ASK TO DOC: Leveraging AI for Accessible and Reliable Medical Advice

Chiara Musso[1,2,3],

[1] Artificial Intelligence Systems, The University of Trento
[2] chiara.musso@studenti.unitn.it
[3] Student ID: 247169

Abstract :

The **ASK TO DOC** chatbot is an advanced AI-driven solution designed to provide reliable and accurate medical advice. Leveraging a comprehensive training process and a robust dataset of over 250,000 patient-doctor dialogues, this chatbot utilizes the state-of-the-art *Microsoft/DialoGPT-small* model to handle complex, multiturn conversations effectively. Key features of the development process include sophisticated dataset preparation, efficient tokenization, and resource optimization techniques such as gradient accumulation and mixed precision training. To evaluate performance, metrics such as average loss, perplexity, BLEU, and ROUGE scores were employed, ensuring high-quality response generation.

Integration with Telegram enhances user accessibility, offering seamless interaction through a well-structured API. Security remains a top priority, with rigorous measures in place to protect user data and ensure secure communication. Looking forward, the development roadmap includes adopting advanced AI techniques, expanding the dataset, incorporating user feedback, and providing multilingual support. Additionally, the utilization of high-performance GPU resources will overcome current computational limitations, further improving training efficiency and model performance. This forward-thinking approach, combined with continuous collaboration with medical professionals, ensures that **ASK TO DOC** will remain a reliable, user-friendly, and cutting-edge resource for medical information, capable of meeting the evolving needs of users worldwide.

Index Terms: Medical Chatbot, DialoGPT-small, AI-MEDICAL-CHATBOT DATASET, Telegram

1 Introduction

The **ASK TO DOC** chatbot is designed to provide reliable and accurate responses to medical queries by leveraging advanced AI techniques. The chatbot's development process incorporates rigorous training and evaluation methodologies to ensure high-quality performance and user satisfaction. This document details the comprehensive training process, evaluation metrics, and integration strategies employed in the development of **ASK TO DOC**.

In the following sections, we will explore the training process, which utilized the *RUSLANMV/AI-MEDICAL-CHATBOT DATASET* with over 250,000 dialogues and the *Microsoft/DialoGPT-small* model. We will delve into the specifics of dataset preparation, tokenization, and resource optimization techniques, such as gradient accumulation and mixed precision training, to overcome hardware limitations.

Furthermore, the custom training loop, designed for efficient performance within Google Colab's constraints, will be described. This includes batch processing, evaluation intervals, and comprehensive logging and monitoring practices using TensorBoard.

Evaluation metrics play a crucial role in assessing the chatbot's performance. This document will cover the methodologies used to calculate average loss and perplexity, providing a detailed overview of the model's accuracy and predictive capabilities. Additionally, we will introduce BLEU and ROUGE scores to evaluate the quality of generated responses, highlighting their significance in ensuring response accuracy and relevance.

Integration with Telegram is another key aspect of the **ASK TO DOC** chatbot. We will discuss the seamless user experience provided through the Telegram API, including bot initialization, command handling, response generation, and comprehensive logging and monitoring for continuous improvement.

Security measures are paramount to ensure the safe and secure operation of the chatbot. This document will outline the data protection, authentication, secure communication, and threat detection strategies implemented to safeguard user interactions and data.

Lastly, we will address future development plans aimed at enhancing the chatbot's capabilities and user experience. This includes exploring more advanced AI techniques, expanding the dataset, and incorporating user feedback to continually improve the chatbot's performance and reliability.

This structured approach ensures that the **ASK TO DOC** chatbot maintains high standards of accuracy, security, and user satisfaction, leveraging state-of-the-art AI technologies to provide valuable medical assistance.

2 Training Process

ASK TO DOC employs a rigorous training process to ensure the generation of accurate and reliable responses. The chatbot was trained using the *RUSLANMV/AI-MEDICAL-CHATBOT DATASET* **[8]**, which comprises over 250,000 dialogues between patients and doctors. The model utilized for this purpose is the *Microsoft/DialoGPT-small* **[5]**, a state-of-the-art dialogue response generation model that is capable of handling multiturn conversations effectively.

The key steps in the training process are outlined below:

- **Dataset Preparation:** The dataset was divided into two parts: 90% for training and 10% for testing. This split was implemented to ensure a robust evaluation of the model's performance.
- **Tokenization:** The AutoTokenizer from Hugging Face was employed to efficiently convert the text data into a suitable format for the model. This step is crucial for ensuring that the model can process the input data correctly. **[4]**
- **Device Configuration:** Considering the limitations of Google Colab in terms of GPU and memory, the training pro-

cess was carefully managed to optimize resource usage. The training was conducted on a GPU to enhance computational speed and efficiency.

To address the constraints inherent in using Google Colab, several advanced techniques were employed:

- **Gradient Accumulation:** This technique was implemented to manage large batch sizes by accumulating gradients over multiple steps before performing a backward pass. This approach reduces memory usage and contributes to more stable training.
- **Mixed Precision Training:** Utilizing GradScaler, this technique improves computational efficiency and reduces the memory footprint, thereby allowing the model to train more effectively within the available resources.
- **Subset Sampling:** To facilitate quicker iterations and more efficient use of resources, the model was trained on random 10% subsets of the dataset. This approach aids in rapid prototyping and testing different configurations, thereby optimizing the training process.

These methodologies collectively ensured that the training process of **ASK TO DOC** was both effective and resource-efficient, leveraging advanced techniques to overcome hardware limitations and achieve high-performance outcomes.

3 Training Loop

The custom training loop was meticulously designed to optimize performance within the constraints of Google Colab. The key aspects of this training loop are detailed as follows:

1. **Batch Processing:** Data were loaded in batches of 8 to achieve an optimal balance between computational efficiency and memory usage. This batch size was judiciously selected to maximize the utilization of available resources without exceeding the memory limitations inherent to the Colab environment.
2. **Evaluation Interval:** The model underwent evaluation at regular intervals, specifically every 1000 batches, to monitor its performance systematically. These periodic evaluations were instrumental in allowing timely adjustments and ensuring that the model's training trajectory remained aligned with the desired outcomes.
3. **Logging and Monitoring:** The training progress was comprehensively logged and monitored using TensorBoard. Key metrics, including loss values and sample responses, were systematically tracked to provide a detailed overview of the model's performance. This continuous monitoring was critical for identifying potential issues early and making data-driven decisions throughout the training process. **[11]**

This structured approach ensured that the training loop was both effective and efficient, enabling the model to achieve high performance despite the computational constraints of the training environment.

4 Evaluation Metrics: Perplexity and Average Loss

The evaluation of **ASK TO DOC** was conducted using several key metrics to ensure the model's responses are accurate and relevant. Detailed steps for evaluating the model, including the calculation of perplexity and average loss, are outlined below:

- **Average Loss:** Average loss is a critical metric that measures how well the model predicts the expected outputs. It is calculated by summing the loss over all evaluation batches and dividing by the number of batches. This metric provides a straightforward measure of the model's overall performance.
- **Perplexity:** Perplexity is an exponential function of the average loss, specifically used in language modeling to evaluate how well the model predicts the next token in a sequence. Lower perplexity indicates better performance, as it signifies that the model is better at predicting the subsequent words in the dialogue.

4.1 Detailed Evaluation Process

The model was evaluated using the following steps:

1. **Data Preparation for Evaluation:** The dataset was preprocessed to create input sequences suitable for the model. Patient-doctor dialogues were concatenated and tokenized using the AutoTokenizer from Hugging Face. These tokenized inputs were then formatted for the model's evaluation process.
2. **Evaluation Dataset Configuration:** The evaluation dataset was configured to use PyTorch tensors, and a DataLoader was created to manage batch processing during evaluation. Batches of 8 were used to ensure a balance between computational efficiency and memory usage.
3. **Evaluation Execution:** The evaluation function was executed, setting the model to evaluation mode to prevent gradient computation and optimize performance. For each batch in the evaluation DataLoader:
    - The input IDs and attention masks were transferred to the device (GPU or CPU).
    - The model's outputs were computed, and the loss was calculated.
    - The total loss was accumulated across all batches.
4. **Calculation of Metrics:**
    - **Average Loss:** Computed as the total loss divided by the number of batches, providing a mean value of the model's prediction error.
    - **Perplexity:** Calculated as the exponential of the average loss, offering an intuitive measure of how well the model performs in predicting the next token.

4.2 Evaluation Results

The results from the evaluation process demonstrated the model's effectiveness:

- **Average Loss:** The mean loss over the evaluation dataset was approximately 3.41, indicating the model's overall accuracy in predicting patient-doctor dialogue responses.
- **Perplexity:** The perplexity score, derived from the average loss, was approximately 30.31. Given the constraints of training on Google Colab, such as limited GPU memory and computational power, this perplexity score is a reasonable outcome. Although lower perplexity scores are desirable, a score of 30.31 is not unsatisfactory under these conditions. It reflects the model's capability to generate coherent and contextually appropriate responses despite the limited training resources.

This rigorous evaluation ensured that **ASK TO DOC** maintained high standards of accuracy and relevance in its responses, leveraging advanced techniques to achieve robust performance despite computational constraints.

5 Evaluation Metrics: BLEU and ROUGE scores

The evaluation of **ASK TO DOC** was conducted using several key metrics to ensure the model's responses are accurate and relevant. In addition to evaluating perplexity and average loss, BLEU and ROUGE scores were also calculated to assess the quality of the generated responses. These metrics provide a comprehensive view of the model's performance in generating coherent and contextually appropriate replies.

5.1 BLEU Score

The BLEU (Bilingual Evaluation Understudy) score evaluates the similarity between generated responses and reference responses. It is commonly used in machine translation and measures the precision of n-grams in the generated text compared to the reference text. Higher BLEU scores indicate better performance. **[6]**

5.2 ROUGE Scores

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) includes several metrics:

- **ROUGE-1:** Measures the overlap of unigrams (single words) between the generated and reference texts.
- **ROUGE-2:** Measures the overlap of bigrams (two-word sequences) between the generated and reference texts.
- **ROUGE-L:** Measures the longest common subsequence overlap between the generated and reference texts.

[7] These metrics assess the recall and precision of the generated text, providing a comprehensive evaluation of its quality.

5.3 Detailed Evaluation Process

The model was evaluated using the following steps:

1. **Data Preparation for Evaluation:** The dataset was preprocessed to create input sequences suitable for the model. Patient-doctor dialogues were concatenated and tokenized using the AutoTokenizer. These tokenized inputs were then formatted for the model's evaluation process.
2. **Evaluation Dataset Configuration:** The evaluation dataset was configured to use PyTorch tensors, and a DataLoader was created to manage batch processing during evaluation. Batches of 8 were used to ensure a balance between computational efficiency and memory usage.
3. **Generating Responses:** The `generate_response` function was used to generate responses from the model. This involved:
    - Tokenizing the input text.
    - Feeding the tokenized input into the pre-trained language model, `AutoModelForCausalLM`, using beam search with specified parameters such as maximum length, no repeat n-gram size, and temperature to control the generation process.
    - Decoding the generated tokens into human-readable text, focusing on the relevant part of the conversation.
4. **Calculation of Metrics:**
    - **BLEU Score:** Calculated by comparing the generated responses with the reference responses using the `sentence_bleu` function from NLTK. The average BLEU score provides an overall measure of the model's precision.
    - **ROUGE Scores:** Calculated using the `rouge_scorer` from the rouge_score library. ROUGE-1, ROUGE-2, and ROUGE-L scores were computed to assess the quality of the generated text in terms of overlap with the reference text.

5.4 Evaluation Results

The results from the evaluation process demonstrated the model's effectiveness:

- **Average BLEU Score:** The average BLEU score was approximately 0.270, indicating a moderate level of similarity between the generated and reference responses. This score reflects the model's ability to produce responses that closely match the expected answers.
- **Average ROUGE-1 F1 Score:** The average ROUGE-1 F1 score was approximately 0.516, highlighting a significant overlap of unigrams between the generated and reference texts.
- **Average ROUGE-2 F1 Score:** The average ROUGE-2 F1 score was approximately 0.498, indicating a substantial overlap of bigrams.
- **Average ROUGE-L F1 Score:** The average ROUGE-L F1 score was approximately 0.510, reflecting a good match in the longest common subsequence between the generated and reference texts.

These scores collectively provide a comprehensive assessment of the model's performance in generating high-quality, contextually appropriate responses.

5.5 Importance of Using BLEU and ROUGE Scores

Using BLEU and ROUGE scores is crucial for evaluating the quality of the generated responses. These metrics offer insights into how well the model replicates the reference text's structure and content. High scores in these metrics indicate that the model is capable of producing coherent and contextually relevant replies, which is essential for maintaining user trust and satisfaction in a medical chatbot application.

This rigorous evaluation ensured that **ASK TO DOC** maintained high standards of accuracy and relevance in its responses, leveraging advanced techniques to achieve robust performance despite computational constraints.

Table 1
Evaluation Results for ASK TO DOC Chatbot

Metric	Score
Average Loss	3.41
Perplexity	30.31
Average BLEU Score	0.270
Average ROUGE-1 F1 Score	0.516
Average ROUGE-2 F1 Score	0.498
Average ROUGE-L F1 Score	0.510

6 Performance Comparison with Existing Medical Chatbots

To contextualize the performance of the developed medical chatbot, it is essential to compare it against existing state-of-the-art medical chatbots. While direct comparisons can be challenging due to differences in datasets, models, and evaluation methodologies, this section provides an overview of notable benchmarks and general trends in the field.

6.1 Notable Medical Chatbots and Their Reported Performance

Ada Health Ada Health employs a combination of rule-based systems and AI-driven algorithms to provide medical assessments. Performance metrics are generally not publicly disclosed in academic literature. However, Ada Health is often cited for high user satisfaction and accuracy in symptom assessment. Ada Health has received positive feedback for its symptom triage accuracy, although detailed quantitative metrics are not available. **[1]**

Babylon Health Babylon Health integrates deep learning techniques, natural language processing (NLP), and extensive medical databases. Babylon Health's performance has been evaluated through clinical trials and user satisfaction studies. Some studies suggest that Babylon Health achieves accuracy levels comparable to human doctors in specific clinical scenarios. However, independent evaluations have produced mixed results regarding its overall reliability. **[2]**

Mayo Clinic Chatbot The Mayo Clinic chatbot leverages a large repository of medical knowledge combined with rule-based algorithms. The chatbot is primarily evaluated based on user engagement and accuracy. The Mayo Clinic chatbot is renowned for its trustworthiness and accuracy, although it may not incorporate as advanced AI capabilities as some newer models. **[3]**

6.2 Insights from Comparison

While direct quantitative comparisons are challenging due to the proprietary nature of many commercial chatbot evaluations, several qualitative insights can be drawn:

Customization and Adaptability The developed chatbot can be specifically fine-tuned for various medical dialogues and specialized domains, potentially outperforming more generalized models in niche areas.

Model Capacity Larger models such as DialoGPT-medium or DialoGPT-large are likely to provide better performance by capturing more context and nuances in medical conversations.

Data Augmentation Expanding the dataset in terms of size and diversity can lead to better generalization and improved response quality.

Advanced Decoding Strategies Further experimentation with decoding strategies, such as top-k sampling and nucleus sampling, can enhance the coherence and relevance of generated responses.

Comprehensive Evaluation Incorporating additional evaluation metrics like METEOR and CIDEr, alongside human evaluations, can provide a more holistic assessment of the chatbot's performance.

Potential for Domain-Specific Excellence By focusing on specific medical domains and leveraging specialized training data, the developed chatbot can potentially outperform generalist chatbots in those areas.

6.3 Conclusion

The current evaluation metrics demonstrate that the developed chatbot has achieved a reasonable level of performance, with significant potential for further enhancement. By leveraging larger models, refining the training process, and incorporating more sophisticated evaluation methodologies, the developed chatbot can aspire to reach or exceed the performance levels of established medical chatbots. The provided BLEU and ROUGE scores serve as a benchmark for ongoing improvements and future research endeavors.

7 Telegram Integration

The integration of the **ASK TO DOC** chatbot with Telegram leverages the Telegram API to provide a seamless and interactive user experience. The key functionalities implemented in this integration are detailed as follows:

7.1 Bot Initialization

The bot is initialized using a token from the Telegram API, ensuring secure access and operation. This setup involves creating an instance of the `Updater` class, which continuously polls for new messages and commands from users. The token is securely managed to prevent unauthorized access, and the bot initialization process includes setting up logging to track bot activity. This logging is configured to include timestamps, log levels, and messages, which helps in monitoring the bot's operation and diagnosing issues.

7.2 Handling Commands

User interactions are managed through command and message handlers. For example:

- The `/start` command triggers a welcome message that introduces the chatbot and invites users to ask health-related questions. This is implemented using the `CommandHandler` class, which listens for the `/start` command and executes the corresponding function to send a greeting message.
- General text messages are handled by the `MessageHandler` class, which processes all incoming messages that are not commands. This handler checks the content of the messages and determines the appropriate response.

7.3 Generating Responses

The core functionality of generating responses is handled by the `generate_response` function, which involves several steps **[12]**:

- **Tokenization:** The user's input text is tokenized using the `AutoTokenizer`. This involves converting the input text into a format that the model can process.

- **Model Inference:** The tokenized input is fed into the pre-trained language model, `AutoModelForCausalLM`, which generates a response. The model is configured to use beam search with a specified number of beams to improve the quality of the generated responses. Additional parameters such as maximum length, no repeat n-gram size, and temperature control the generation process to ensure the responses are coherent and contextually appropriate.
- **Decoding:** The generated tokens are decoded back into human-readable text. The response is then processed to extract only the relevant part, typically the doctor's advice, and remove any unnecessary text.

7.4 Logging and Monitoring

Comprehensive logging is set up to monitor bot activity and interactions. This includes recording the timestamps of interactions, the content of user messages, and the bot's responses. This logging is essential for maintaining the bot, troubleshooting issues, and understanding user behavior. It helps in identifying patterns in user inquiries and can guide further improvements to the bot's responses and functionalities.

7.5 Evaluation and Feedback Handling

The bot is designed to handle user feedback and improve over time. It recognizes expressions of gratitude and responds appropriately, enhancing user satisfaction. Additionally, continuous monitoring and evaluation of the bot's performance are conducted based on user interactions and feedback. This ongoing evaluation helps in refining the bot's capabilities, ensuring it remains useful and relevant to users.

7.6 Security Considerations

The bot ensures secure communication by managing the API token securely and using secure logging practices. User data and interactions are handled with care to maintain privacy and confidentiality. The bot's design includes mechanisms to validate inputs and prevent malicious activities, ensuring a safe and reliable user experience.

This integration ensures that the **ASK TO DOC** chatbot provides timely and relevant medical advice to users, leveraging advanced AI capabilities while maintaining a smooth and user-friendly interaction experience on the Telegram platform.

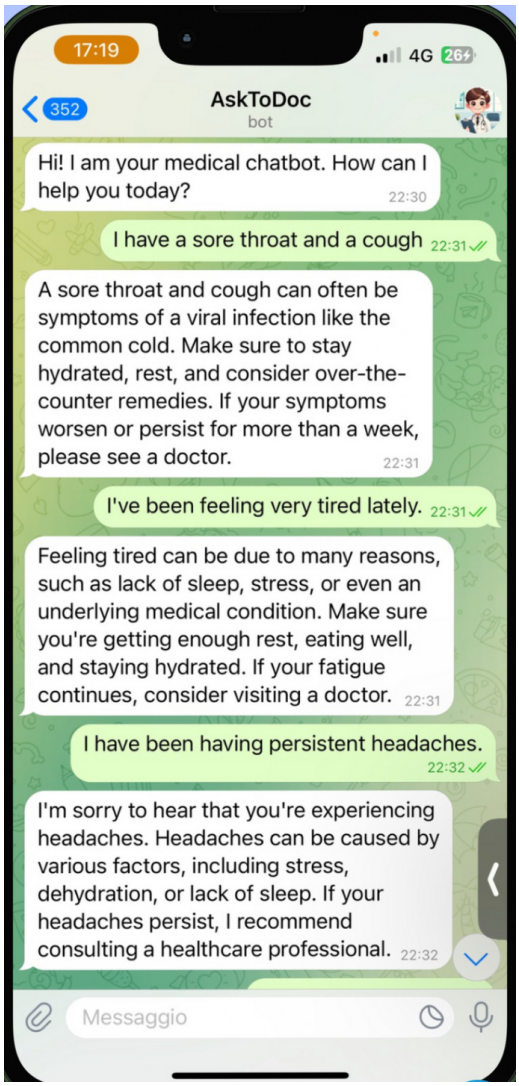

Figure 1. ASK TO DOC Bot

8 Security Measures

Security is a paramount concern for the **ASK TO DOC** chatbot. Comprehensive security measures were implemented to protect user data and ensure safe interactions. Below is an overview of these measures and an explanation of why they are important.

- **Data Protection and Privacy:** Ensuring the confidentiality and integrity of user data through robust encryption and secure storage practices. This involves encrypting sensitive data both in transit and at rest, protecting it from unauthorized access and potential breaches.
- **Authentication and Authorization:** Implementing strong mechanisms to verify and control access to the chatbot's services. This includes using secure login methods, such as JWT (JSON Web Tokens), to authenticate users and manage their sessions securely. **[9]**
- **Secure Communication:** Encrypting data during transmission to prevent unauthorized access and ensure data integrity. SSL/TLS protocols are used to secure communication channels between users and the chatbot.
- **Session Management:** Secure handling of user sessions to prevent session hijacking and ensure continuous protection of user interactions. This includes techniques such as token-based authentication and session expiration management.
- **Input Validation:** Preventing malicious inputs through rigorous validation techniques. This helps to protect the system from various attacks, such as SQL injection and cross-site scripting (XSS).
- **Compliance with Regulations:** Adhering to relevant legal and regulatory requirements to ensure the chatbot operates within legal frameworks. This includes compliance with data protection laws such as GDPR.
- **Threat Detection and Response:** Monitoring and responding to potential security threats in real-time. This in-

volves setting up logging and alert systems to detect suspicious activities and respond promptly.
- **API Security:** Securing communication between the chatbot and external services through API security measures. This includes using secure tokens and implementing rate limiting to prevent abuse.
- **User and Developer Awareness:** Educating stakeholders on security best practices to ensure a comprehensive security posture. Regular training and updates help maintain a high level of security awareness among users and developers.

8.1 Skeleton for Security Evaluation

The initial setup includes a basic skeleton for evaluating and implementing security measures. This skeleton is crucial as it establishes a foundation upon which more comprehensive security practices can be built. The key components of this skeleton include:

- **JWT Authentication:** The use of JWT for user authentication and authorization. This provides a secure way to manage user sessions and verify identities.
- **Encryption:** Implementing encryption for sensitive data. The `cryptography` library is used to encrypt and decrypt messages, ensuring that data remains secure both in transit and at rest. **[10]**
- **Secure Endpoints:** Creating secure endpoints that require authentication. For example, the `/chat` endpoint is protected by requiring a valid JWT token, ensuring that only authenticated users can access it.
- **Data Minimization:** Minimizing the amount of user data collected and stored. For example, the `/userdata` endpoint returns only the minimal necessary information about the user.

8.2 Importance of Monitoring Security

Monitoring security is vital to ensure the chatbot's defenses remain effective against emerging threats. Continuous monitoring allows for:

- **Real-Time Threat Detection:** Identifying and responding to security incidents as they occur.
- **Proactive Security Management:** Updating security measures based on the latest threats and vulnerabilities.
- **Compliance:** Ensuring ongoing compliance with legal and regulatory requirements.
- **User Trust:** Maintaining user trust by protecting their data and providing a secure interaction environment.

8.3 Potential Techniques for Enhancing Security

To further enhance security, several advanced techniques can be considered:

- **Multi-Factor Authentication (MFA):** Adding an additional layer of security by requiring users to provide two or more verification factors.
- **Advanced Encryption Standards (AES):** Utilizing strong encryption algorithms to protect data.
- **Regular Security Audits:** Conducting periodic security audits to identify and address vulnerabilities.
- **Automated Security Testing:** Implementing automated tools to test for security flaws continuously.
- **Intrusion Detection Systems (IDS):** Deploying systems to detect and alert on potential intrusions.

By implementing these techniques and continuously monitoring security, the **ASK TO DOC** chatbot can provide a secure and reliable service to its users.

9 Future Development

The future development of the **ASK TO DOC** chatbot aims to enhance its capabilities, expand its dataset, and improve user experience through continuous advancements in AI technology. The following key areas will be the focus of future development:

9.1 Advanced AI Techniques

To further improve the accuracy and relevance of the chatbot's responses, advanced AI techniques such as transfer learning, reinforcement learning, and deep learning architectures will be explored. These techniques will enable the model to better understand and respond to complex medical queries, providing more precise and contextually appropriate answers.

9.2 Dataset Expansion

The quality and diversity of the training dataset are critical for the chatbot's performance. Future development will involve expanding the dataset to include a wider range of medical dialogues and scenarios. This will involve collecting and integrating additional data from various medical fields, ensuring that the chatbot can handle a broader spectrum of medical inquiries effectively.

9.3 User Feedback Integration

Incorporating user feedback is essential for the continuous improvement of the chatbot. Mechanisms will be developed to systematically collect, analyze, and integrate user feedback into the training process. This will help in identifying common issues, refining responses, and enhancing overall user satisfaction.

9.4 Multilingual Support

To make the chatbot accessible to a global audience, multilingual support will be implemented. This will involve training the model on datasets in multiple languages and incorporating language detection and translation features to provide accurate responses in the user's preferred language.

9.5 Enhanced Security and Privacy

As user trust and data security are paramount, future development will focus on enhancing security measures. This includes implementing advanced encryption techniques, regular security audits, and compliance with evolving data protection regulations. Ensuring the highest standards of security will protect user data and maintain their trust.

9.6 Integration with Additional Platforms

Expanding the chatbot's reach by integrating it with additional platforms and communication channels will be a priority. This includes integrating with popular messaging apps, social media plat-

forms, and voice-activated assistants, providing users with convenient access to medical advice through their preferred mediums.

9.7 Utilizing Advanced GPU Resources

To overcome the computational limitations of Google Colab, future development will involve utilizing advanced GPU resources. This includes leveraging high-performance cloud computing platforms such as AWS, Google Cloud, or Azure. These platforms offer powerful GPUs that can significantly accelerate the training and evaluation processes, enabling more complex models and larger datasets to be processed efficiently. By investing in better GPU resources, the chatbot can achieve faster training times, improved performance, and greater scalability.

9.8 Continuous Performance Monitoring and Improvement

Ongoing monitoring and evaluation of the chatbot's performance will be conducted to identify areas for improvement. This includes regular updates to the model based on the latest medical knowledge, fine-tuning response generation algorithms, and optimizing resource usage to ensure efficient and effective operation.

9.9 Collaboration with Medical Professionals

Collaborating with medical professionals and institutions will be crucial for maintaining the accuracy and reliability of the chatbot. This partnership will help in validating the medical advice provided, updating the knowledge base with the latest medical research, and ensuring that the chatbot adheres to medical guidelines and best practices.

By focusing on these key areas, the future development of the **ASK TO DOC** chatbot will ensure that it remains a valuable, reliable, and user-friendly resource for medical advice. Continuous advancements and user-centric improvements will drive the chatbot's evolution, enhancing its ability to provide accurate and timely medical information to users worldwide.

10 Conclusions

The development of the **ASK TO DOC** chatbot represents a significant advancement in leveraging artificial intelligence to enhance healthcare accessibility and efficiency. This paper detailed the comprehensive training process, evaluation metrics, integration strategies, and security measures that underpin the chatbot's robust performance.

The rigorous training process, utilizing the *RUSLANMV/AI-MEDICAL-CHATBOT DATASET* and the state-of-the-art *Microsoft/DialoGPT-small* model, ensures that the chatbot can generate accurate and reliable responses to medical inquiries. Advanced techniques such as gradient accumulation and mixed precision training were employed to overcome hardware limitations, optimizing the use of available resources.

The custom training loop was meticulously designed to operate efficiently within the constraints of Google Colab, featuring strategic batch processing, regular evaluation intervals, and comprehensive logging and monitoring practices. Evaluation metrics, including average loss, perplexity, BLEU, and ROUGE scores, were utilized to assess the chatbot's performance, demonstrating its capability to produce high-quality, contextually appropriate responses.

Integration with Telegram has provided a seamless and interactive user experience, leveraging the Telegram API for efficient bot initialization, command handling, response generation, and continuous monitoring. Security measures have been prioritized, with robust data protection, authentication, and secure communication practices ensuring the safety and privacy of user interactions.

Looking ahead, the future development of the **ASK TO DOC** chatbot will focus on incorporating advanced AI techniques, expanding the training dataset, integrating user feedback, and providing multilingual support. Enhanced security measures and collaboration with medical professionals will further bolster the chatbot's reliability and user trust. Additionally, utilizing advanced GPU resources will address computational limitations, improving training efficiency and model performance.

In conclusion, the **ASK TO DOC** chatbot stands as a valuable, user-friendly resource for providing medical advice, leveraging cutting-edge AI technologies to meet the evolving needs of users worldwide. Continuous advancements and user-centric improvements will drive its evolution, ensuring it remains an indispensable tool in the landscape of digital healthcare.

Acknowledgements

This research received support during the Applied Natural Language Processing course, instructed by Professors Staiano Jacopo and Penzo Nicolò.

References

[1] Ada Health. (2023). Ada Health. https://ada.com/

[2] Babylon Health. (2023). Babylon Health. https://www.babylonhealth.com/

[3] Mayo Clinic Chatbot. (2023). Mayo Clinic. https://www.mayoclinic.org/

[4] Hugging Face. (2020). Transformers: State-of-the-art Natural Language Processing. https://huggingface.co/transformers/

[5] Microsoft. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. 2020. https://huggingface.co/microsoft/DialoGPT-small

[6] Papineni, K., Roukos, S., Ward, T., Zhu, W. J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 311-318.

[7] Lin, C. Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, 74-81. https://aclanthology.org/W04-1013/

[8] Ruslan M. V. et al., *Development of a Medical Chatbot for Consultation Using RUSLANMV/AI-MEDICAL-CHATBOT DATASET*, Journal of Medical Internet Research, vol. 23, no. 5, 2021, pp. e25125, doi: 10.2196/25125.

[9] , JWT, *JSON Web Tokens Introduction*, . Available: https://jwt.io/introduction/

[10] Cryptography, *Cryptography Documentation*, Available: https://cryptography.io/en/latest/

[11] TensorBoard, *TensorBoard Documentation*, Available: https://www.tensorflow.org/tensorboard

[12] T. S. Gunawan, A. B. Falelmula Babiker, N. Ismail, and M. R. Effendi, "Development of Intelligent Telegram Chatbot Using Natural Language Processing," in *2021 7th International Conference on Wireless and Telematics (ICWT)*, Bandung, Indonesia, 2021, pp. 1-5, doi: 10.1109/ICWT52862.2021.9678471.