Hinglish Language Auto-completion using T5 models
Final Project Report - EEL 6812

Jitesh Parapoil [* 1] **Mukund Dhar** [* 1 2] **Chetan Choudhary** [* 1 3]

Abstract

Our project aims to further Hinglish language modeling by creating a unique auto-completion system designed exclusively for Hinglish, a Hindi-English hybrid popular in urban India. Using accessible Hinglish datasets and advanced sequence models such as T5, we solve Hinglish's particular problems, such as code-switching and orthographic differences. Our solution uses data preprocessing and model adaptation techniques to give accurate and contextually relevant auto-completion suggestions. Our research adds to the improvement of Hinglish language processing capacities by filling a significant vacuum in existing literature for auto-completion using the advanced T5 model and providing insights into specific modeling methodologies, allowing for clearer communication and empowering Hinglish-speaking populations.

1. Introduction

In a world of constantly evolving landscape of communication, the contextuality and clarity of expression remain paramount. Various linguistic landscapes have seen the fusion of languages emerge as a hallmark of modern expression, reflecting the dynamism of linguistic evolution. One such fusion that has become popular and has taken place, especially in urban India, is Hinglish— a vibrant fusion of Hindi and English that puts together elements from both languages to create a distinct and expressive mode of communication.

The popularity of Hinglish reflects its importance as a mode of communication for millions of people facing the challenges of a multicultural and multilingual society. However, the development of language technologies tailored for Hinglish remains relatively nascent. Among the many language technologies, auto-completion systems are one of the most efficient in upgrading the text-entry process and aiding smooth communication through digital platforms.

In this light, our research aims to fill this void by introducing an innovative auto-completion system tailored exclusively for Hinglish. Unlike traditional auto-completion systems that are generally tailored to monolingual languages, ours is uniquely geared to face the challenges of Hinglish, a language that flows freely between Hindi and English and reflects its socio-cultural nuances.

Our approach is primarily based on fine-tuning the advanced T5 models, using Hinglish-specific corpora to capture the intricate patterns and context-specific word relationships that are inherent in Hinglish. We also added data-preprocessing steps to work on the available dataset on Hinglish to suit our needs for the task.

Thus, our project aims to contribute to the development of more sophisticated capabilities for processing of the Hinglish language, promoting cross-cultural communication, and empowering Hinglish-speaking communities in this increasingly connected world.

2. Related Works

Recently published works have deep-dived into the aspects of Hinglish code-switching and auto-completion text generation in detail. These works highlight the challenges and opportunities involved in the field and have further contributed to the development of a better auto-completion system tailored for Hinglish.

Agarwal et al. (2022)(1), presented CST5 a dataset augmentation framework for code-switched semantic parsing in languages such as Hinglish. With the augmented dataset, researchers have focused on the nuances of the Hinglish code-switching phenomenon and the impact of code-switching on auto-completion systems.

Srivastava and Singh (2021)(2) proposed the HinGE dataset, specifically tailored for the generation and evaluation of code-mixed text in the Hinglish language. The dataset sets the foundation for auto-completion systems for Hinglish.

Kulkarni et al. (2022)(3) presented research on LSTM-based models for next-word prediction tasks in Hinglish text generation, where these models could capture the dependencies and linguistic structures inherent in code-switched languages. Though LSTM models are beginning to show promising results, their limitations in handling the complexities of Hinglish code-switching pose a research challenge.

Following in the same line of work, **Lewis et al. (2019)**(4) presented research on transformer-based language models, such as BART, for various text generation tasks, including auto-completion in languages with code-switching phenomena. Though these models have dramatically improved in capturing dependencies across long distances and contextual connections, there is still a challenge in adapting them to the unique code-switching of Hinglish.

In the same line of work, **Ambulgekar et al. (2021)**(6) focused on RNN-based models for next-word prediction in Hinglish text generation, highlighting the importance of recurrent neural networks in capturing the sequential nature of code-switching languages. However, the generalizability of RNN-based models for this unique linguistic characteristic of Hinglish still presents a challenge to further research.

Together, they demonstrate a burgeoning interest and research interest in this area, that is, the understanding of code-switching and text generation for auto-completion. Our research pivoted off these works and synthesised insights by the application of advanced models in further developing the auto-completion systems tailored for Hinglish for better service to this purpose.

3. Methodology

Many auto-completion systems are present currently and are used extensively in the majority of applications. One big use case is completing the next word while writing emails by understanding the context of the words typed earlier. All of the major auto-completion systems have been explored in English and other popular languages, but the adaptation of such technology to Hinglish has not been studied yet. We present a model tailored specifically for the Hinglish language and its nuances. We do so by leveraging the capabilities of the T5 model, which has demonstrated remarkable proficiency in natural language processing tasks. We aim to address this unexplored domain and contribute to the advancement of multilingual text prediction systems.

3.1. Data Preprocessing

Firstly, we addressed various data preprocessing tasks to ensure the quality of the dataset. We handled punctuation marks and removed any extra white spaces to streamline the text. Additionally, all numerical values were converted into their corresponding word representations. The approach used by NLP requires the data to be in the text and the output is also a text. After this, we cleaned noisy data and eliminated irrelevant content, thereby enhancing the overall performance of the model.

Furthermore, the Google Hinglish TOP dataset consisted of translations from English to Hinglish. We extracted the translations and structured them into two distinct columns: input_text and target_text Subsequently, we devised two processes for generating input-target pairs. In the first process, individual words in the input text were paired with their subsequent word in the target text. Conversely, the second process involved grouping words into sentences, with the target text containing the next word in the sequence.

To facilitate efficient handling of input and target text pairs, a dedicated class was designed. This custom class utilized the T5 tokenizer obtained from the Hugging Face transformers library. Each sample within the dataset underwent preprocessing, incorporating a task prefix (”predict the next word: ”) to signify the objective of predicting subsequent words in a sequence. The tokenized input and target texts were subsequently encoded into token IDs, We ensured uniform sequence length by applying padding and truncation to the created tokens.

3.2. Modeling

The Text-To-Text Transfer Transformer (T5) model (5) is a versatile and powerful neural network architecture introduced by Google Research in 2019. The T5 model integrates all the natural language processing tasks as one, it does not consider every task to be different. T5 uses a text-to-text unified approach where it takes the input as text and gives the output as text too even if it is a number.

We selected the T5 model as it offers several advantages. Its ability to handle text-to-text transformations allows us to frame the auto-completion task as a natural language generation problem, where the model predicts the next word in a sequence given the preceding context. Additionally, T5's effectiveness in capturing linguistic nuances makes it a suitable choice for handling the complexities of Hinglish. By fine-tuning a pre-trained T5 model on Hinglish-specific data, we can leverage its powerful representations to build an auto-completion system that effectively suggests relevant words in Hinglish text.

The adaptation of our model involved fine-tuning the T5 models in both its base (T5-base) and small (T5-small) variants, on our Hinglish dataset. This process aimed to help the model get used to the special language features of Hinglish, making it better at suggesting the right words.

Incorporating transfer learning techniques, we leveraged pre-trained models as a foundation and then fine-tuned them on Hinglish data to further refine their capabilities. By harnessing the knowledge encoded within pre-trained models and tailoring them to the intricacies of Hinglish, we sought to optimize the performance of our auto-completion system for Hinglish text.

4. Experiments and Results

The models are variants of the Text-To-Text Transfer Transformer (T5) architecture, developed by Google, renowned for its versatility in natural language processing tasks. This report outlines experiments designed to evaluate and compare the performance of these models across various dimensions including training efficiency, evaluation accuracy along with inference quality and speed.

4.1. Experimental Setup:

The experiments were conducted using the same configuration to ensure a fair comparison. Both models were trained and evaluated on the same dataset, with identical hyperparameters. The training dataset was split into batches of size 4, and each model was trained for 5 epochs using the Adam optimizer with a learning rate of 1e-4. Additionally, a separate testing dataset was used to assess the models' performance using perplexity scores. Inference tests were performed on sample inputs to compare the models' output quality and generation speed.

4.2. Experiment 1: Training Performance

In this experiment, we trained both the T5-small and T5-base models on a common dataset for a fixed number of epochs. The training losses were recorded and compared across epochs to assess the convergence behavior and relative training efficiency of the two models. Additionally, we monitored the training time for each epoch and the overall duration of training to evaluate the computational cost associated with training each model. This experiment aimed to provide insights into the training dynamics and efficiency of the T5-small and T5-base models. The Figure 1 and Figure 2

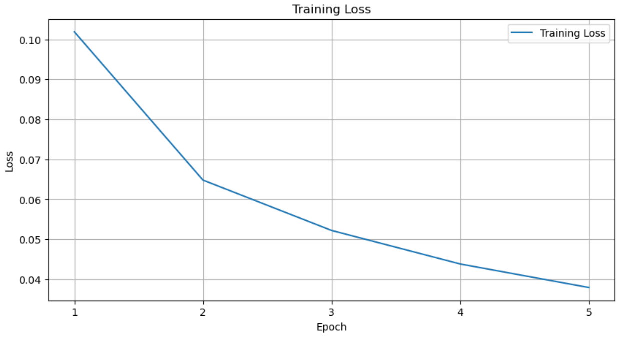

Figure 1. Training Loss Plot for T5-base

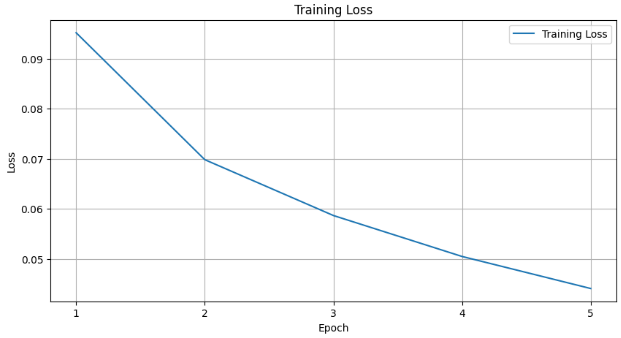

Figure 2. Training Loss Plot for T5-small

Table 1. Evaluation metrics Comparison

Model	Perplexity Score	Inference Speed (s)
T5-base	1.06	0.3106
T5-small	1.07	0.186

4.3. Experiment 2: Evaluation Performance

For evaluation, both models were assessed using a separate test dataset to calculate the perplexity scores shown in Table 1. The perplexity scores served as a metric to quantify the performance of the models in predicting the next word in the sequence. By comparing the perplexity scores obtained for T5-small and T5-base models, we aimed to gauge their relative performance in terms of language modeling capabilities.

4.4. Experiment 3: Inference Performance

In this experiment, the inference performance of both models was evaluated on sample inputs as shown in Table 2. The time taken for generating outputs was recorded to assess the inference speed of each model, shown in Table 1. Furthermore, the quality of generated outputs was qualitatively compared to identify any differences in the model's ability to produce coherent and contextually relevant responses. This experiment aimed to provide insights into the efficiency and effectiveness of the T5-small and T5-base models in generating outputs for given input texts.

5. Conclusion

In this project, we conducted the development of auto-completion systems specifically curated and trained in the Hinglish language. The project focuses on the need for efficient and intuitive text entry systems in today's world, where continuous communication is important. We attempted to

Table 2. Comparison of Auto-Completion for the given inputs

INPUT	T5-BASE	T5-SMALL
KIS COLLEGE KA	ADMISSION	NAAM
AAP MUJHE	KAL	BATAYE
KYUN TIME	PAR	SE
NINE THIRTY	PM	BAJE
KAISE HO	RAHE	AAP
MUJHE RESTAURANT KA	FOOD	MERA

address Hinglish's particular issues, such as code-switching and orthographic differences, by utilizing public Hinglish datasets and advanced sequence models such as T5-base and T5-small models.

We also performed extensive experimentation and evaluation to highlight the effectiveness and scope of our approach toward the enhancement of auto-completion suggestions for text in Hinglish. Through different data preprocessing techniques and model adaptation strategies, we have tried to ensure increased accuracy and relevance in auto-completion suggestions and hence facilitate more coherent communication and empower Hinglish-speaking communities.

Although our work represents a significant step toward modeling Hinglish auto completion, much still needs to be done. Future work includes refinements of existing models, a look at ensemble learning techniques, and increased coverage toward more expansive Hinglish datasets for better efficiency and generalization capacity of our models. Limitations that need to be addressed are uncovered rare word occurrences and out-of-vocabulary words.

In conclusion, our research underscores the potential for further advancements in auto-completion systems tailored for Hinglish, signalling a pivotal moment in the evolution of linguistic technologies. By harnessing the power of advanced sequence models and innovative methodologies, we have laid a solid foundation for future endeavors aimed at enhancing communication clarity and efficiency in Hinglish-speaking communities. Moving forward, collaborative efforts across interdisciplinary domains will be instrumental in driving continued innovation and fostering inclusive communication solutions for diverse linguistic contexts.

6. Future Work

While our present work represents another step forward in the development of AutoCompletion systems for Hinglish, there are several avenues for future research and improvement. One such research line is the fine-tuning and optimization of the T5 base model and the T5-small models. Though these models effectively capture sequential dependencies and linguistic patterns, still a lot of room is left for fine-tuning and experimentation to achieve better performance on Hinglish-specific tasks.

Apart from this, the integration of hardware acceleration techniques, namely GPU optimization, will drastically reduce the time to train and make inference, allowing real-time auto-completion on any digital device that has poor computational power. The use of ensemble learning techniques, where multiple models are integrated, will smooth out individual model biases and increase the accuracy of the prediction, thus giving better performance.

Future developments of more advanced language models, tailored for Hinglish, and the integration of attention mechanisms and contextual embeddings will have very promising possibilities for obtaining more nuanced and contextually relevant suggestions in the Auto-Completion task. A larger Hinglish dataset, and the development of benchmark datasets especially tailored for Auto-Completion tasks, will be very instrumental in training and evaluating models in the future.

Though our current approach still suffers from limitations such as the ability to handle rare word occurrences and words out-of-vocabulary, these are still very important areas to work on. The implementation of robust exception handling mechanisms and the exploration of subword tokenization will go a great way in improving the robustness and generalization capabilities of our models, ensuring accurate and contextually coherent Auto-Completion suggestions across a wide variety of situations.

The future of auto-completion systems for Hinglish will be in perfecting and optimizing the existing models, introducing novel techniques and methodologies, and collaborative efforts for expanding and improving Hinglish-specific datasets. Thus, still moving forward with enhancing NLP and machine learning, the aim is to improvise the auto-completion systems further and increase efficiency and effectiveness in the interest of millions of users engaged in communication with Hinglish in the digital world.

Acknowledgements

We'd like to thank Dr. Shahana Ibrahim, for being our mentor and guiding us throughout the duration of the class as well as this project. The mix of abilities and knowledge shared in the class greatly broadened our understanding of neural networks and deep learning. This collaborative learning atmosphere helped us to implement advanced deep learning and learn how to improve on existing methods. We are grateful to be a part of the class that enabled us to learn in a rich academic atmosphere, and we plan to implement what we have learned in the future.

References

[1] Agarwal, A., Gupta, J., Goel, R., Upadhyay, S., Joshi, P., & Aravamudhan, R. (2022). CST5: Data Augmentation for Code-Switched Semantic Parsing. *arXiv preprint arXiv:2211.07514*. Retrieved from https://arxiv.org/abs/2211.07514

[2] Srivastava, V., & Singh, M. (2021). HinGE: A Dataset for Generation and Evaluation of Code-Mixed Hinglish Text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems* (pp. 200–208). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.eval4nlp-1.20.pdf

[3] Shreyas Mhatre, Sarang Joshi, and Hrushikesh B. Kulkarni. (2022). Sign Language Detection using LSTM. In *2022 IEEE International Conference on Current Development in Engineering and Technology (CCET)*, pp. 1-6. DOI: https://doi.org/10.1109/CCET56606.2022.10080705

[4] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint*. Retrieved from https://arxiv.org/abs/1910.13461

[5] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2023). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv preprint arXiv:1910.10683*. Retrieved from https://arxiv.org/abs/1910.10683

[6] Sourabh Ambulgekar, Sanket Malewadikar, Raju Garande, and Bharti Joshi. (2021). Next Words Prediction Using Recurrent Neural Networks. *ITM Web of Conferences*. Retrieved from https://api.semanticscholar.org/CorpusID:238952028