TextGen-BERT Classifier

Samira Malek
sxm6547@psu.edu

Farzad Azarmi
fka5196@psu.edu

1 Abstract

The primary goal of text generation is to transform input data into coherent natural language output. Essentially, this process involves automatically learning a mapping from input to output using data, thereby establishing a comprehensive end-to-end solution.These models performed greatly in recent years. The evaluation of such text generation models is still an open research area, however. This study evaluates the performance of eight distinct text generation models—CodeGen, GPT-2, GPT-Neo, OPT, T5, BART, XLNet, and BlooM—using the BERT-base model for classification and analysis of generated outputs. The testing section revealed an accuracy of 85.5% indicating the effectiveness of the approach.

2 Introduction

Natural Language Processing (NLP) is a branch of artificial intelligence that studies the interaction between computers and humans using natural language. It allows machines to perceive, interpret, and produce human language in a meaningful and practical manner. In this regard, Text generation, often known as natural language generation, is a fundamental subfield in NLP (Li et al., 2024). Accordingly, the process of automatically producing human, such as text from structured data including databases or other information sources, is known as natural language generation. It transforms unstructured data into comprehensible and insightful stories, summaries, or reports using models and algorithms.

The main objective of text creation is to Convert input data into natural language output. In other words, text generation aims to automatically learn a mapping from input to output using data, creating a comprehensive end-to-end solution (Yu et al., 2022). This is using predetermined input—such as context, keywords, or numerical data—to create logical writing that appropriately conveys the input while preserving its relevance and intelligibility. Examples of text generation tasks include, but are not limited to, translation, question answering, as well as sequence-to-sequence procedures (Dušek et al., 2020; Yu et al., 2022).

Text generation is important for several reasons. It improves efficiency by automating large volumes of text creation, enhancing creativity through the rapid generation of unique content. Additionally, it increases accessibility for individuals with disabilities or language barriers, fosters better customer engagement through personalized interactions, and supports language learning by providing feedback and structured writing practice (McKeown, 1992).

Text generation in tweets is important because it streamlines communication, allowing users to share information quickly and effectively. It offers several benefits, including time efficiency by automating tweet creation, ensuring consistent tone and style for brand identity, and personalizing content to enhance audience engagement. Additionally, it helps generate creative ideas, overcoming writer's block, and leverages data-driven insights to produce content based on trends and user preferences.

Implementing a classification method to identify the source of generated text from various LLMs offers several significant benefits. Firstly, model attribution allows for accurate identification of which LLM produced specific text, ensuring accountability in sensitive areas like journalism, law, and education. This transparency enables users to assess the reliability of the content. Secondly, by associating text outputs with their originating models, users can monitor performance patterns, facilitating better model selection based on task-specific requirements and leading to improved quality control. Additionally, recognizing the biases that different LLMs may introduce—stemming from their unique training data—enables the detection and analysis of these biases, enhancing the understanding of model behavior. Finally, as LLMs become increasingly integrated into public-facing applications, the ability to trace the origin of generated text fosters trust and transparency in AI-generated content, which is especially crucial in regulated industries and consumer interactions. Overall, these advantages promote a more responsible and informed use of AI technologies. In this research, we aim to find the best text generation algorithm based on the accuracy of a classifier on several tweets captured from the web.

3 Related Work

Text generation evaluation and quality control, Bias detection, and trust in AI have been the subject of several research, with an emphasis on metrics, human evaluation, and controllability methodologies. The ethical aspects of AI use, encompassing prejudice and disinformation, are also crucial and should spark conversations. The overall goal of this research is to enhance AI-generated text's coherence, relevance, and ethical implications for a range of applications.

In this regard, Yuan et al. (Yuan et al., 2021) evaluated various variants of BARTSCORE from seven perspectives across 16 datasets. BARTSCORE demonstrated superior performance, achieving the best results in 16 out of 22 test settings compared to existing top-scoring metrics. Their findings also highlighted the effectiveness of the prompting strategy used with BARTSCORE. Additionally, BARTSCORE showed greater robustness when assessing high-quality texts generated by leading systems.

In another research (Sellam et al., 2020), The researchers focused on enhancing quality control in text generation by training BLEURT, a neural metric for evaluating generated text. The study begins by confirming BLEURT's superior performance on the WMT Metrics Shared Task from 2017 to 2019, demonstrating its effectiveness across various English-to-other language pairs. To assess its robustness, the researchers conduct stress tests using a synthetic benchmark that highlights how BLEURT handles quality variations. They further illustrate its adaptability to different domains by applying it to tasks from the WebNLG 2017 dataset, showcasing its versatility. Overall, the findings emphasize BLEURT's potential as a reliable tool for ensuring high-quality text generation across diverse contexts.

Zhong et al. (Zhong et al., 2022) also addressed the limitations of current evaluation methods in Natural Language Generation by proposing a unified multi-dimensional evaluator called UNIEVAL. Unlike traditional automatic evaluation, which primarily relies on similarity-based metrics, UNIEVAL reframes NLG evaluation as a Boolean Question Answering (QA) task. By guiding the model with various evaluative questions, UNIEVAL can assess multiple dimensions of text quality, such as coherence and fluency, using a single framework. UNIEVAL correlates significantly better with human judgments compared to existing metrics, achieving a 23% higher correlation in text summarization and over 43% in dialogue response generation.

In another research (Zhu et al., 2018), Texygen was introduced as a metric to evaluate the performance of the text quality. Texygen is a benchmarking platform designed to advance research in open-domain text generation models. It includes a comprehensive implementation of various text generation models and offers a suite of metrics to evaluate the diversity, quality, and consistency of generated texts. By standardizing research practices in this area, Texygen aims to enhance the reproducibility and reliability of future studies in text generation, making it a valuable resource for researchers.

In another research (Zhao et al., 2019), the authors approach the evaluation of text generation systems by measuring the semantic distance between the generated outputs and reference texts, relying on robust continuous representations to capture semantic and syntactic variations effectively. They explore the use of existing contextualized representations in conjunction with the Earth Mover's Distance—a metric that quantifies the dissimilarity between distributions—to compare system predictions against reference texts. This investigation leads to the development of a new automated evaluation metric that demonstrates a high correlation with human assessments of text quality. Their metric shows competitive or superior performance compared to strong baseline methods across four key text generation tasks: summarization, machine translation, image captioning, and data-to-text generation, indicating its potential as a valuable tool for future evaluations in the field.

Krishna et al. (Krishna et al., 2022) also used RankGen to improve text generation quality. RankGen is a 1.2 billion parameter encoder model designed to improve text quality generated by language models, addressing issues like repetition, incoherence, and irrelevance. It operates by scoring model-generated sequences based on a given prefix, allowing it to be integrated as a scoring function in beam search for any pretrained language model. Trained through large-scale contrastive learning, RankGen positions ground-truth sequences closer to the prefix while distancing them from irrelevant or low-quality sequences. Experiments across various language models demonstrate that RankGen outperforms decoding methods such as nucleus sampling and top-k sampling, achieving higher scores on both automatic metrics and human evaluations. The outputs generated by RankGen show improved relevance, continuity, and coherence, significantly enhancing overall text quality. The researchers have made their model checkpoints, code, and preference data publicly available to support future research.

Overall, text generation evaluation is crucial for ensuring that generated outputs meet desired standards of coherence, relevance, and fluency. Effective evaluation helps identify and mitigate issues like repetition and incoherence, which can undermine user trust in AI-generated content. Accordingly several metrics has been used to identify quality control, bias detection, as well as trust and transparency in text generation procedures.

4 Project Plan

In this research, we explored the capabilities of various text generation models to complete truncated tweets collected from the web. Our study utilized several state-of-the-art models, including CodeGen, GPT-2, GPT-Neo, OPT, T5, BART, XLNet, and BlooM, each selected for their distinct strengths in natural language processing. Then Bert-base classification method was used to evaluate the output of the models in terms of having distinctive approach.

To achieve our goal, we employed pretrained models specifically designed to understand and generate human-like text. These models were fine-tuned to enhance their performance on our dataset, which comprised a diverse array of incomplete tweets. The results demonstrated a significant increase in processing efficiency and output quality when using these pretrained models compared to non-trained models. This improvement was evident in both the fluency of the generated text and the ability to capture the nuances of social media language.

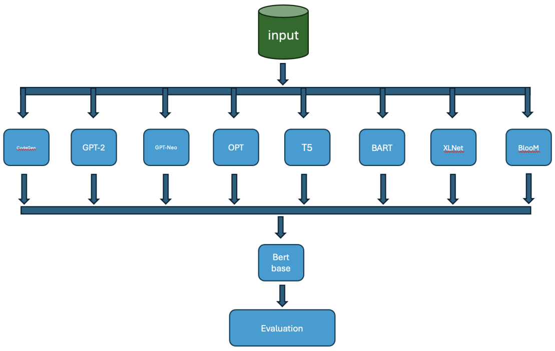

Figure 1: The overall procedure of the research. Multiple text generation models to were used produce outputs, followed by the application of the BERT-base model for classification and evaluation of these texts.

In the following sections, we provide a detailed description of the dataset, the characteristics of each model used, and the methodologies implemented in our study.

5 Data

The COVID19 Fake News Detection in English dataset (Patwa et al., 2021) is specifically designed to combat fake news related to COVID-19, comprising social media posts and articles labeled as either real or fake. The data collection focuses on popular social networking platforms that facilitate peer communication and information sharing, encompassing news, events, and social phenomena. To ensure comprehensive coverage, we gather fake claims from various reputable fact-checking websites, such as Politifact and NewsChecker, as well as tools like Google Fact Check Explorer and the IFCN chatbot. Real news is sourced from verified Twitter accounts, ensuring authenticity. Additionally, we conduct exploratory data analysis to gain insights into the dataset and implement four machine learning models to further analyze and differentiate between real and fake information on the COVID-19 topic. This dataset aims to enhance the understanding and detection of misinformation in the context of a global health crisis.

In this research, we utilized a dataset comprising 2,000 tweets for each model, and then employing a classifier to effectively identify the text generation method used. In each model we used 10 tokens for the input and 50 tokens for the output. The dataset is separated into 10,080 samples for training, 1,014 samples for validation, and 4,906 samples for testing. In this context, the training dataset is used to train the model, allowing it to learn patterns and relationships in the data; the validation dataset is employed to tune hyperparameters and evaluate the model's performance during training, helping to prevent overfitting; and the test dataset is utilized to assess the final performance of the model on unseen data, providing an unbiased evaluation of its effectiveness This approach allowed us to systematically analyze the characteristics and performance of different text generation techniques, facilitating a comprehensive understanding of their outputs. Table3 shows the size of the train, test, and validation datasets respectively.

	Size
Train dataset	10080
Validation dataset	1014
Test dataset	4906

Table 1: Table showing the sizes of the train, validation, and test datasets.

6 Models

Eight text generation models were utilized to produce outputs, which were then classified and evaluated using the BERT-base model. The text generation models are explained in detail in the next subsections.

CodeGen

CodeGen is a transformer-based model primarily designed for code generation tasks. Leveraging a large corpus of programming code, it excels at generating syntactically correct and contextually relevant code snippets. Its architecture allows for the seamless integration of natural language prompts, making it particularly useful for applications that require bridging the gap between natural language and programming. In this algorithm a method is proposed to that breaks the process down into multiple subprograms. By doing so, breaking a complex specification into multiple steps can enhance model understanding and improve program synthesis. This multi-turn approach allows the model to focus on the specifics of individual subprograms, reducing the complexity of interdependencies and making it easier for users to express their intent. Our experiments confirm that this strategy leads to higher quality synthesized programs. Additionally, we observe a weak pattern of interleaved natural and programming language within code, often seen in programmer comments. Although this interleaving provides a noisy supervision signal for generating programs from natural language descriptions, scaling the model and dataset can help leverage this pattern. Ultimately, this enables users to articulate their intent in multiple, manageable turns, allowing for a more structured and clear decomposition of programming tasks (Nijkamp et al., 2022).

GPT-2

GPT-2, developed by OpenAI, is a generative language model that uses unsupervised learning to predict the next word in a sentence based on context. Known for its impressive ability to generate coherent and contextually relevant text, GPT-2 has been widely adopted for various applications, including conversational agents and creative writing, due to its versatility and adaptability.

Indeed, GPT-2 exemplifies the advancements in multitask learning and transfer learning in natural language processing. By leveraging a large-scale unsupervised pretraining approach, GPT-2 is trained on vast amounts of text data, enabling it to learn intricate language patterns without the need for task-specific architectures. This model can effectively perform a variety of downstream tasks in a zero-shot setting, meaning it can generate relevant responses or complete tasks without prior task-specific training or architectural adjustments. Its ability to generalize across different tasks showcases the potential of language models to execute complex language functions, such as commonsense reasoning and sentiment analysis, achieving competitive results across various benchmarks. This flexibility and adaptability highlight GPT-2's role in advancing NLP capabilities and its significance in the evolution of machine learning approaches (Radford et al., 2019).

GPT-Neo

GPT-Neo is Developed by EleutherAI as a free substitute for proprietary models such as OpenAI's GPT-3. Indeed, this model is an open-source version of the GPT (Generative Pre-trained Transformer) architecture. Tailored for a range of natural language processing tasks, including text production, summarization, and question answering, GPT-Neo was created to produce text that resembles that of a person. Researchers and developers may experiment with large-scale language models without the high resource demands usually associated with commercial systems since it is accessible in multiple sizes, including models with 1.3 billion and 2.7 billion parameters. EleutherAI facilitates openness and promotes cooperative research and innovation in NLP and artificial intelligence by making GPT-Neo available to the community.

OPT

The Open Pretrained Transformer (OPT) model, created by Meta AI, is designed to facilitate research in large language models. Similar to GPT, OPT aims to produce human-like text through unsupervised learning. Its architecture and training methodologies allow it to be fine-tuned for specific tasks, enhancing its utility across different natural language processing applications.

OPT is a suite of decoder-only pre-trained transformer models developed to facilitate responsible and reproducible research in natural language processing. OPT aims to match the performance of models like GPT-3 while employing the latest practices in data collection and efficient training. The initiative seeks to broaden the understanding of large language models (LLMs) by providing researchers with access to these models, enabling the community to collaboratively address issues such as risk, harm, bias, and toxicity. The models achieve high computational efficiency while significantly reducing its carbon footprint compared to GPT-3. However, the report acknowledges the substantial energy costs associated with training such large models, emphasizing the need for ongoing dialogue about the environmental impact of LLMs (Zhang et al., 2022).

T5

The Text-to-Text Transfer Transformer (T5) is a model developed by Google that reframes all NLP tasks as a text-to-text format. This innovative approach allows it to handle a variety of tasks—ranging from translation to summarization—using a unified architecture. T5's versatility and strong performance across multiple benchmarks make it a valuable asset for text generation and understanding.

T5 is a model architecture that extends the capabilities of the original Transformer framework by reframing all natural language processing tasks as text-to-text tasks. This means that both the input and output of the model are treated as text strings, allowing for a unified approach to various tasks such as translation, summarization, and question answering. Building on the self-attention mechanism of Transformers, T5 employs a stack of encoder and decoder layers, where the encoder processes the input text and the decoder generates the output. T5 incorporates innovations such as relative position embeddings, which enhance its ability to understand context, and it has been designed to scale effectively across different model sizes and complexities. The training of T5 utilizes a combination of model and data parallelism on advanced hardware like Cloud TPU Pods, allowing it to handle large datasets and achieve state-of-the-art performance across multiple NLP benchmarks. By adopting a flexible and efficient architecture, T5 significantly advances the field of transfer learning in NLP (Raffel et al., 2020).

BART

BART (Bidirectional and Auto-Regressive Transformers) combines the strengths of both autoregressive and autoencoding models. Developed by Facebook AI, it is particularly effective for tasks like text generation, summarization, and translation. BART's unique training approach involves corrupting text and then reconstructing it, which enhances its ability to generate coherent and contextually appropriate responses.

BART is a powerful model designed for a wide range of natural language processing tasks, functioning as a denoising autoencoder based on a sequence-to-sequence architecture. Its pretraining involves two main stages: first, text is corrupted through various noising functions, and second, the model learns to reconstruct the original text. By leveraging a Transformer architecture that combines the bidirectional encoding of BERT and the autoregressive decoding of GPT, BART exhibits flexibility in handling arbitrary text transformations. It employs innovative noising techniques, such as random sentence shuffling and in-filling, which enhance the model's reasoning about text structure and length. BART excels in both text generation and comprehension tasks, achieving state-of-the-art results on benchmarks like GLUE and SQuAD, as well as outperforming previous models in specific applications such as summarization and dialogue generation. Additionally, BART's architecture allows for novel fine-tuning strategies, including machine translation tasks, where it is utilized as a pre-trained target-side language model, further boosting performance on translation benchmarks. Overall, BART's design enables it to deliver consistently strong results across a diverse array of

NLP challenges (Lewis, 2019).

XLNet

XLNet is a generalized autoregressive pretraining model that improves upon BERT's capabilities by capturing bidirectional context while maintaining an autoregressive mechanism. Developed by Google Brain, it utilizes permutation-based training to predict the next word in a sequence, allowing it to understand context more effectively. This makes XLNet particularly adept at tasks requiring nuanced language understanding.

XLNet is a generalized autoregressive model designed to enhance language pretraining by effectively combining the strengths of autoregressive (AR) and autoencoding (AE) methods while addressing their limitations. Unlike traditional AR models that rely on a fixed factorization order, XLNet maximizes the expected log likelihood across all possible permutations of the token sequence, allowing each token to learn from both left and right contexts. This permutation approach enables XLNet to capture bidirectional information without the need for data corruption, thereby avoiding the pretrain-finetune discrepancy seen in models like BERT. Additionally, XLNet incorporates architectural innovations from Transformer-XL, such as segment recurrence and relative encoding, which improve performance on tasks with longer text sequences. Empirical results demonstrate that XLNet consistently outperforms BERT across a variety of benchmarks, including language understanding tasks (GLUE), reading comprehension (SQuAD, RACE), text classification (Yelp, IMDB), and document ranking (ClueWeb09-B), making it a powerful tool for diverse natural language processing applications (Yang, 2019).

BlooM

BlooM is a relatively newer model that focuses on multilingual understanding and generation. Its architecture is designed to handle diverse linguistic structures and contexts, making it suitable for global applications. BlooM's ability to generate text across multiple languages while maintaining fluency and coherence positions it as a strong contender in the landscape of modern NLP models.

7 Experiments

We classified eight text generation datasets using the BERT-base model with a focus on specific features to assess whether the outputs from these datasets differ significantly across classifiers. The model was optimized using the AdamW method, with a learning rate set to 2×10^{-5}, enabling effective weight updates during training. We trained the model for 10 epochs, using a batch size of 32 to balance training speed and performance. Additionally, a weight decay of 0.01 was applied to prevent overfitting by penalizing overly complex models. This setup allowed us to rigorously evaluate the outputs of the text generation datasets, helping to determine if distinct features influence the variability and quality of generated text across different contexts. Table 2 provides the classification model and the corresponding parameter values.

	Parameters
Model	BERT-base
Optimization Method	AdamW
Learning Rate	2×10^{-5}
Number of Epochs	10
Batch Size	32
Weight Decay	0.01

Table 2: Summary of training hyperparameters and optimization method used in the experiment. The table includes details on the optimization method, learning rate, number of epochs, batch size, and weight decay.

7.1 Results and Discusion

We used Bert-base model to evaluate the perfomance of the text generation of 8 different models. During the first epoch of training, the error on the training dataset was 0.59, but this loss decreased as the number of epochs increased, indicating that the model was learning effectively. In the validation dataset, the loss decreased up to epoch 4, after which it began to increase again. This pattern suggested that the model was starting to overfit, meaning it was becoming too specialized to the training data and losing its ability to generalize to new data. To mitigate this risk, we selected epoch 4 for testing, as it provided the best balance between training accuracy and generalization. At this epoch, the validation accuracy reached 84.4%, reflecting the model's strong performance before overfitting began to occur. Accordingly, we selected epoch 4 and the test accuracy was high 85.5%, which indicates the models are performing various and distinct methods to produce the data. Figure2, 3, and 4 are showing the training and validation error, as well as validation accuaracy in different epochs

The high test accuracy at epoch 4 implies that the different text generation models are employing varied and distinct methods to produce their outputs. By analyzing the outputs through the BERT-based evaluation, we can identify specific performance patterns associated with each text generation approach, enabling us to discern their unique characteristics. This methodology allows researchers to make informed decisions about which generation methods excel in specific contexts, thereby enhancing the understanding of their relative strengths and weaknesses.

Using this model for evaluation allows for nuanced insights into the effectiveness of different text generation techniques, highlighting their individual strengths and weaknesses. The advantages of this approach include the ability to capture complex semantic patterns and measure performance quantitatively. However, the potential disadvantages involve the risk of overfitting, as seen in the validation loss trends, and the need for careful selection of training parameters to ensure generalizability. Overall, while the BERT-base model provides robust evaluation capabilities, attention must be paid to model training dynamics to maximize its effectiveness in discerning differences among text generation methods.

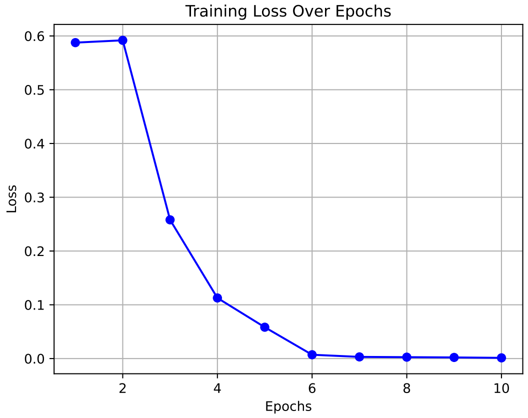

Figure 2: Training loss progression over 10 epochs during model training.

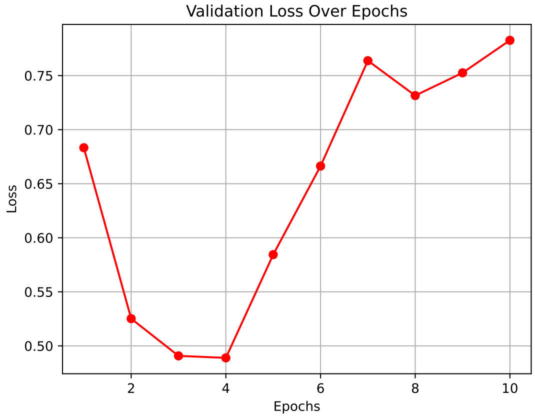

Figure 3: Validation loss over 10 epochs, reflecting model performance on the validation set.

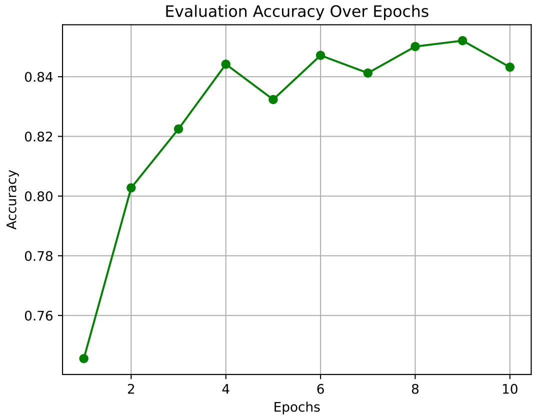

Figure 4: Evaluation accuracy plotted over 10 epochs, demonstrating the model's accuracy during evaluation.

Figure 5 presents the classification accuracy for each model separately. The results show that for models like **BART**, **T5**, **CodeGen**, and **XLNet**, we achieved accuracies of 98.69%, 96.74%, 92.66%, and 99.18%, respectively. These high accuracy scores indicate that the text generated by these models is significantly different from the text produced by other models. In particular, the high accuracy for **BART** and **XLNet** suggests that their generated text exhibits unique characteristics, making it easier for the classifier to distinguish between them and other models.

In general, the higher the accuracy for a given model, the more distinct and recognizable its text generation style is in comparison to the others. This implies that the models with the highest accuracies tend to produce text that is more easily identifiable and less similar to outputs from other models, highlighting the distinctiveness in their generative processes.

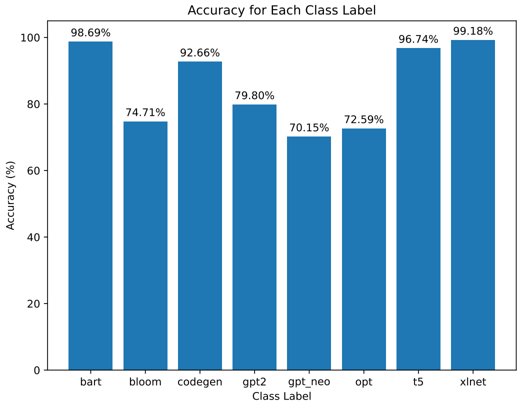

Figure 5: Accuracy of the trained model for each class.

Model	Accuracy for all classes
Bert-base	85.5%

Table 3: Table showing the overall accuracy of the trained model.

8 Conclusion

This evaluation approach based on classification not only captures complex semantic patterns but also quantifies performance, facilitating informed decisions regarding which generation methods are best suited for specific applications. While the BERT-base model offers robust capabilities, attention must be given to training dynamics, as overfitting could impact generalizability. Thus, while this methodology holds great promise for advancing the understanding of text generation techniques, careful parameter selection and monitoring of validation trends are crucial for optimizing evaluation outcomes.

References

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123–156.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. *arXiv preprint arXiv:2205.09726*.

M Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.

Kathleen McKeown. 1992. *Text generation*. Cambridge University Press.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*.

Parth Patwa, Shivam Sharma, Srinivas Pykl, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. 2021. Fighting an infodemic: Covid-19 fake news dataset. In *Combating Online Hostile Posts in Regional Languages during Emergency Situation: First International Workshop, CONSTRAINT 2021, Collocated with AAAI 2021, Virtual Event, February 8, 2021, Revised Selected Papers 1*, pages 21–29. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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. *Journal of machine learning research*, 21(140):1–67.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Zhilin Yang. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):1–38.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore:

Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*.

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.