Neural Machine Translation From English To Dravidian languages Using MT5 Model

Vishnu Choundur
U00869233
University of Memphis
vchundur@memphis.edu

Vishnu Teja Yerrapragada
U00869711
University of Memphis
vyrrprgd@memphis.edu

Naresh Bolishetty
U00868540
University of Memphis
nblshtty@memphis.edu

Abstract

Despite significant advancements in Neural Machine Translation (NMT) over the past few decades, the quality of translation for Indic languages remains less satisfactory, largely due to the scarcity of parallel training data. In this paper, we present our approach to limit the availability of such data for Indic languages, particularly between English to Dravidian Languages. This study explores the application of the multilingual Text-To-Text Transfer Transformer (mT5) model for NMT between English and two Dravidian Languages: Telugu and Tamil. Due to the linguistic diversity and structural differences between these languages and English, traditional translation models often struggle with accuracy and fluency. The mT5 model, known for its proficiency in multiple languages, offers a promising solution by leveraging a pre-trained, multilingual corpus that includes low-resource languages. We detail our methodology for fine-tuning the mT5 model specifically for English to Telugu and English to Tamil translation tasks. We evaluate the performance of the fine-tuned mT5 model against baseline models using standard NMT metrics such as BLEU.

1 Introduction

The challenge of machine translation (MT) between languages with significant structural and lexical differences, such as between English and Dravidian languages like Telugu and Tamil, remains a focal area of research in computational linguistics. These languages, originating from the South of India, are characterized by rich morphological structures and a syntax that is markedly different from Indo-European languages. The complexity is compounded by the limited availability of high-quality, parallel corpora, making it challenging for traditional machine translation systems to achieve high accuracy and naturalness in translations (Chakravarthi and Muralidaran, 2021) . We have 76 Dravidian languages. Among these 4 languages are majorly spoken in India such as Telugu, Malayalam, Kannada, Tamil. Malayalam, Telugu and Kannada are derived from the Tamil. This paper presents the development of NMT systems from English to Dravidian languages such as Tamil and Telugu.

The south India Languages are Low resources languages. The translation problem for South Indian Languages (Rajendran and Cn, 2019):

- **Limited Data Availability:** The most significant issue is the scarcity of digital text data. Many low-resource languages have limited written content available online, few digitized books, and minimal presence in digital media. This lack of data hinders the training of robust language models.
- **Lack of Annotated Corpora:** Beyond sheer volume, there is often a lack of annotated corpora (texts annotated for linguistic features such as part-of-speech, syntax, and semantics) which are crucial for training more sophisticated models like parsers or entity recognizers.
- **Orthographic Challenges:** Many low-resource languages have writing systems that are not well-supported by modern technology. This includes non-standard alphabets, lack of uniform spelling conventions, and extensive use of diacritics.
- **Technological Infrastructure:** Limited access to technology, such as the internet and computing resources, can hinder the development of language technologies in regions where low-resource languages are spoken. This also affects the ability to gather data and deploy language-based applications.
- **Multilingualism and Code-Switching:** Many low-resource language communities

Figure 1: Tamil Alphabets

are inherently multilingual, and code-switching is common. Capturing the nuances of mixed-language texts requires models that can handle multiple languages simultaneously, which can be particularly challenging without adequate data.

- **Financial and Institutional Support:** Often, there is limited commercial interest in developing technologies for low-resource languages, which affects funding and resource allocation. Academic and governmental support can also be sparse, depending on the region.

- **Cultural and Social Barriers:** There might be cultural sensitivities and social dynamics that complicate data collection and technology deployment. For instance, there may be resistance to digitizing certain cultural knowledge or skepticism towards technological solutions from outside the community.

Tamil is a classical Dravidian language predominantly spoken in the Indian state of Tamil Nadu and the northern and eastern regions of Srilanka. It is also official Language in Singapore and has significant speakers in Malasia, Mauritius, Fiji. It has existed for over 2,000 years, with its script evolving from the ancient Brahmi script around the 3rd century BCE. The modern Tamil script, an abugida, is characterized by rounded letters adapted for writing on palm leaves, comprising 12 vowels and 18 consonants that combine to form numerous compound characters. This script, which beautifully encapsulates the phonetic features of the language by omitting distinctions between voiced and unvoiced consonants, is central not only to the Tamil

கர்மா	கங்கா	ராகுல்
karma	Ganga	Rahul
Ka	Ga	Ha

Figure 2: Multiple Sounds for single Letter 'ka'

language but has also been historically used for Sanskrit and other regional languages.

Normally, In English we have 26 Alphabets, which comprise of 44 phonetics sounds. but Tamil, we have 12 vowels and 18 consonants, which can comprise of 12*18 and 12*4 combinations. But, In Tamil, some alphabets can produce dual sounds. We can see the Image Figure 2, how a single letter produce multiple character sounds.

Telugu, a Dravidian language, is predominantly spoken in the Indian states of Andhra Pradesh and Telangana, where it is the official language. It features a syllabic script derived from the Brahmi script, known for its rounded characters suitable for carving on stone and palm leaves, comprising 16 vowels and 36 consonants. Telugu's script is phonetic, representing each sound distinctly, which makes it easy to learn.

Generally, In Telugu, we have 16 vowels and 36 consonants, which comprise of 16*36 combinations. It is very easy to map a sound with word. In Telugu, each letter have unique sound. In spite of having less alphabet, Tamil have multiple sounds for some letter.It will be very difficult to map it. So, Telugu can train easily when compared to Tamil (S et al., 2023).

1.1 Objectives

- **Address Limited Resources:** The primary goal is to develop a viable machine translation system that operates effectively with the limited linguistic resources available for Dravidian languages. Innovation in data synthesis and augmentation techniques is crucial to overcome the scarcity of digital language data.

- **Focus on Dravidian Languages:** Targeting English-to-Dravidian language translation, the project aims to create systems that are customized for the translation nuances of Tamil, Telugu, Kannada, and Malayalam. These systems are designed to comprehend and translate the complex grammatical struc-

Figure 3: Telugu Alphabets

tures unique to these languages.

- **Experiment with Pre-trained and Fine Tuning:** The approach leverages pre-trained NMT models as a foundational base, applying fine-tuning methods to adapt these models to the specific linguistic features of Dravidian languages. This process includes techniques that augment limited monolingual data to improve model performance.
- **Evaluate with BLEU Scores:** To measure the effectiveness of the translation models, BLEU scores are used as the standard metric. This provides a quantifiable assessment of translation accuracy, allowing for comparative analysis across different language pairs and setting benchmarks for the translation quality.

1.2 Motivation

The motivation for employing Neural Machine Translation (NMT) from English to Dravidian languages such as Tamil and Telugu using the mT5 model is deeply rooted in the need to bridge linguistic divides and enhance digital inclusivity. Dravidian languages, spoken by millions, often suffer from inadequate representation in digital and technological realms. This limitation restricts access to information, educational resources, and equal participation in the digital economy for these language speakers. The mT5 model, renowned for its effectiveness due to extensive pre-training across multiple languages, offers an advanced solution. Its capability to understand and generate text in over 100 languages makes it particularly suitable for handling the intricate linguistic features of Dravidian languages, which include complex agglutinative grammar and diverse phonetic systems. Utilizing the mT5 model for translating between English and these languages promises not only to break communication barriers but also to foster cultural preservation, educational access, and broader socio-economic opportunities for Tamil and Telugu speakers. This initiative thus supports the broader goal of creating more equitable technology access and promoting multicultural integration.

2 Related Work

Neural Machine Translation Using Sequence to Sequence: This is a traditional models which involved sequence to sequence learning (Seq2Seq)(Sutskever et al., 2014). Seq2Seq is an encoder-decoder approach, in which an encoder will read the input sentence, to produce the a hidden vector and the decoder produce the output sequence from the vector received from the encoder. A variety of RNN's have been proposed to enhance the translation from English to Indic Languages. Both Long-short term memory (LSTM)(Hochreiter and Schmidhuber, 1997), Recurrent Neural network (RNN)(Rezk et al., 2020) are designed to process sequential data and utilize a recurrent structure where the output from previous step is fed into current step. They are used extensively for tasks that involve sequential input such as Natural Language Processing (NLP). These architectures will learn through weights and bias that are adjusted during training using back-propogation. Simple models such as LSTM, Bi-RNN have been implemented for various languages from English to Tamil, Malayalam, Hindi, Punjabi with 200 and 500 hidden layers. English-Punjabi have achieved the good BLEU (27 for 500 layers) score when compared to remaining languages(Premjith et al., 2019).

Neural Machine Translation Using Transformer: The Transformer model(Vaswani et al., 2017), prominently used for machine translation, revolutionizes sequence processing through its unique architecture that separates into an encoder and a decoder, both employing self-attention mechanisms. This design allows it to process words in parallel, significantly enhancing training speed

and efficiency. Unlike RNNs, Transformers do not process data sequentially, which aids in managing long-range dependencies within text more effectively. By integrating positional encodings, Transformers maintain word order information, crucial for understanding and generating coherent translations. This architecture not only surpasses previous models in speed and scalability but also improves translation quality by effectively capturing complex linguistic nuances across different languages. Multilingual Bidirectional Encoder Representations from Transformers (mBERT) Model(Jain et al., 2020) is used, Instead of developing monolingual language models from the ground up without reusing any parameters from previously trained models, there is evidence suggesting that this approach may be necessary for some languages. For instance, research indicated by citation demonstrates that Multilingual BERT (mBERT) does not provide high-quality representations for every language. Specifically, mBERT tends to under perform compared to non-BERT models on various tasks for the languages that make up the lowest 30 percent in terms of dataset size used for training mBERT. Therefore, assessing language models in a monolingual context for low-resource languages is deemed crucial.

Neural machine Translation Using Back Translation: Neural Machine Translation (NMT) utilizing back translation(Kandimalla et al., 2022) is an effective technique to improve translation quality, especially when dealing with limited parallel corpora. In this method, a model first translates text from a source language to a target language (forward translation). This translated output is then translated back into the original language (back translation). The back-translated texts serve as additional training data, helping the model learn to produce more natural and accurate translations. By iteratively refining translations through this dual process, NMT systems can enhance their understanding of linguistic nuances and context, thereby improving overall translation performance even in low-resource language pairs. This technique leverages the strengths of neural networks' capacity for pattern recognition and generation, making it a powerful tool for boosting translation accuracy in diverse language applications.

Neural machine Translation Using Back Translation and Fine-Tuning: Neural Machine Translation (NMT) enhanced by back translation and fine-tuning (Vyawahare et al., 2022) is a sophisticated approach aimed at refining translation accuracy in scenarios with sparse data. Initially, the NMT system translates texts from the source to the target language, which are then translated back to the source language to create synthetic parallel texts. This augmented dataset helps in addressing the scarcity of training data. Subsequently, fine-tuning is employed, where the NMT model is further trained on a smaller, high-quality dataset to adapt more closely to specific linguistic nuances and domain-specific terminology. This combined methodology not only broadens the model's exposure to varied linguistic structures but also sharpens its proficiency in producing precise and contextually appropriate translations, particularly beneficial for under-resourced languages or specialized domains.

Neural Machine Translation using mT5 and fine-tuning: Neural Machine Translation (NMT) using the mT5 model(Jha et al., 2023) represents a state-of-the-art approach in translating text between multiple languages, including low-resource pairs. The mT5 model is a part of the Text-To-Text Transfer Transformer (T5) family(Mastropaolo et al., 2021), designed to handle a variety of natural language processing tasks by converting all tasks into a text-to-text format. This model is pre-trained on a large multilingual corpus(Vyawahare et al., 2022) and can be fine-tuned on specific language pairs or translation tasks to achieve superior performance. Its architecture leverages the Transformer's attention mechanisms, allowing it to effectively manage dependencies and nuances in language without the constraints of traditional RNN-based models. By fine-tuning mT5 on targeted datasets, it becomes highly effective in providing accurate, contextually appropriate translations, making it a powerful tool for global communication and content accessibility across different languages.

3 Data

Samanantar v0.3 is a comprehensive parallel corpora collection for Indic languages, boasting about 49.6 million sentence pairs spanning languages such as Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, and Telugu. It is meticulously structured into directories with pre-existing and newly mined data, provided in a randomly shuffled, untokenized,

and deduplicated format. This release also includes LaBSE scores and benchmarks like the Semantic Textual Similarity test sets. Released under the Creative Commons CC0 license by AI4Bharat, Samanantar v0.3 is available for download from several sources including direct links and Google Drive mirrors, making it an invaluable resource for enhancing multilingual NLP technologies AI4Bharat .

Figure 4: Tamil Vs English Data PairSet

Dataset	ta	en
Train	5,264,867	5,264,867
Validation	1,000	1,000
Test	2,390	2,390

Table 1: Dataset sizes for Tamil (ta) and English (en)

The Telugu-English and Tamil-English datasets within Samanantar v0.3 are crucial components for developing machine translation systems between manguages and English. Both datasets comprise millions of sentence pairs, curated to enhance language processing capabilities. They are presented in an untokenized, deduplicated, and randomly shuffled format, which is optimal for training robust language models. Each dataset pair set contain the train, validation, test dataset.

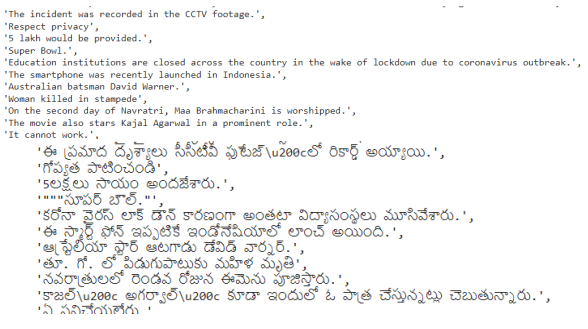

Figure 5: Telugu Vs English Data PairSet

From the Figure 4 and Figure 5, we have enormous number of sentence pair. The Table 1, it have approximately 5.26 Million sentence pair and Table 2, it have approximately 4.9 Million sentence pair. In this project, we have implement the model with one Million senetence pair for both english-tamil and english-telugu.

Dataset	te	en
Train	4,946,035	4,946,035
Validation	1,000	1,000
Test	2,390	2,390

Table 2: Dataset sizes for Telugu (te) and English (en)

4 Methodology

The methodology for the NMT system using the mT5 model for translating from English to Tamil and Telugu involves several detailed steps, starting with the setup and preparation of the environment. This includes installing essential Python libraries such as transformers, sentencepiece, datasets, and sacrebleu, which are crucial for model handling, data processing, and evaluation. The system leverages Google Colab for computation, where the script begins by mounting Google Drive to access stored datasets.

Data is meticulously prepared by specifying paths and loading the English-Tamil and English-Telugu sentence pairs into a structured format using a custom function. This ensures correct alignment and consistency between the datasets. The mT5 model Figure 6 and its tokenizer are loaded from Hugging Face's Transformers library, chosen for its proficiency in multilingual translation tasks. If a GPU is available, it is utilized to enhance processing speed and efficiency.

For model training, the script converts datasets into Pandas DataFrames to facilitate manipulation and processes texts into tensor formats suitable for the model. A generator function is crafted to handle batch preparation, and the training setup includes configuring the AdamW optimizer and a learning rate scheduler with warmup steps to optimize the model's learning process.

The training loop is executed over specified epochs, with real-time loss tracking and periodic outputs to monitor progress and adjust parameters as necessary. Post-training, functions for saving and loading the model and tokenizer are implemented to ensure the model can be reused without retraining from scratch.

Evaluation is a critical component, where the

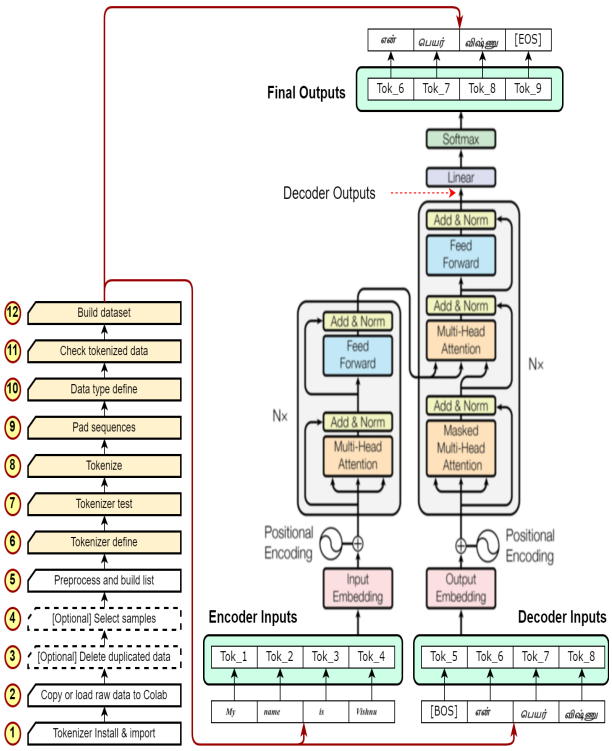

Figure 6: mT5 model

model's performance is assessed using the BLEU score metric provided by the sacrebleu library. This metric compares machine-generated translations to reference translations to quantify translation accuracy. Additionally, the script allows for specific sentences to be translated and outputs these translations to demonstrate the model's real-world application. Finally, training losses are plotted to visualize and analyze the model's learning trends over time, aiding in the identification of any potential issues and the overall evaluation of the training process.

This robust methodology integrates advanced NMT practices, including leveraging pre-trained models and fine-tuning on specific linguistic data, ensuring the system is well-suited for the complexities of English to Tamil translation while focusing on achieving high translation quality.

4.1 Multilingual Text-To-Text Transfer Transformer Working

The mT5 model revolutionizes multilingual natural language processing by presenting a unified framework capable of handling multiple languages within a single model. Building upon the success of the T5 architecture, mT5 leverages a massive-scale pretraining dataset covering numerous languages, enabling it to perform a wide range of natural language understanding and generation tasks. Through a shared vocabulary and encoder-decoder architecture, mT5 learns to translate between different languages, summarize text, answer questions, and perform other language-related tasks without requiring language-specific fine-tuning as shown in Figure 6. Its versatility and effectiveness stem from its ability to encode and decode text in multiple languages, making it a valuable asset for cross-lingual applications and improving accessibility to natural language processing technologies for a global audience.

5 Experiments

In the experimental phase of the project, extensive efforts were made to fine-tune the mT5 model for translating English into Tamil and Telugu, focusing on achieving high accuracy in neural machine translation for these Dravidian languages. Initially, the datasets were carefully curated to include a vast array of linguistic expressions, with 1,000,000 rows dedicated to training, complemented by smaller subsets of 1,000 and 2,390 rows for validation and testing respectively. This robust dataset was crucial in training the model comprehensively. The training process spanned three epochs, utilizing the AdamW optimizer coupled with a linear schedule for warm-up steps, which was meticulously configured to refine the learning rate progressively, enhancing the model's learning curve. The choice of Google Colab Pro as the computational platform, equipped with a powerful GPU (T4 GPU and L4 GPU), facilitated efficient model training and iteration. The Google Drive is used as Storage purpose, where we have store the datasets. It is more efficient than the local RAM on the System. It is more efficient to save and load the model on to Google colab. Throughout this phase, particular attention was given to the BLEU score calculation for each language pair, employing this metric as a quantitative measure of translation quality. The BLEU score served not only as an indicator of how well the model's translations aligned with human translations but also provided insights into areas where further model adjustments were needed. By leveraging these methods and technologies, the project aimed to push the boundaries of what is achievable with current NMT technology, especially in the context of languages with limited digital resources.

Table 3: Hyperparameters for Model Training

Hyperparameter	Value
Batch Size (bs)	16
Max Length	model.config.max-length
Model Repository	google/mt5-small
Number of Epochs	3
Learning Rate	5×10^{-4}
Print Frequency	500
Checkpoint Frequency	100
Number of Batches	ceil(len(t-df)/bs)
Total Steps	n-epochs $\times$ n-batches
No of Warmup Steps	0.01 $\times$ Total-Steps
Maximum Iterations	8

5.1 Hyperparameters

The following are hyper-parameters used in the model as shown in Table 3. we have the max length of the model is 20. The length of the train dataset (t-df) is taken as 1000000. So, the number of batches, total steps and Number of warmup steps will be 62500, 187500 and 1875, respectively. In mT5, we have mT5-small, mT5-base, mT5-large and mT5-3B. The main difference among these versions is the number of parameters and the computational resources required for training and inference. Larger versions tend to have more parameters and typically offer better performance, but they also require more computational resources. As the model parameters increases, the computational time also increase. So, we used the mT5-small model.

6 Software and Hardware Requirements

6.1 Hardware Requirements

Library/Version	Version
Python	3.10.12 [GCC 11.4.0]
Transformers	4.40.0
Sentencepiece	0.1.99
Datasets	2.19.0
Sacrebleu	2.4.2
GPU - CUDA	12.1
Google Colab	0.0.1a2
Matplotlib	3.7.1
Pandas	2.0.3
Seaborn	0.13.1
Numpy	1.25.2
IPython	7.34.0

Table 4: Software Requirements

Application	Specification
Google Drive	15 GB
Google Golab	1.0.0
Gmail Account	mail Id

Table 5: Application Requirements

6.2 Hardware Requirements

Hardware	Specification
Windows	10 and above
RAM	16 GB
ROM	1 TB SSD

Table 6: Hardware Requirements

7 Results

BLEU (Bilingual Evaluation Understudy) is a critical metric for evaluating the performance of Neural Machine Translation (NMT) systems, particularly when applied to Dravidian languages such as Tamil and Telugu. The BLEU score provides a quantitative measure of the quality of machine-generated translations by comparing them to one or more reference translations. This method calculates the number of matching n-grams between the machine output and the reference texts, adjusting for proper sentence length and ensuring that each word or phrase is not counted more than its occurrence in the reference, a technique known as brevity penalty.

We can observe the some of few translation from English to Tamil, Telugu using mT5 Model as shown in Figure 7. We can observe the translation have performed by the mT5 model and fine tuning.

Table 7: BLEU Scores for En-te and En-Ta

Language	En-Te	En-Ta
BLEU Score	31.35	36.55

Finally, we compared the sample translation and BLEU score, We know that BLEU score is good for Tamil when compared to Telugu dataset. Inspite of having good score for the tamil model, telugu model works far better due to language complexity. we observe few translations, english-telugu model is performing better when compared to english-tamil model. From translation sample as shown in Figure 7, it is clear that we got exact translation for telugu whereas we got translation in Tamil but in indirect way. And we also find few draw backs

Results

Language	Translations
En	My name is Vishnu Chunduru
Ta	விஷ்ணு சஞ்சுரு என்னுடைய பெயர்
Te	నా పేరు విష్ణు చుండూరు.

Figure 7: Neural Machine Translation results from English to Dravidian Languages such as Tamil (ta) and Telugu (te)

of tamil, due to complexity of language and poly phonetics sounds.

8 Conclusion

This project embarked on the ambitious task of enhancing Neural Machine Translation (NMT) capabilities for Dravidian languages, specifically focusing on Tamil and Telugu. Through the application of the mT5 model, we have demonstrated that it is feasible to achieve meaningful translations despite the inherent challenges presented by these low-resource languages. The experimental results underscored the potential of advanced NMT systems to make significant strides in language translation quality.

Our findings revealed that while the translation accuracy for Tamil was relatively higher, achieving a BLEU score of 0.36, Telugu translations lagged slightly behind with a BLEU score of 0.31. This disparity not only highlights the diverse linguistic complexities of Dravidian languages but also pinpoints the areas where further model tuning and training are necessary. The experiment's outcome emphasizes the need for continuous refinements and enhancements in NMT technology to cater effectively to each language's unique characteristics.

Furthermore, the project's success in utilizing a pre-trained model like mT5, optimized through fine-tuning and careful hyperparameter adjustments, sets a precedent for future research in this area. It encourages further exploration into more sophisticated models and training techniques that could potentially bridge the gap in translation quality among different Dravidian languages.

In conclusion, while the project has taken significant steps toward improving NMT for Dravidian languages, the journey towards perfecting this technology continues. Future work will involve deeper linguistic analysis, broader datasets, and more nuanced model adjustments to better capture the linguistic nuances of each language. With continued efforts, NMT can be expected to play a pivotal role in breaking down language barriers and enriching communication across diverse linguistic landscapes.

9 Futurescope

The future scope for enhancing NMT from English to Dravidian languages is promising. Future research could extend to include more Dravidian languages such as Kannada and Malayalam, improving accessibility for a broader demographic. Efforts to refine the NMT model could explore advanced machine learning techniques and domain-specific adaptations to boost accuracy and fluency. Another key area is the expansion of training datasets, especially for underrepresented languages, to enhance model training with high-quality, diverse language samples. Additionally, developing real-time translation applications could provide immediate benefits in educational and commercial settings. Incorporating user feedback mechanisms could further allow continuous improvement of the translation system based on real-world usage. Lastly, focusing on the translation of idiomatic expressions and cultural nuances could significantly enhance the contextual accuracy of translations, fostering better communication and understanding across different cultures.

10 Contribution

Vishnu Choundur:

- Loads English-Tamil translation datasets for training, validation, and testing.
- Preprocesses the data by combining English and Tamil sentences into translation pairs.
- Defines training parameters such as epochs, batch size, learning rate, etc.
- Iterates over the training data in batches, performs forward pass, calculates loss, and updates weights.
- Prints training updates including epoch, step, average loss, and learning rate.
- Periodically evaluates the model on the test set to monitor performance.
- Encodes the source (English) and target (Tamil) texts using the tokenizer, ensuring padding and truncation to fit the model's maximum input length.

Vishnu Teja Yerrapragada:

- Loads a pre-trained sequence-to-sequence model (mt5-small) and its tokenizer.
- Optionally moves the model to GPU if available.
- Evaluates the trained model using BLEU score on the test set.
- Calculates BLEU score for individual sentences.
- Saves the trained model and tokenizer to specified paths in Google Drive.
- Optionally loads the model and tokenizer from the saved paths.

Naresh Bolishetty:

- Installs necessary packages like transformers, sentencepiece, datasets, and sacrebleu.
- Mounts Google Drive to access data files.
- Visualizes the smoothed training loss over epochs.
- Demonstrates translation by providing an English sentence and generating its Tamil translation using the trained model.

References

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 90–98.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*. Springer, pages 382–398.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. pages 65–72.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. Stereotyping norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. pages 1004–1015.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33:1877–1901.

Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. 2021. Findings of the shared task on hope speech detection for equality, diversity, and inclusion. In Bharathi Raja Chakravarthi, John P. McCrae, Manel Zarrouk, Kalika Bali, and Paul Buitelaar, editors, *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics, Kyiv, pages 61–72. https://aclanthology.org/2021.ltedi-1.8.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201* .

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2023a. Instructzero: Efficient instruction optimization for black-box large language models. *arXiv preprint arXiv:2306.03082* .

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023b. Exploring the use of large language models for reference-free text quality evaluation: A preliminary empirical study. *arXiv preprint arXiv:2304.00723* .

Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. 2019. Mixture content selection for diverse sequence generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. pages 3121–3131.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234* .

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Zhihao Fan, Yeyun Gong, Zhongyu Wei, Siyuan Wang, Yameng Huang, Jian Jiao, Xuan-Jing Huang, Nan Duan, and Ruofei Zhang. 2020a. An enhanced knowledge injection model for commonsense generation. In *Proceedings of the 28th International Conference on Computational Linguistics*. pages 2014–2025.

Zhihao Fan, Yeyun Gong, Zhongyu Wei, Siyuan Wang, Yameng Huang, Jian Jiao, Xuan-Jing Huang, Nan Duan, and Ruofei Zhang. 2020b. An enhanced knowledge injection model for commonsense generation. In *Proceedings of the 28th International Conference on Computational Linguistics*. pages 2014–2025.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554* .

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the aaai conference on artificial intelligence*. volume 32 (1).

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., volume 30.

John Hewitt, Christopher D Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. pages 3414–3427.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* .

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

EunJeong Hwang, Veronika Thost, Vered Shwartz, and Tengfei Ma. 2023. Knowledge graph compression enhances diverse commonsense generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, pages 558–572. https://aclanthology.org/2023.emnlp-main.37.

Kushal Jain, Adwait Deshpande, Kumar Shridhar, Felix Laumann, and Ayushman Dash. 2020. Indic-transformers: An analysis of transformer language models for indian languages.

Abhinav Jha, Hemprasad Yashwant Patil, Sumit Kumar Jindal, and Sardar M N Islam. 2023. Multilingual indian language neural machine translation system using mt5 transformer. In *2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*. pages 1–5. https://doi.org/10.1109/PCEMS58491.2023.10136051.

Akshara Kandimalla, Pintu Lohar, Souvik Kumar Maji, and Andy Way. 2022. Improving english-to-indian language neural machine translation systems. *Information* 13(5). https://doi.org/10.3390/info13050245.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint arXiv:2302.14520* .

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35:22199–22213.

Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*. pages 12–24.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, pages 431–469. https://doi.org/10.18653/v1/2023.findings-acl.29.

Bei Li, Rui Wang, Junliang Guo, Kaitao Song, Xu Tan, Hany Hassan, Arul Menezes, Tong Xiao, Jiang Bian, and JingBo Zhu. 2023. Deliberate then generate: Enhanced prompting framework for text generation.

Haonan Li, Yeyun Gong, Jian Jiao, Ruofei Zhang, Timothy Baldwin, and Nan Duan. 2021. Kfcnet: Knowledge filtering and contrastive learning for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. pages 2918–2928.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 110–119.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In *Proceedings of the 27th International Conference on Computational Linguistics*. pages 1033–1043.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning. *ArXiv preprint* .

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. pages 74–81.

Xin Liu, Dayiheng Liu, Baosong Yang, Haibo Zhang, Junwei Ding, Wenqing Yao, Weihua Luo, Haiying Zhang, and Jinsong Su. 2022. Kgr4: Retrieval, retrospect, refine and rethink for commonsense generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. volume 36, pages 11029–11037.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S Yu. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. volume 35, pages 6418–6425.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 4288–4299.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for abstractive text summarization. *arXiv preprint arXiv:2303.15621* .

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651* .

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2023. Gpteval: A survey on assessments of chatgpt and gpt-4. *arXiv preprint arXiv:2308.12488* .

Antonio Mastropaolo, Simone Scalabrino, Nathan Cooper, David Nader Palacio, Denys Poshyvanyk, Rocco Oliveto, and Gabriele Bavota. 2021. Studying the usage of text-to-text transfer transformer to support code-related tasks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. pages 336–347. https://doi.org/10.1109/ICSE43902.2021.00041.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023a. Locally typical sampling. *Transactions of the Association for Computational Linguistics* 11:102–121.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023b. Locally typical sampling. *Transactions of the Association for Computational Linguistics* 11:102–121.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. pages 311–318.

B. Premjith, M. Anand Kumar, and K.P. Soman. 2019. Neural machine translation system for english to indian language translation using mtil parallel corpus. *Journal of Intelligent Systems* 28(3):387–398. https://doi.org/doi:10.1515/jisys-2019-2510.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476* .

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21(1):5485–5551.

Srinivasan Rajendran and Subalalitha Cn. 2019. Automated named entity recognition from tamil documents. pages 1–5. https://doi.org/10.1109/ICESIP46348.2019.8938383.

Nesma M. Rezk, Madhura Purnaprajna, Tomas Nordström, and Zain Ul-Abdin. 2020. Recurrent neural networks: An embedded computing perspective. *IEEE Access* 8:57967–57996. https://doi.org/10.1109/ACCESS.2020.2982416.

Gokila S, S. Rajeswari, and S. Deepa. 2023. Tamil- nlp: Roles and impact of machine learning and deep learning with natural language processing for tamil. pages 1–9. https://doi.org/10.1109/ICONSTEM56934.2023.10142680.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems* 29.

Shanya Sharma, Manan Dey, and Koustuv Sinha. 2021. Evaluating gender bias in natural language inference. *arXiv preprint arXiv:2105.05541* .

Tianxiao Shen, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. Mixture models for diverse machine translation: Tricks of the trade. In *International conference on machine learning*. PMLR, pages 5719–5728.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*. volume 31 (1).

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(56):1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27.

Guy Tevet and Jonathan Berant. 2021. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. pages 326–346.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need .

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 4566–4575.

Raj Vyas, Kirti Joshi, Hitesh Sutar, and Tatwadarshi P. Nagarhalli. 2020. Real time machine translation system for english to indian language. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. pages 838–842. https://doi.org/10.1109/ICACCS48705.2020.9074265.

Aditya Vyawahare, Rahul Tangsali, Aditya Mandke, Onkar Litake, and Dipali Kadam. 2022. PICT@DravidianLangTech-ACL2022: Neural machine translation on Dravidian languages. In Bharathi Raja Chakravarthi, Ruba Priyadharshini, Anand Kumar Madasamy, Parameswari Krishnamurthy, Elizabeth Sherly, and Sinnathamby Mahesan, editors, *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics, Dublin, Ireland, pages 177–183. https://doi.org/10.18653/v1/2022.dravidianlangtech-1.28.

Han Wang, Yang Liu, Chenguang Zhu, Linjun Shou, Ming Gong, Yichong Xu, and Michael Zeng. 2021. Retrieval enhanced model for commonsense generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. pages 3056–3062.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, pages 9840–9855.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846* .

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, pages 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wenhao Yu, Chenguang Zhu, Lianhui Qin, Zhihan Zhang, Tong Zhao, and Meng Jiang. 2022a. Diversifying content generation for commonsense reasoning with mixture of knowledge graph experts. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*. pages 1–11.

Wenhao Yu, Chenguang Zhu, Zhihan Zhang, Shuohang Wang, Zhuosheng Zhang, Yuwei Fang, and Meng Jiang. 2022b. Retrieval augmentation for commonsense reasoning: A unified approach. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. pages 4364–4377.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Extractive summarization via chatgpt

for faithful summary generation. *arXiv preprint arXiv:2304.04193* .

Tianhui Zhang, Danushka Bollegala, and Bei Peng. 2023b. Learning to predict concept ordering for common sense generation. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Nusa Dua, Bali, pages 10–19.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. pages 9134–9148.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. *Advances in Neural Information Processing Systems* 31.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023c. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923* .

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685* .

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren. 2021. Pre-training text-to-text transformers for concept-centric common sense. In *International Conference on Learning Representations*. https://openreview.net/forum?id=3k20LAiHYL2.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. pages 1097–1100.