RaMToE-FINE: Randomized Multilingual Training with Ordered Evaluation via Fine-Tuning and Incremental Data Expansion

Pradeep Sai Teja Sanka[1], Raja Pavan Vignesh Kajjayam[1], Sai Karthik Nallamothu[1], Yeshwanthvenktakumar Vidhuvaluru[1] , Tapas Kumar Mishra[1]

[1]*Department of Computer Science and Engineering, SRM University – AP, Andhra Pradesh, India*

Email: pradeepsaiteja_sanka@srmap.edu.in, rajapavanvignesh_k@srmap.edu.in, saikarthik_nallamothu@srmap.edu.in, tapaskumar.m@srmap.edu.in

Abstract: Multilingual textual summarization is a demanding task due to the semantic, syntactic and structural variance across many languages. In this research, we experiment with mT5 (Multilingual Text-to-Text Transfer Transformer) to do abstractive summarization across seven different major languages (Telugu, Urdu, Marathi, Hindi, Tamil, Bengali, English) spoken in India to analyze how a jumbled multi-lingual dataset volume influences model performance as the dataset increases. Our experiment steps involve incrementally increasing the number of data samples per language, starting with 10 instances per language and progressively upgrading up to 1000 samples and beyond. This controlled dataset expansion allows us to methodically check the impact of training data volume on the quality of output summaries. Model performance is evaluated using metrics such as ROUGE and BLEU scores. Generally, mT5 showcases good generalization capabilities, its effectiveness in many low resource languages is significantly influenced by the size of data available. To further improve summarization quality for, we use transfer learning techniques and fine-tuning methods. This research experiment provides promising understanding of adaptability and scalability of large multilingual transformer-based models, giving a path for optimizing summarization tasks in a wide range of linguistic works. Our key findings contribute to booming the domain of multilingual NLP, particularly in inventing efficient summarization methods for inadequate languages.

Keywords: Multilingual Summarization; mT5; Abstractive Summarization; Low-Resource Languages; Dataset Size; Transfer Learning; ROUGE; BLEU; NLP.

INTRODUCTION

The growth of digital media across the earth has expanded the use of automated models or systems that are capable of amalgamating the information from various linguistic origins. Multilingual textual summarization, specifically of the abstractive type, represents a critical challenge in the natural language processing (NLP) domain due to the tangled interaction of syntactic complication, semantic wealth and structural difference across many languages. In a

linguistically varied nation like India, where hundreds of languages coexist with each other of which remain under-resourced in computational terms—this challenge takes on a peak significance. This study uses the mT5(Multilingual Text-to-Text Transfer Transformer) , a state-of-the-art transformer model, to perform abstractive based summarization across seven major languages spoken in India(Telugu, Urdu, Marathi, Hindi, Tamil, Bengali, and English). These languages, representing a mixture of Dravidian and Indo-Aryan groups along with English which is a widely spoken language in the world, offers an extensive testbed for checking the potentiality of large-scale multilingual models on these languages. Our principal objective is to investigate how the size of a mixed and shuffled multilingual dataset influences the performance of a multilingual transformer(mT5 in this research) in generating contextually, concise, coherent and accurate summaries. To achieve this, we took a controlled experimental procedure, incrementally increasing the number of training samples per language, from an initial set of 10 samples per language to over 1000 and beyond along with fixed fine tuning settings. This systematic expansion allows us to methodically assess the impact of data size and shuffled learning on summarization quality, a critical factor given the differences in resource availability among the languages under study. Model performance is evaluated using metrics such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual Evaluation Understudy), which gives numerical insights into the crossover among model-generated and human based summaries .

mT5 is one of the widely used transformer models due to its ability to perform on diverse languages and its availability to public usage. Generally, this model exhibits good generalization capabilities across linguistically diverse datasets. However, its effectiveness in low-resource languages such as Telugu, Marathi or Urdu remains highly dependent on the size of the training dataset. To address this limitation and improve summarization quality, we employ techniques including transfer learning and fine-tuning, which shapes the pre-trained mT5 model to our particular multilingual work. These methods prove instrumental in pumping performance, particularly for languages with sparse digital notation.

This research paper not only highlights the adaptability and scalability of large multilingual transformer-based models but also pays a path for optimizing abstractive summarization across a wide range of linguistic contents and checks whether the model is able to capture the patterns despite being confused by different languages in different order. By throwing light on the relation between data volume and model performance with jigsaw learning, our work contributes to the evolving field of multilingual NLP, with a particular focus on developing efficient summarization techniques for under-resourced languages along with high ones. The insights gained from these experiments hold promise for applications ranging from real-world news headlines to educational content usage in our society. At last, our key findings aim to advance the domain of

NLP by providing an overview for handling linguistic diversity, ensuring that even languages with less computational resources can benefit from summarization.

The remaining part of the paper is distributed as follows. Section II discusses related works previously done in this area. Section III presents the detailed information of the suggested workflow and the implemented system is evaluated in Section IV. Finally, Section V concludes the paper by underlining the conclusion and future works.

LITERATURE REVIEW

Discuss the related works in this section Research on multilingual text summarization methods has rapidly grown from the past few years with the rise of neural nets, especially transformer models, even though risk occurs in using these models across languages with different data availability. Traditional extractive summarization methods, such as LexRank (Erkan \& Radev, 2004) and TextRank (Mihalcea \& Tarau, 2004), performed well for monolingual but poorly performed with multilingual languages due to linguistic variations and syntactic inconsistencies. Early neural network models, particularly sequence-to-sequence (seq2seq) architectures with attention mechanisms (Bahdanau et al., 2015; See et al., 2017), improved abstractive summarization by giving more fluent and concise summaries. However, their performance in multilingual settings was struggled by the lack of large parallel datasets for training.

The dissemination of pre-trained language models, like mBART (Liu et al., 2020), mT5 (Xue et al., 2021), and multilingual BERT (Devlin et al., 2019), has entirely revolutionized multilingual text summarization tasks in NLP. These models employ self-supervised learning on large volumes of multilingual corpora, making the transfer of knowledge among languages, particularly improving low-resource languages (Conneau and Lample, 2019). Cross-lingual transfer learning facilitates the transfer of summarization models trained on high-resource languages such as English, Spanish and Chinese to low-resource languages with little training data. Further, methods such as back-translation and synthetic data creation have been established to address data unavailability in low-resource languages.

New multilingual standards such as XL-Sum (Hasan et al., 2021) and WikiLingua (Ladhak et al., 2020) offer extensive multilingual summaries in many languages with improved benchmarking. Summarization performance continues to be unequal across languages owing to different linguistic complexities. For example, very spoken languages with abundant morphological complexity, e.g., Hindi, Urdu and Arabic, exhibit more issues than syntactically less complex languages such as English (Goyal et al., 2022). In addition, code-switching in which several languages are employed within a single text tends to complicate summarization processes.

Although there are developments, multilingual summarization continues to have a number of controversies, such as data sparsity, cross-lingual generalization, and dialectal differences. Numerous pre-trained multilingual models inherit biases from their training data, subjecting to fairness issues(Shah et al., 2020). Methods like adversarial debiasing and bias-controlled training have been suggested and employed to alleviate such impacts. Further, evaluation metrics for performance such as ROUGE and BLEU have been found to lack the semantic quality of the summary, especially for low-resource languages with reversed word orders and alternative grammar principles. Recent methods such as BERTScore (Zhang et al., 2020) and COMET (Rei et al., 2020) have come to counter such limitations through contextual embeddings. Yet another major annoyance in multilingual text summarization is computation time. Transformer models are extremely resource intensive and thus prove to be challenging to implement in real-world situations, particularly in low-resource environments. Methods like quantization, pruning and knowledge distillation (Jiao et al., 2020) are being investigated to enhance model's throughput without heavily compromising the performance.

METHODOLOGY

System Setup

Intel system with CPU training with 32 GB of Ram is used. Python is used for system implementation due to its popularity and libraries (numpy, pandas and matplotlib) for Deep Learning. NLP Libraries such as NLTK (Natural Language Toolkit), Transformers are used for tokenization and segmentation, ROUGE, BLEU is utilized for score calculation, and Git is used for code-base management and team collaboration. Jupyter Notebook is used to write and run code.

Dataset

Dataset which is used in this research is a mixed and shuffled multilingual summarization data which is prepared from the XL-Sum(Extreme Long Summarization) dataset, which contains various news articles and their summaries in various languages along with urls, title and date. The XL-Sum dataset, created by CSEBUET NLP (Bangladesh University of Engineering and Technology, Computer Science and Engineering, Natural Language Processing) team, a massive-scale benchmark multilingual summarization dataset of news articles and their summaries respectively}. It is constructed from BBC News articles in 45 languages, one of the largest datasets for evaluation and benchmarking. The dataset was imported from hugging face and iterated through a predefined list of seven languages, loading subsets in a loop from the training split of each language using the datasets library. The extracted data are converted and concatenated into a single pandas data-frame.

Data Preparation

1) Padding and Attention Mask

The tokenized sequence is padded to a fixed length L:

$$T' = (t1, t2, ..., tn, P, P, ..., P) \quad (1)$$

where P represents padding tokens, ensuring a uniform length L, which ensures that the model ignores padding during training.

The attention mask is defined as:
- Ai = 1, if ti is not a padding token (P)
- Ai = 0, if ti is a padding token (P)

2) Label Adjustment for Loss Computation

The summary labels Y are also tokenized, and padding token IDs are replaced with -100 so they do not contribute to the loss function:

- Yi' = Token ID of Yi, if Yi is not a padding token (P)
- Yi' = -100, if Yi is a padding token (P) (2)

3) Training Data Split

The dataset is divided into training (80%) and testing (20%) data.

4) Jigsaw Learning

Jigsaw Learning is used, where the training data is shuffled to help the model learn diverse patterns without overfitting. The test data remains in its original sequence (Telugu, Urdu, Marathi, Hindi, Tamil, Bengali, and English). This helps the model generalize better while ensuring structured evaluation.

5) Batch Creation

Each batch contains 10 samples, and the training data is shuffled for better generalization.

A custom PyTorch dataset (Summary Dataset) is created for text summarization, where news articles and summaries are tokenized with a max token length (200 for text, 12 for summaries). Padding, truncation, and attention masks are applied, and padding token IDs in labels are replaced with -100 to be ignored during loss computation. The dataset is divided into training (80%) and testing (20%) subsets, and PyTorch DataLoader objects are initialized with a batch size of 10.

mT5 Model

T5 (Text-To-Text Transfer Transformer) was developed by Google in 2019 and treats all NLP tasks as text-to-text, enabling efficient transfer learning and fine-tuning. mT5 (Multilingual T5) is an extension of T5, supporting over 100 languages. It is trained using a masked span prediction objective over the mC4 dataset.

mT5 Transformer Architecture

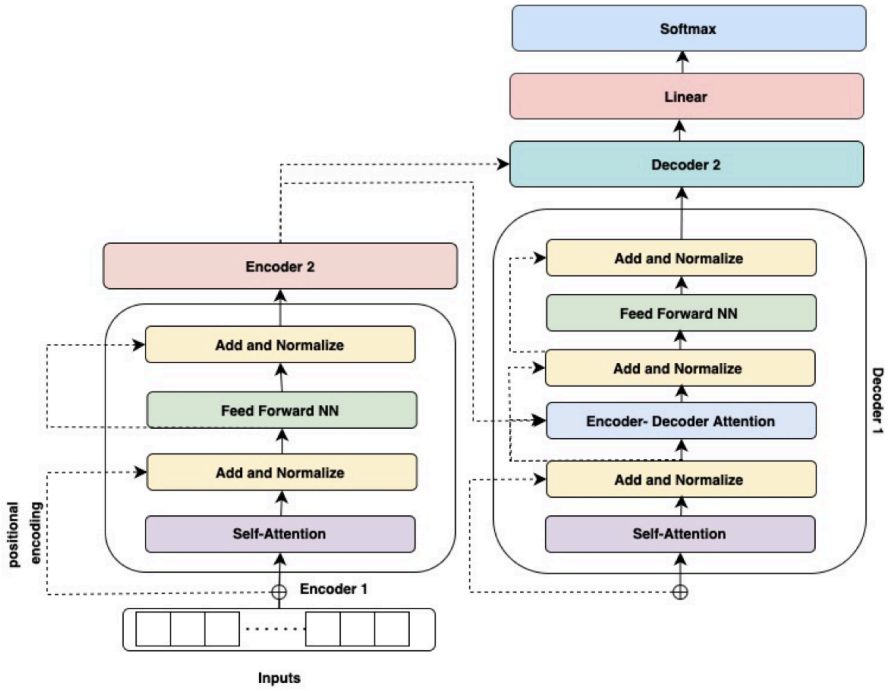

Fig. 1: mT5 Architecture

1) **Encoder**: Processes input text into a continuous representation that captures word relationships.

2) **Decoder**: Generates text based on encoder input. The decoder functions autoregressive, generating one token at a time.

3) **Self-Attention Mechanism:** This mechanism allows the Transformer to weigh the relevance of words in the text, regardless of their positional distance. The attention formula:

$$\text{Attention}(Q, K, V) = \text{softmax}((Q * K^T) / \text{sqrt}(dk)) * V \quad (3)$$

where Q, K, and V are the queries, keys, and values derived from the input embeddings, and dk is the dimension of the keys.

4) **Feed-Forward Neural Networks:** Each encoder and decoder layer includes a fully connected neural network with ReLU activation:

$$FFN(x) = \max(0, x * W1 + b1) * W2 + b2 \quad (3)$$

where W1, W2 are weight matrices and b1, b2 are bias terms.

5) **Layer Normalization and Residual Connections :** These help stabilize the learning process in each encoder and decoder block.

6) **Softmax Function :** The softmax function is used in the attention mechanism to compute attention weights. It converts logits into a probability distribution:

$$Softmax(zi) = \exp(zi) / \text{sum}(\exp(zj)) \text{ for } j = 1 \text{ to } n \quad (4)$$

This ensures that all output values sum to 1, making it useful for probability-based decisions in the Transformer model.

Pre-processing in mT5

1) **Text Tokenization:** Raw input text is tokenized using the mT5 tokenizer, which splits text into subwords. Tokenized text is converted into numerical IDs using mT5's vocabulary.

2) **Model Inputs:** The tokenized input is fed into the model, where attention masks prevent the model from focusing on padding tokens.

3) **Running the Model:** The processed input is passed to the mT5 model, producing raw logits or token embeddings. Text generation uses decoding methods like beam search, greedy decoding, or sampling algorithms.

4) **Fine-Tuning :** Fine-tuning adjusts the weights of the pre-trained model using task-specific data. The model is trained for 5 epochs using the AdamW optimizer and a linear learning rate scheduler. Checkpoints are stored after each epoch for resumption or testing.

Hyperparameters Used in Fine-Tuning:

Hyperparameter	Value
Number of Epochs	5
Batch Size	10
Optimizer	AdamW
Learning Rate Scheduler	Linear

Table 1. Hyperparameters Used

5) **Post-Processing:** Convert numerical IDs back to text using the tokenizer's decode function. Remove unnecessary tokens or spaces introduced during tokenization.

Evaluation Metrics

Text-based tasks such as summarization, question answering, or translation require automatic evaluation metrics to compare generated text with a given reference text. The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual Evaluation Understudy) scores are the most commonly used evaluation metrics..

1. **ROUGE-1:** ROUGE-1 measures the unigram (single-word) overlap between the generated text and the reference text:

ROUGE-1 = (Number of unigrams in both generated and reference text) / (Total unigrams in reference text) (5)

2. **ROUGE-2:** ROUGE-2 extends ROUGE-1 by measuring the bigram (two consecutive words) overlap:

ROUGE-2 = (Number of bigrams in both generated and reference text) / (Total bigrams in reference text) (6)

3. **ROUGE-L:** ROUGE-L measures the longest common subsequence (LCS) between the generated and reference text:

ROUGE-L = (LCS length between generated and reference text) / (Total words in reference text) (7)

4. **ROUGE-L Sum:** ROUGE-LSum is a variant of ROUGE-L designed for multi-sentence summarization. It considers sentence-wise matching across summaries:

ROUGE-L Sum = (Sum of LCS between generated and reference text) / (Sum of total words in reference text) (8)

5. **BLEU Score:** The BLEU score measures the similarity between generated and reference text using n-gram precision and a brevity penalty. It is computed as:

$$BLEU = BP \times \exp\left(\sum (w_n \times \log P_n)\right) \quad (9)$$

where:
- P_n is the precision for n-grams.
- w_n are weights assigned to each n-gram.
- BP (brevity penalty) prevents bias toward short translations.

- **Precision Calculation**: Individual precision values for 1-gram, 2-gram, 3-gram, and 4-gram sequences represent the proportion of matched n-grams. The precision for an n-gram in the BLEU score is computed as:

$$P_n = (\text{Number of matched n-grams}) / (\text{Total generated n-gram}) \quad (10)$$

where:
- "Number of matched n-grams" is the count of n-grams in the generated text that also appear in the reference text.
- "Total generated n-grams" is the total number of n-grams in the generated text.

- **Brevity Penalty (BP):** The brevity penalty penalizes shorter translations to prevent artificially high scores for very short outputs. It is defined as:

$$\begin{aligned} &- BP = 1, \text{ if } c > r \\ &- BP = e^{(1 - r/c)}, \text{ if } c \leq r \end{aligned} \quad (11)$$

where:
- c is the length of the generated text.
- r is the length of the reference text.

EXPERIMENTAL STUDY AND RESULT ANALYSIS

ROUGE scores, especially ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum, show a linear increase with growing numbers of samples. ROUGE-1, the score for unigram overlap, reaches 500 samples (0.0605) and then decreases slowly for larger datasets. This shows that the expansion of the dataset improves the model's capability to retain crucial words of the reference text early but that additional expansion is of decreasing returns. ROUGE-2, which computes bigram overlap, also shows a similar trend, with the highest value at 1000 samples (0.0221) before slightly declining at 5000 samples (0.0181). The decline of ROUGE-2 at larger sample sizes can be interpreted as an indication of more variation in phrasing and fewer exact bigram matches.

Samples	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum
10	0.0252	0.0000	0.0252	0.0252
50	0.0420	0.0108	0.0387	0.0392
100	0.0513	0.0190	0.0485	0.0480
500	0.0605	0.0180	0.0601	0.0597
1000	0.0551	0.0221	0.0540	0.0536
5000	0.0555	0.0181	0.0543	0.0544

Table 2. Rouge Scores of Different samples used

ROUGE-L and ROUGE-Lsum exhibit the stabilization trend after 500 samples. This implies that while overall structure and meaning are still being maintained in the model, larger dataset increases no longer affect long-sequence matching much. Also, consistent with the first result, stability of alignment for ROUGE-L and ROUGE-Lsum means consistent content preservation in generated summaries with respect to complete reference documents. These indicate the model achieving peak performance within the range 500-1000 samples and more training data from there would no longer bring appreciable change to text similarity.

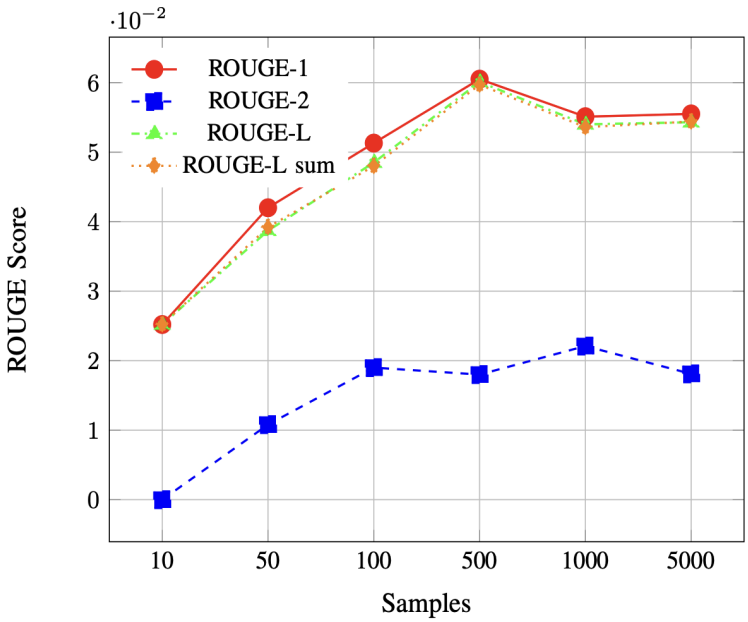

Fig. 2: ROUGE Scores of Different samples

BLEU scores rise up to 1000 samples with the score being 0.1006. This is evidence that the model writes more accurate and fluent text as the dataset rises. But when it gets to 5000 samples, the BLEU score drops to 0.0935. This decrease is a signal that as the model produces more varied text, it also moves away from literal n-gram matches with the reference, impacting BLEU's precision-oriented scoring. Since BLEU quantifies exact n-gram overlaps, this decrease means more linguistic variation is introduced in the output text as sample sizes increase.

Samples	BLEU Score
10	0.0000
50	0.0603
100	0.0659
500	0.0828
1000	0.1006
5000	0.0935

Table 3. BLEU Scores of Different samples used

A comparison of the BLEU and ROUGE scores directly reveals that both are improving with more samples, peaking at approximately 1000 samples. Although ROUGE remains flat at 5000 samples, BLEU dips slightly. This suggests that while the model is continuing to generate text that is similar to reference summaries in terms of overall content (as scored by ROUGE), it does

so with greater paraphrasing and structural variation, resulting in lower BLEU scores. This difference highlights one of the key distinctions between the two metrics—ROUGE is recall-oriented and focused on content preservation, while BLEU is precision-oriented and penalizes non-exact wording.

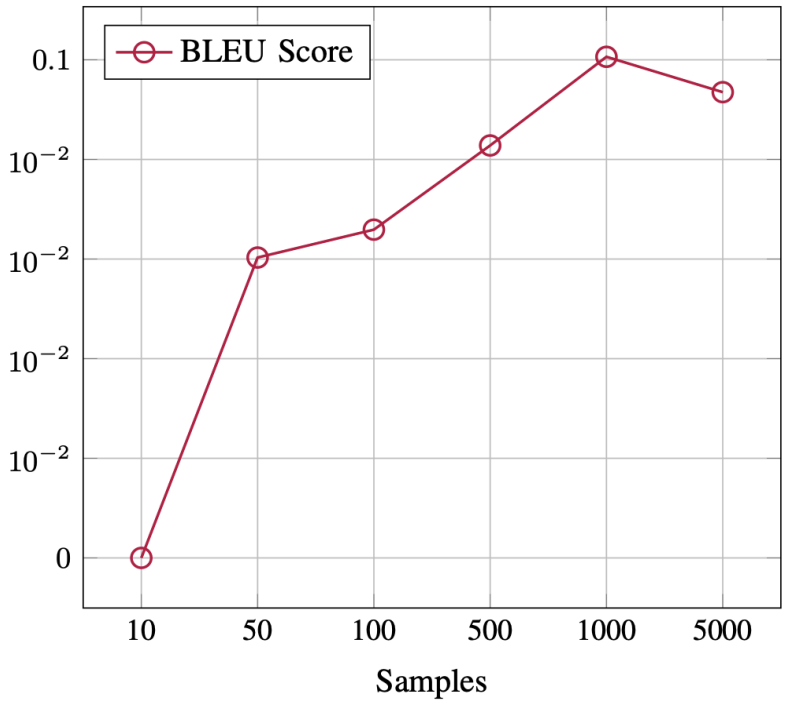

Fig. 3: BLEU Scores of Different samples

These findings indicate that for the sake of rigid text matching tasks such as machine translation, BLEU is a superior evaluation metric. However, for summarization and other tasks where meaning preservation is more critical than exact wording, ROUGE provides a more reliable assessment. The slight drop in BLEU at 5000 samples suggests the need for fine-tuning decoding strategies, such as beam search or reinforcement learning, to balance precision and linguistic diversity.

CONCLUSION AND FUTURE WORK

Analysis of BLEU and ROUGE score indicates that the model is best at 500–1000 samples, and improvement diminishes. BLEU scores indicate rising deviation from reference text, with strategies such as data augmentation, paraphrase-sensitive loss functions, or controlled generation methods needed to improve n-gram accuracy. ROUGE stabilizes for maintaining meaning, with potential gains through reinforcement learning to render content selection more precise. Future studies can explore models like BART, GPT models, and PEGASUS, which are abstractive summarization models and can hopefully achieve higher ROUGE scores. Retrieval-augmented generation (RAG) and reinforcement learning-based fine-tuning also enhance coherence and factuality. Investigating multilingual and domain-specific variants of these models can further enhance text generation. Hybrid evaluation metrics that combine semantic similarity measures can give a more holistic evaluation beyond the typical BLEU and ROUGE scores.Subsequent studies might also explore human-in-the-loop feedback mechanisms to further enhance evaluation metrics and confirm that the generated text is consistent with human expectations. Further extension to low-resource languages and real-time applications of text generation might also increase the reach of this research.

REFERENCES

1. K. Nair and R. Gupta, “Application of AI technology in modern digital marketing environment,” *World Journal of Entrepreneurship, Management and Sustainable Development*, vol. 17, no. 3, pp. 318–328, 2021.
2. L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, “mT5: A massively multilingual pre-trained text-to-text transformer,” *arXiv preprint arXiv:2010.11934*, 2020.
3. P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury, “The state and fate of linguistic diversity and inclusion in the NLP world,” *arXiv preprint arXiv:2004.09095*, 2020.
4. Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, “Efficient transformers: A survey,” *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–28, 2022.
5. G. Erkan and D. R. Radev, “LexRank: Graph-based lexical centrality as salience in text summarization,” *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.
6. R. Mihalcea and P. Tarau, “TextRank: Bringing order into text,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411, 2004.
7. Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, “Multilingual denoising pre-training for neural machine translation,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.
8. 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)*, pp. 4171–4186, 2019.
9. A. Conneau and G. Lample, “Cross-lingual language model pretraining,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
10. T. Hasan, A. Bhattacharjee, M. S. Islam, K. Samin, Y.-F. Li, Y.-B. Kang, M. S. Rahman, and R. Shahriyar, “XL-Sum: Large-scale multilingual abstractive summarization for 44 languages,” *arXiv preprint arXiv:2106.13822*, 2021.

11. F. Ladhak, E. Durmus, C. Cardie, and K. McKeown, "WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization," *arXiv preprint arXiv:2010.03093*, 2020.
12. Y. Liu, J. Gu, N. Goyal, X. Li, S.Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.
13. T. Hasan, A. Bhattacharjee, M. S. Islam, K. Samin, Y.-F. Li, Y.-B. Kang, M. S. Rahman, and R. Shahriyar, "XL-Sum: Large-scale multilingual abstractive summarization for 44 languages," *arXiv preprint arXiv:2106.13822*, 2021.
14. T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," *arXiv preprint arXiv:1904.09675*, 2019.
15. R. Rei, C. Stewart, A. C. Farinha, and A. Lavie, "COMET: A neural framework for MT evaluation," *arXiv preprint arXiv:2009.09025*, 2020.
16. X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.
17. K. N. Matthews, *The Impact of the Jigsaw Technique on Improving Knowledge Retention*. PhD thesis, University of South Alabama, 2021.
18. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:2301.12345*, 2023.
19. L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mT5: A massively multilingual pre-trained text-to-text transformer," *arXiv preprint arXiv:2010.11934*, 2020.
20. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
21. N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, et al., "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

22. C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*, pp. 74–81, 2004.
23. 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 of the Association for Computational Linguistics*, pp. 311–318, 2002.