequence length of all the sequences in train and test dataset respectively. The x-axis denotes the sequence lengths of the sentences and, y-axis denotes the probability of occurrence for corresponding sequence length.

Feature engineering was then applied for converting the raw text data into useful tokenized data. This phase of the experiment first collects the pre-built tokenizer of each model under study. Fig. 4.5 shows the vocabulary size of each tokenizer. Vocabulary works like a hash map of the tokenization process for converting text into tokens. Next, the source language was defined as English and the target language was defined as Hindi. Now, with the help of the vocab of each model, the English and Hindi sentences were converted into a list of integers which are indices of the vocabulary. Lastly, a data collator was initialized to set the padding according to the maximum sequence length in each batch of sequence. The tokenized data and the data collator were then passed to the training phase as shown in Fig. 4.1.

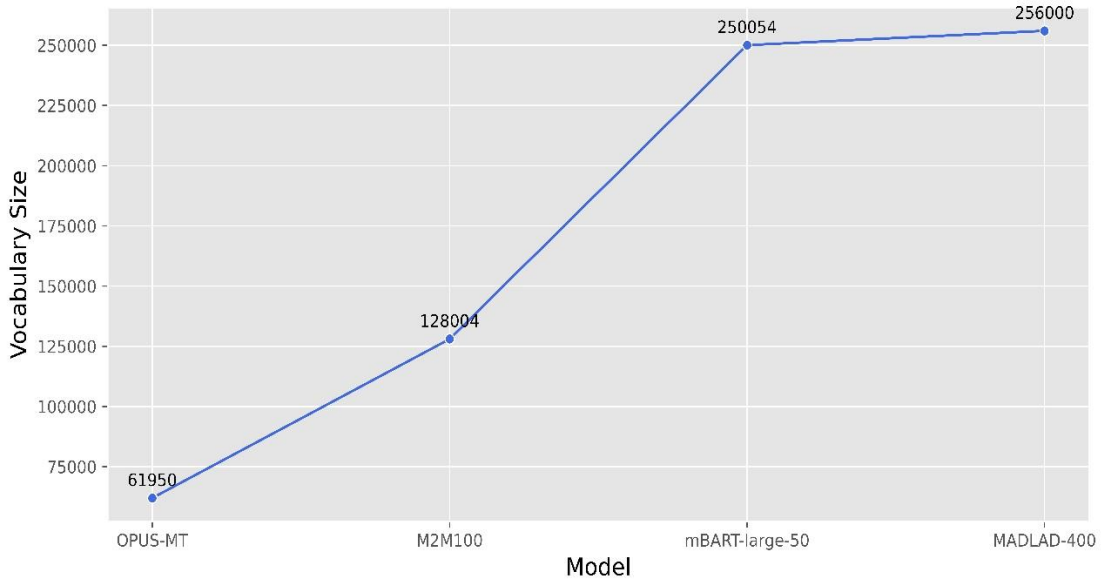


Fig. 4.5 Plot of vocabulary size of tokenizer

4.2.3 Model Training

The first requirement before model training is to acquire the base model which is intended to fine-tune. This study uses the transformer model hub of the Hugging Face community to access all four base models mentioned in Table 2.2. The transformer library contains several Python modules. Therefore, the Auto-ModelForSeq2SeqLM module was used for fine-tuning OPUS-MT-en-hi model, the

M2M100ForConditionalGeneration module was used for M2M100 model, MBartForConditionalGeneration module was used for mBART-large-50 and, T5ForConditionalGeneration module was used for MADLAD-400 model.

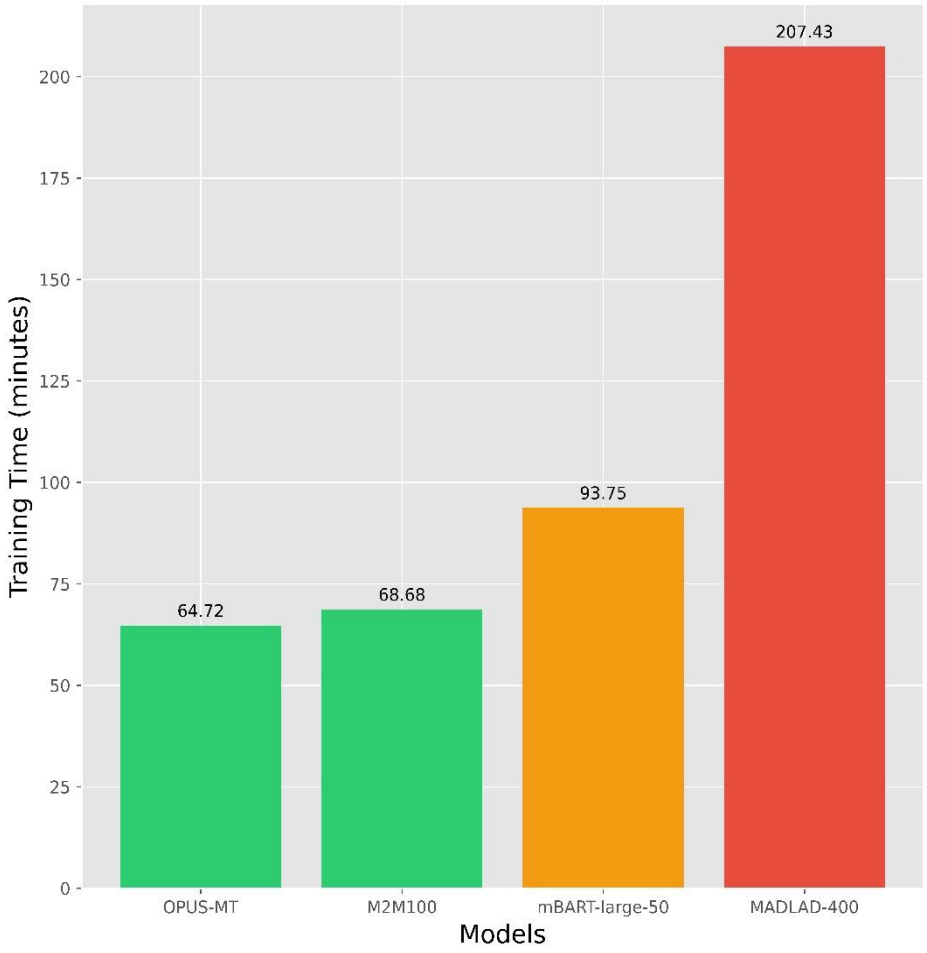


Fig. 4.6 Plot of training time of models

Two high-performance GPUs were used to train all these models. It was provided by Google Cloud namely V100 (16GB) and, A100 (40GB). Except OPUS-MT model all others were trained in A100 due to their increasing model size. However, the training was very much dependent on the batch size taken for training data. The model and their checkpoints gathered from the transformer library were sent to respective GPUs for training. The training cost was very high for MADLAD-400 as compared to other models due to its size and batch size. It is denoted by red colour in Fig. 4.6.

4.2.4 Model Evaluation

Several studies show that the BLEU score is one of the most frequently used evaluation metrics to check the quality of translated text with respect to human reference translation [43], [44], [45], [46]. However various other metrics can be seen in Table 2.2. For many NLP tasks common metrics like accuracy, precision, F1 score, etc. can be used. However, this paper used BLEU to measure the quality of text generated from models.

$$BP = \begin{cases} 1, & m > h \\ e^{(1-h/m)}, & m \leq h \end{cases} \tag{4.1}$$

$$BLEU = BP \times e^{\left(\sum_{k=1}^{N} w_k \cdot \log p_k\right)} \tag{4.2}$$

BLEU calculates a similarity score based on the n-gram common between the model-generated translation and the human reference translations [8]. Its mathematical equation is denoted by Eqn 4.2 in which BP means brevity penalty. Eqn 4.1 represents a formula for BP that penalizes short-length translations [48]. In Eqn 4.2, p_k denotes the geometric mean of n-gram precisions till length N and, w_k denotes weight. Whereas in Eqn 4.1, m denotes the length of model translation and h denotes the corpus length of human reference translation.

4.3 EXPERIMENTAL SETUP

Table 4.1 Training parameters of model

Arguments	**Model**			
	OPUS-MT	**M2M100**	**mBART-50**	**MADLAD-400**
Train batch size	64	64	16	8
Test batch size	64	64	16	8
Learning rate	2×10^{-5}	2×10^{-5}	2×10^{-5}	2×10^{-5}
Weight decay	10^{-2}	10^{-2}	10^{-2}	10^{-2}

The most important factor for the fine-tuning process is to set the training arguments of the respective models. These parameters may vary according to the downstream task and the computational resources available. This study captured five

major arguments which can affect the performance of the translation. The arguments are: Train and test data batch size for each step of convergence, learning rate, and weight decay to prevent overfitting. The value of the arguments for all four models is explained in Table 4.1.

4.4 SUMMARY

This chapter demonstrates all the steps that were taken to successfully develop a fine-tuned versions of four models namely OPUS-MT-en-hi, M2M100, mBART-large-50, and MADLAD- 400. The steps like data collection, EDA, feature engineering, model training, model evaluation metrics were described in a detaild way. A detailed view of the HLD proposed method was also shown to make the process easy and understandable. In fine-tuning process, the model parameters are to be chosen very carefully and hence, a table was described showing all the values of training arguments. At last, this chapter is dedicated to showcase the experiment method from data collection to model deployment as all the models a pushed back to Hugging face private Hub.

CHAPTER 5

RESULTS AND DISCUSSION

This chapter states the results generated after further training of the pre-trained models. This chapter includes the Experimental results and discussion with validation of result through comparing it with existing SOTA models for English to Hindi translation in IITB dataset.

5.1 OVERVIEW

This chapter will answer all the three-research question based on the results of the training of 4 machine translation model. This section will also validate the result based on the comparison with SOTA models that were trained on IITB dataset. The research question on how can transformers effectively process long sequences? Can be seen using TUPE method. TUPE used a different scaled dot product by using different projection weights for each embedding taken input separately. Now, adding one extra scalar learnable bias term to TUPE will introduce inductive bias to learn relative position. The learnable bias term ensures that the encoding vectors are not stuck in a range of –1 to 1 and can consider long-range sequences. The sliding window variant can also ensure that tokens under communication are selected after a dilated factor of which can increase the receptive field for training. The increase in the receptive field ensures long-range communication between tokens. Several training variants also show a reduction in memory consumption in their technique which increases the efficiency of communication with long ranged tokens with less time complexity.

5.2 EXPERIMENTAL RESULTS

The training process was rigorously performed by GPUs with limited resources like RAM. After each epoch of training, the BLEU metric was applied to the predictions of validation data as mentioned in Fig. 4.1. For better comparisons, three extra models were studied. Earlier, a Sequential Adaptive Memory (SAM) translation system was introduced on the IITB dataset but performed very poorly on it. It is based on the architecture of the neocortex area of the human brain [49]. Subsequently, two novel models were introduced namely Context-Based Forward Encoder (FE-Encoder) and, Context-Based Backward Encoder (BE-Encoder). Both

consist of multi-level Gated Recurrent Unit (GRU) [47]. BE-Encoder was considered as SOTA with 44.34 BLEU points.

Table 5.1 Comparison analysis of BLEU score

Model	BLEU	Gen Len	Steps
SAM	17.08	-	8
FE-Encoder	43.52	-	-
BE-Encoder	44.34	-	-
OPUS-MT	**89.11**	**7.59**	**3178**
M2M100	75.19	12.93	3178
mBART-large-50	86.83	11.74	12710
MADLAD-400	31.56	10.39	15818

The results of fine-tuning can be seen in Table 5.1. It was found that OPUS-MT outperformed all other models by achieving a BLEU score of 89.11. mBART-large-50 was also very close to OPUS-MT with a BLEU score of 86.83. MADLAD-400 did not show good performance among the fine-tuned models. However, its BLEU of 31.56 outperformed the SAM approach. M2M100 shows decent performance among fine-tuned models but still, it shows higher BLEU than SAM and Context-based Encoders. In Table 5.1, Gen Len represents the average generation length of a prediction made by models on a validation set whereas, steps denote the number of training steps taken by the model.

A log was created to capture the train loss and validation loss for each model under study. Fig. 5.1 shows a visual representation of the loss learning curve of the models. Due to limited trainable resources, MADLAD-400 shows near zero loss after training for 15818. However, it may change after further training and subsequently, its BLEU can also improve. Different models were trained for different numbers of steps. OPUS-MT and M2M100 were trained for 3178 steps and showed 0.20 and 0.26 validation loss respectively. Since the other two models are comparatively very large. Therefore, they were trained for a larger number of steps. Mbart-large-50 was trained for 12710 steps and achieved a validation loss of 0.15.

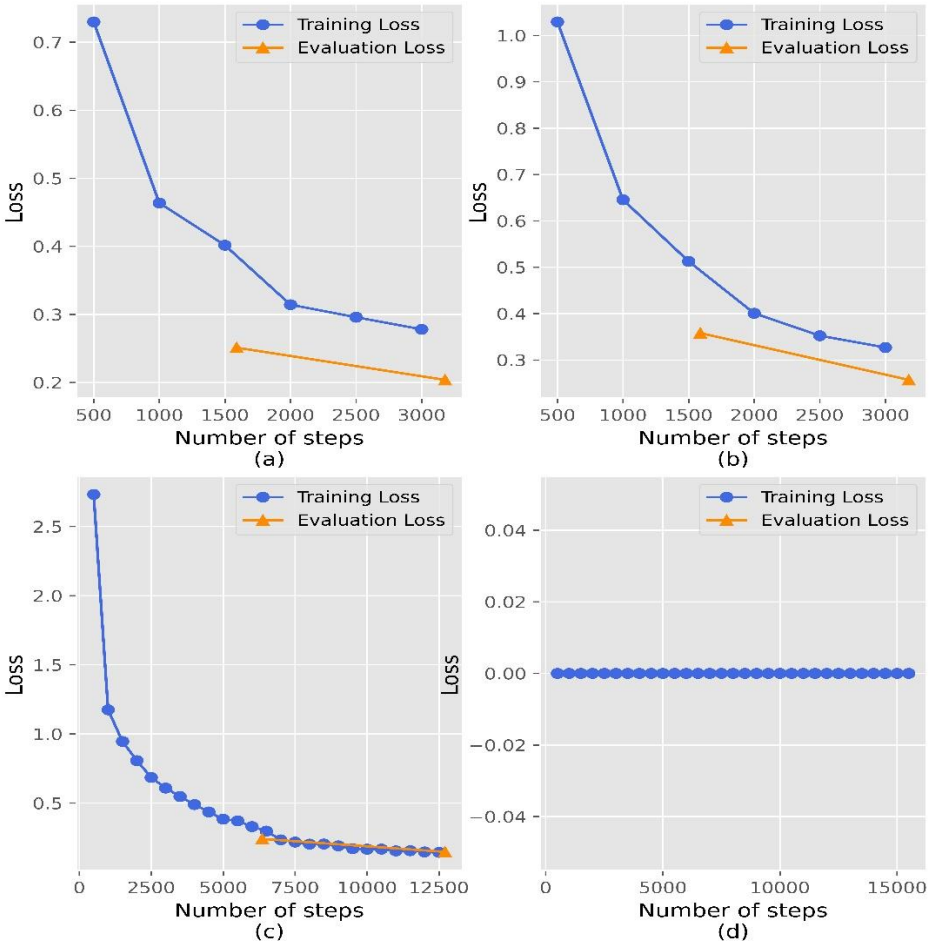


Fig. 5.1 Loss curve of (a)OPUS-MT (b)M2M100 (c)mBART-large-50 (d)MADLAD-400

5.3 DISCUSSION

There are also many relevant text quality evaluation metrics other than BLEU. However, This can be a motivation for other researchers to explore metrics like spBLEU, chrF, chrf++, and COMET. Furthermore, experiments could be performed based on the entire dataset of IITB-English-Hindi. But, this would require huge computational power. Hence, the future scope of the study will intend to carry forward research on High RAM GPUs to counter the low BLEU score of MADLAD-400. Lastly, this study can be important to all the researchers who aim to train large-sized models on an NLP task but also suffer from limited availability of computational resources.

This research also extracts the important modifications of transformers specifically useful for machine translation tasks. A classification was made for different variants based on their modification on a particular layer in Transformer architecture. It was found that the three major challenges for creating a transformer model were its computational complexity, its ability to capture long-range tokens, and its ability to perform reversible translation. Therefore, future developments of transformers should focus solely on addressing the issues identified in this paper. Finally, this study also provides a mathematical reference for a better understanding of the latest transformer variants and aims to motivate researchers to build a hybrid X-former that solves all the challenges that promise improvement in BLEU score.

5.4 SUMMARY

This chapter intends to showcase the experimental result that were gathered after training pre-trained models extensively on Large shared GPUs of Google. This chapter also discusses the findings and limitations of results. All the three research question were also answered in the previous. It was properly stated from results of four pre-trained models that fine-tuning pretrained model can perform very well irrespective of the model size. It was also clearly mentioned that English-Hindi dataset were less used for experimental use cases due to small dataset of parallel corpus of English-Hindi. However, it was shown in the dataset collection section under methodology that IITB CFILT dataset is one of the popular large corpus datasets and hence was utilised for this study to demonstrate a comparative analysis. This chapter also mentions how transformers process long range sequence through sliding window concept and TUPE’s bias term.

CHAPTER 6

CONCLUSION, FUTURE SCOPE AND SOCIAL IMPACT

6.1 CONCLUSION

This study demonstrated the power of the transfer learning technique on four different models namely OPUS-MT, M2M100, mBART-large-50 and, MADLAD-400. All the models were not pre-trained on the IITB CFILT dataset which set up an inductive bias during fine-tuning. All the models performed training on high computational GPUs like A100 and V100. Hugging Face Hub was used to save all the model states and create logs of losses. Thereafter, it can be concluded that OPUS-MT is one of the most effective Machine translation models to translate English to Hindi sentences with 89.11 BLEU points. This model was also able to overcome the limitations of the SOTA model BE-Encoder. mBART-large-50 can also be a good alternative for robust Hindi translation from English with 86.83 BLEU points.

6.2 FUTURE SCOPE

The improvement and good translated results from pre-trained model is a good start for future developments. Looking at the results of further training pre-trained transformer-based model, the future aim is to develop a hybrid X-former which will work as an ensemble technique having the capacity to work on the entire IITB dataset. A non-shared GPU will also be used in future to avoid low BLEU scores due to limited GPU as seen in the case of MADLAD-400. Several other metrics such as spBLEU, chrF, chrf++, and COMET will be used to cross validate the translation quality from hybrid model. For validation of model different English to Hindi dataset will be used like WMT2014, WMT2016. Another addition to future scope would be to check the comparison of dot product attention and additive attention. Where those two could be analysed on an experimental basis.

6.3 SOCIAL IMPACT

Globalization is at its peak. During this time translation among people becomes very essential to understand the context of each individual. A higher BLEU score can determine a good translation due to which several aspects like trade of goods from one country to another could be easily done through good communication. Translation model could help an individual to travel a new place without any discomfort in speaking a new language. Therefore, Translation is very important in today's world of globalization. BLEU score has been improved and the popularity of several other NLP task have also emerged only because of the release of vanilla transformer architecture. This encoder-decoder model helped to create a baseline for other researchers to develop variants of transformer called as X-former which has been discussed briefly in this study. Various pre-trained model was also fine-tuned to enhance the model and improve the BLEU further. Same has been discussed in this thesis in an in-depth experimental comparative analysis.

REFERENCES

[1] S. K. Sheshadri, D. Gupta, and M. R. Costa-Jussà, "A Voyage on Neural Machine Translation for Indic Languages," *Procedia Comput. Sci.*, vol. 218, pp. 2694–2712, 2023, doi: https://doi.org/10.1016/j.procs.2023.01.242.

[2] Y. Yuxiu, "Application of translation technology based on AI in translation teaching," *Syst. Soft Comput.*, vol. 6, p. 200072, 2024, doi: https://doi.org/10.1016/j.sasc.2024.200072.

[3] T. B. Brown *et al.*, "Language models are few-shot learners," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, in NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[4] X. Liu, K. Duh, L. Liu, and J. Gao, "Very Deep Transformers for Neural Machine Translation," *CoRR*, vol. abs/2008.0, 2020, [Online]. Available: https://arxiv.org/abs/2008.07772

[5] S. Takase and S. Kiyono, "Lessons on Parameter Sharing across Layers in Transformers," in *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing, SustaiNLP 2023, Toronto, Canada (Hybrid), July 13, 2023*, N. S. Moosavi, I. Gurevych, Y. Hou, G. Kim, Y. J. Kim, T. Schuster, and A. Agrawal, Eds., Association for Computational Linguistics, 2023, pp. 78–90. [Online]. Available: https://aclanthology.org/2023.sustainlp-1.5

[6] R. Huidrom and Y. Lepage, "Zero-shot translation among Indian languages," in *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, A. Karakanta, A. K. Ojha, C.-H. Liu, J. Abbott, J. Ortega, J. Washington, N. Oco, S. M. Lakew, T. A. Pirinen, V. Malykh, V. Logacheva, and X. Zhao, Eds., Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 47–54. [Online]. Available: https://aclanthology.org/2020.loresmt-1.7

[7] W. Moon, T. Kim, B. Park, and D. Har, "Enhanced Transformer Architecture for Natural Language Processing," in *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, C.-R. Huang, Y. Harada, J.-B. Kim, S. Chen, Y.-Y. Hsu, E. Chersoni, P. A, W. H. Zeng, B. Peng, Y. Li, and J. Li, Eds., Hong Kong, China: Association for Computational Linguistics, Dec. 2023, pp. 841–851. [Online]. Available: https://aclanthology.org/2023.paclic-1.84

[8] Rahul, P. Rawat, Vivek, and A. Elahi, "Abstractive Summarization on Dynamically Changing Text," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 1158–1163. doi: 10.1109/ICCMC51019.2021.9418438.

[9] A. Vaswani *et al.*, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, in NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.

[10] A. Bahrini *et al.*, "ChatGPT: Applications, Opportunities, and Threats," in *2023 Systems and Information Engineering Design Symposium (SIEDS)*, 2023, pp. 274–279. doi: 10.1109/SIEDS58326.2023.10137850.

[11] A. Gupta, Rahul, I. Khatri, and Monika, "A Review on Various Techniques of Automatic Text Summarization," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2020, pp. 1379–1384. doi: 10.1109/ICECA49313.2020.9297641.

[12] Rahul, S. Rauniyar, and Monika, "A Survey on Deep Learning based Various Methods Analysis of Text Summarization," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, 2020, pp. 113–116. doi:

10.1109/ICICT48043.2020.9112474.

[13] Rahul, V. Gupta, V. Sehra, and Y. R. Vardhan, “Hindi-English Code Mixed Hate Speech Detection using Character Level Embeddings,” in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 1112–1118. doi: 10.1109/ICCMC51019.2021.9418261.

[14] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The Long-Document Transformer,” *CoRR*, vol. abs/2004.0, 2020, [Online]. Available: https://arxiv.org/abs/2004.05150

[15] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai, “GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 4895–4901. doi: 10.18653/v1/2023.emnlp-main.298.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[17] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=H1eA7AEtvS

[18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *CoRR*, vol. abs/1910.0, 2019, [Online]. Available: http://arxiv.org/abs/1910.01108

[19] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 1, Jan. 2020.

[20] M. Lewis *et al.*, “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. doi: 10.18653/v1/2020.acl-main.703.

[21] W. Fedus, B. Zoph, and N. Shazeer, “Switch transformers: scaling to trillion parameter models with simple and efficient sparsity,” *J. Mach. Learn. Res.*, vol. 23, no. 1, Jan. 2022.

[22] B. Isik, N. Ponomareva, H. Hazimeh, D. Paparas, S. Vassilvitskii, and S. Koyejo, “Scaling Laws for Downstream Task Performance of Large Language Models,” *arXiv Prepr. arXiv2402.04177*, 2024.

[23] J. Gala *et al.*, “IndicTrans2: Towards High-Quality and Accessible Machine Translation Models for all 22 Scheduled Indian Languages,” *Trans. Mach. Learn. Res.*, 2023, [Online]. Available: https://openreview.net/forum?id=vfT4YuzAYA

[24] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, and P. Kumar, “IndicBART: A Pre-trained Model for Indic Natural Language Generation of Indic Languages,” in *ACL 2022*, May 2022. [Online]. Available: https://www.microsoft.com/en-us/research/publication/indicbart-a-pre-trained-model-for-indic-natural-language-generation-of-indic-languages/

[25] Y. Liu *et al.*, “Multilingual Denoising Pre-training for Neural Machine Translation,” *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 726–742, 2020, doi: 10.1162/tacl_a_00343.

[26] D. Kakwani *et al.*, “IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages,” in *Findings of EMNLP*, 2020.

[27] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, “RoFormer: Enhanced transformer with Rotary Position Embedding,” *Neurocomputing*, vol. 568, p. 127063, 2024, doi: https://doi.org/10.1016/j.neucom.2023.127063.

[28] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, “Linformer: Self-Attention with Linear Complexity,” *CoRR*, vol. abs/2006.0, 2020, [Online]. Available: https://arxiv.org/abs/2006.04768

[29] N. Kitaev, L. Kaiser, and A. Levskaya, “Reformer: The Efficient Transformer,” in *8th International Conference on Learning Representations, ICLR 2020,Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=rkgNKkHtvB

[30] A. Radford *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[31] Y. Li, Y. Shan, Z. Liu, C. Che, and Z. Zhong, “Transformer fast gradient method with relative positional embedding: a mutual translation model between English and Chinese,” *Soft Comput.*, vol. 27, no. 18, pp. 13435–13443, Nov. 2022, doi: 10.1007/s00500-022-07678-5.

[32] G. Ke, D. He, and T.-Y. Liu, “Rethinking Positional Encoding in Language Pre-training,” in *9th International Conference on Learning Representations, ICLR 2021,Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=09-528y2Fgf

[33] N. Shazeer, “Fast Transformer Decoding: One Write-Head is All You Need,” *CoRR*, vol. abs/1911.0, 2019, [Online]. Available: http://arxiv.org/abs/1911.02150

[34] S. Dou *et al.*, “LoRAMoE: Alleviate World Knowledge Forgetting in Large Language Models via MoE-Style Plugin,” 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266335873

[35] Y. Zhao, S. Zhou, and Z. Zhang, “Multi-split Reversible Transformers Can Enhance Neural Machine Translation,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds., Online: Association for Computational Linguistics, Apr. 2021, pp. 244–254. doi: 10.18653/v1/2021.eacl-main.19.

[36] Z. Zheng, H. Zhou, S. Huang, J. Chen, J. Xu, and L. Li, “Duplex Sequence-to-Sequence Learning for Reversible Machine Translation,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., Curran Associates, Inc., 2021, pp. 21070–21084. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/afecc60f82be41c1b52f6705ec69e0f1-Paper.pdf

[37] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” in *Computer Vision -- ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 630–645.

[38] S. Takagi, “On the Effect of Pre-training for Transformer in Different Modality on Offline Reinforcement Learning,” in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=9GXoMs__ckJ

[39] J. Libovický and A. Fraser, “Towards Reasonably-Sized Character-Level Transformer NMT by Finetuning Subword Systems,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational

Linguistics, Nov. 2020, pp. 2572–2579. doi: 10.18653/v1/2020.emnlp-main.203.

[40] A. Chhetri and U. Sharma, “Survey on Detection of Face Mask and Social Distancing using Tensorflow,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2022, pp. 1022–1027. doi: 10.1109/ICACITE53722.2022.9823851.

[41] J. Kasai *et al.*, “Finetuning Pretrained Transformers into RNNs,” *CoRR*, vol. abs/2103.1, 2021, [Online]. Available: https://arxiv.org/abs/2103.13076

[42] G. Paaß and S. Giesselbach, “Improving Pre-trained Language Models,” in *Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media*, Cham: Springer International Publishing, 2023, pp. 79–159. doi: 10.1007/978-3-031-23190-2_3.

[43] A. Fan *et al.*, “Beyond english-centric multilingual machine translation,” *J. Mach. Learn. Res.*, vol. 22, no. 1, Jan. 2021.

[44] S. Kudugunta *et al.*, “MADLAD-400: A Multilingual And Document-Level Large Audited Dataset,” in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper%5C_files/paper/2023/hash/d49042a5d49818711c401d34172f9900-Abstract-Datasets%5C_and%5C_Benchmarks.html

[45] Y. Tang *et al.*, “Multilingual Translation with Extensible Multilingual Pretraining and Finetuning,” *CoRR*, vol. abs/2008.0, 2020, [Online]. Available: https://arxiv.org/abs/2008.00401

[46] J. Tiedemann *et al.*, “Democratizing neural machine translation with OPUS-MT,” *Lang. Resour. Eval.*, 2023, doi: 10.1007/s10579-023-09704-w.

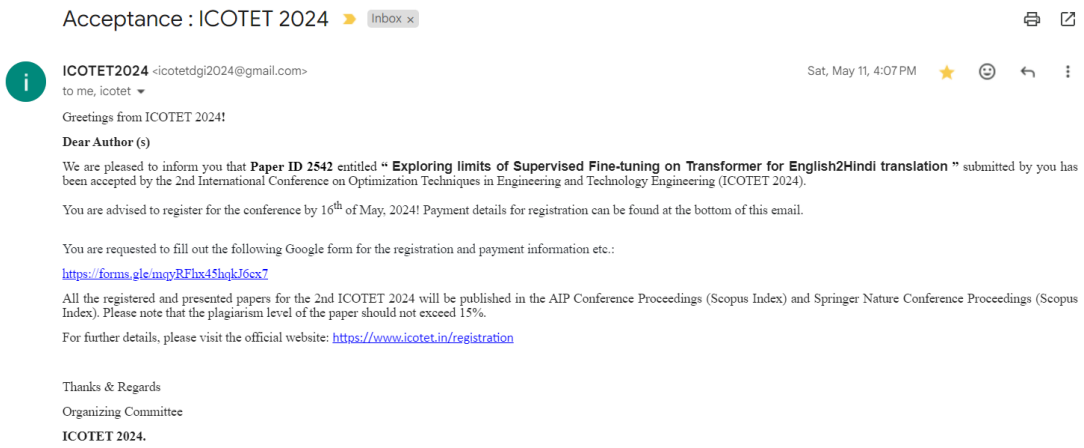
[47] M. Bansal and D. K. Lobiyal, “Context-based Machine Translation of English-Hindi using CE-Encoder,” *J. Comput. Sci.*, vol. 17, no. 9, pp. 827–847, Sep. 2021, doi: 10.3844/jcssp.2021.827.847.

[48] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, in ACL ’02. USA: Association for Computational Linguistics, 2002, pp. 311–318. doi: 10.3115/1073083.1073135.

[49] S. Saini and V. Sahula, “A novel model based on Sequential Adaptive Memory for English–Hindi Translation,” *Cogn. Comput. Syst.*, vol. 3, no. 2, pp. 142–153, Mar. 2021, doi: 10.1049/ccs2.12011.

LIST OF PUBLICATIONS

[1] Anmol Chhetri, Rahul, "Exploring limits of Supervised Fine-tuning on Transformer for English2Hindi translation". The paper has been **Accepted** at the 2nd International Conference on Optimization Techniques in Engineering and Technology Engineering (**ICOTET 2024**), June 2024. Indexed by **Scopus**. Paper Id: 2542.



Acceptance : ICOTET 2024 Inbox ×

ICOTET2024 <icotetdgi2024@gmail.com> Sat, May 11, 4:07 PM

to me, icotet

Greetings from ICOTET 2024!

Dear Author (s)

We are pleased to inform you that **Paper ID 2542** entitled **" Exploring limits of Supervised Fine-tuning on Transformer for English2Hindi translation "** submitted by you has been accepted by the 2nd International Conference on Optimization Techniques in Engineering and Technology Engineering (ICOTET 2024).

You are advised to register for the conference by 16th of May, 2024! Payment details for registration can be found at the bottom of this email.

You are requested to fill out the following Google form for the registration and payment information etc.:

https://forms.gle/mqyRFhx45hqkJ6cx7

All the registered and presented papers for the 2nd ICOTET 2024 will be published in the AIP Conference Proceedings (Scopus Index) and Springer Nature Conference Proceedings (Scopus Index). Please note that the plagiarism level of the paper should not exceed 15%.

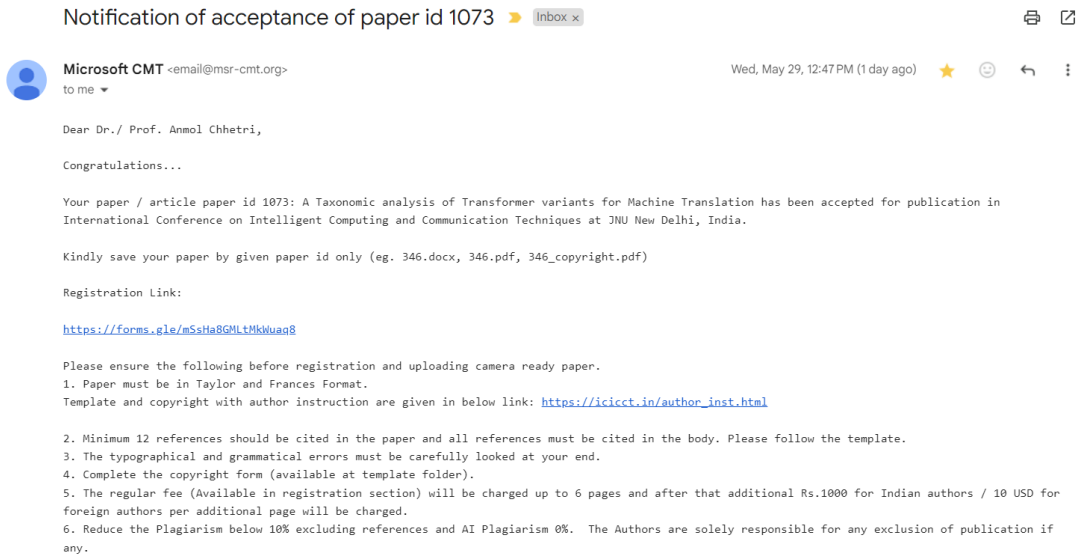
For further details, please visit the official website: https://www.icotet.in/registration

Thanks & Regards

Organizing Committee

ICOTET 2024.

[2] Anmol Chhetri, Rahul, "A Taxonomic analysis of Transformer variants for Machine Translation". The paper has been **Accepted** at the International Conference on Intelligent Computing and Communication Techniques (**ICICCT 2024**), June 2024. Indexed by **Scopus**. Paper Id: 1073.



Notification of acceptance of paper id 1073 Inbox ×

Microsoft CMT <email@msr-cmt.org> Wed, May 29, 12:47 PM (1 day ago)

to me

Dear Dr./ Prof. Anmol Chhetri,

Congratulations...

Your paper / article paper id 1073: A Taxonomic analysis of Transformer variants for Machine Translation has been accepted for publication in International Conference on Intelligent Computing and Communication Techniques at JNU New Delhi, India.

Kindly save your paper by given paper id only (eg. 346.docx, 346.pdf, 346_copyright.pdf)

Registration Link:

https://forms.gle/mSsHa8GMLtMkWuaq8

Please ensure the following before registration and uploading camera ready paper.
1. Paper must be in Taylor and Frances Format.
Template and copyright with author instruction are given in below link: https://icicct.in/author_inst.html

2. Minimum 12 references should be cited in the paper and all references must be cited in the body. Please follow the template.
3. The typographical and grammatical errors must be carefully looked at your end.
4. Complete the copyright form (available at template folder).
5. The regular fee (Available in registration section) will be charged up to 6 pages and after that additional Rs.1000 for Indian authors / 10 USD for foreign authors per additional page will be charged.
6. Reduce the Plagiarism below 10% excluding references and AI Plagiarism 0%. The Authors are solely responsible for any exclusion of publication if any.