ASK TO DOC

Your friendly health buddy on Telegram
Get advice on symptoms, treatments, and wellness

Made by:
Chiara Musso

OVERVIEW

Training

Evaluation

Security

Telegram

TRAINING DETAILS

DATASET AND MODEL

RUSLANMV/AI-MEDICAL-CHATBOT DATASET

This is an experimental Dataset designed to run a Medical Chatbot It contains at least 250k dialogues between a Patient and a Doctor.

MICROSOFT/DIALOGPT-SMALL

DialoGPT is a SOTA large-scale pretrained dialogue response generation model for multiturn conversations

TRAINING SETUP

- **Dataset Preparetion:** Split it into 90% training and 10% testing.
- **Tokenization**: Used AutoTokenizer from Hugging Face
- **Device Configuration**: Leveraged GPU for faster training.
- **Optimizer and Scheduler**: Implemented AdamW optimizer and linear learning rate scheduler.
- **Data Preparation**: Created a custom function to format patient-doctor conversations.
- **DataLoader**: Set up for training and evaluation with a batch size of 8.

TRAINING LOOP

- Used gradient accumulation to manage large batch sizes and reduce memory usage.
- Implemented mixed precision training with GradScaler for efficiency.
- Trained on random 10% subsets for quicker iterations.
- Tracked loss curves and metrics.

EVALUATION

METRICS

Why Evaluate?:

- To ensure the model's responses are accurate and relevant.
- To identify areas for improvement and validate the model's performance.

Perplexity

Measures how well the model predicts the next token in a sequence. Lower values indicate better performance.

BLUE score

Evaluates the similarity between generated responses and reference responses

ROGUE score

ROUGE-1: Measures overlap of unigrams between generated and reference texts.
ROUGE-2: Measures overlap of bigrams.
ROUGE-L: Measures the longest common subsequence overlap.

RESULTS

Perplexity	30.31
Averege Loss	3.41
Average BLUE score	0.270
ROGUE 1	0.516
ROGUE 2	0.498
ROUGUE L	0.510

EXISTING MEDICAL CHATBOT

1. **Ada Health**
 - Combines rules and AI
 - High user satisfaction
 - Accurate symptom triage
 - No detailed metrics available

2. **Babylon Health**
 - Uses deep learning and NLP
 - Accuracy comparable to human doctors in some cases
 - Mixed results from independent evaluations

3. **Mayo Clinic Chatbot**
 - Uses extensive medical knowledge and rules
 - Known for trustworthiness and accuracy
 - Less advanced AI capabilities

Comparison Insights:

1. **Customization**: Fine-tuning for specific medical areas outperforms general models.
2. **Model Size**: Larger models capture more context and nuances.
3. **Data Augmentation**: Diverse, larger datasets improve response quality.
4. **Decoding Strategies**: Top-k sampling enhances response coherence and relevance.
5. **Comprehensive Evaluation**: Metrics like METEOR, CIDEr, and human feedback provide a better performance overview.

SECURITY

SECURITY EVALUATION

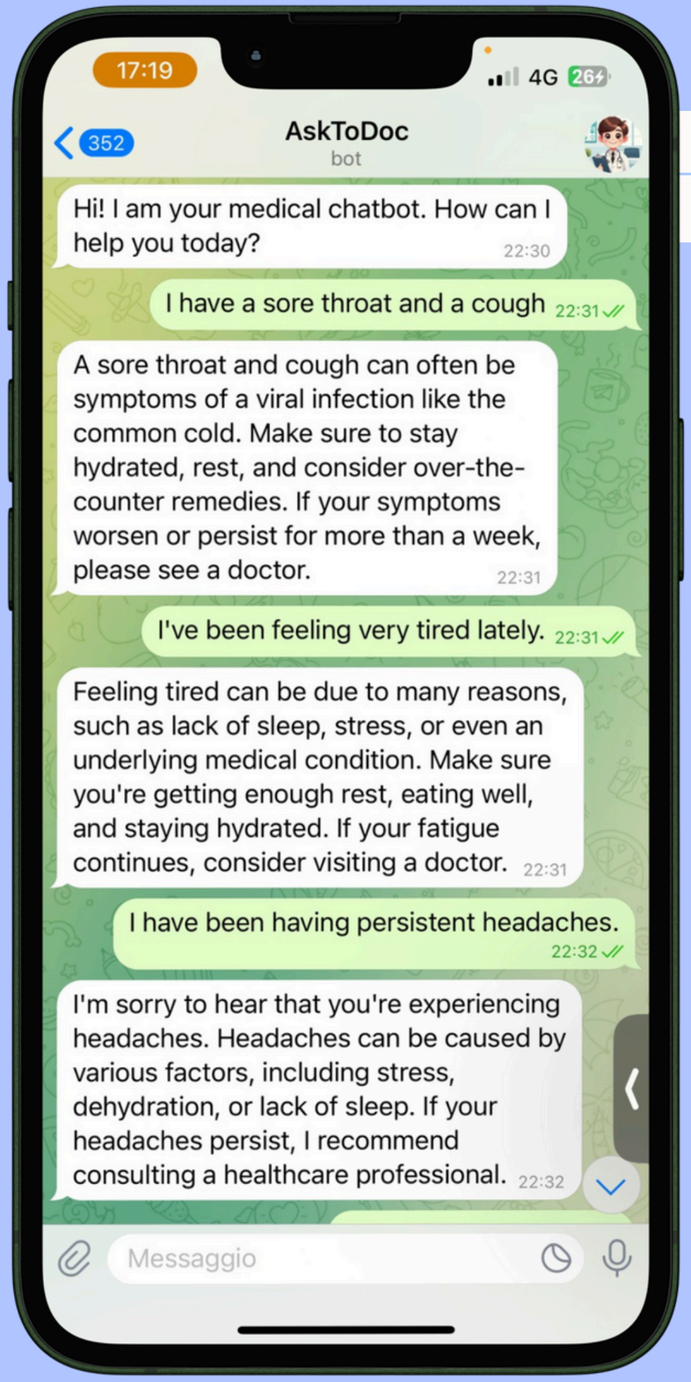
- Data Protection and Privacy
- Authentication and Authorization
- Secure Communication:
- Session Management
- Input Validation
- Storage and Handling of Personal Data
- Compliance with Regulations
- Threat Detection and Response
- API Security
- User and Developer Awareness

SPECIFIC TECHNIQUES AND PRACTICAL IMPLEMENTATIONS

- Regular Expressions and Pattern Matching: To detect and mask personal data.
- Audit Trails: Maintain detailed audit trails of user interactions and data access.

- Secure Backend Infrastructure: Ensure backend servers are secure and data is encrypted.
- Compliance with Data Retention Policies: Align data retention with legal requirements and user agreements.
- User Consent Management: Clearly inform users about data usage and obtain explicit consent.

TELEGRAM



@AskToDoc

HOW IT WORKS?

1. Bot Initialization
   - The bot is set up with a token from the Telegram API.
   - Command and message handlers are registered to manage different types of user interactions.
2. Handling Commands
   - Start Command: When a user sends /start, the bot sends a greeting message.
   - Message Response: The bot processes user messages, checks for keywords, and generates appropriate responses.
3. Generating Responses
   - User input is tokenized and fed into the model.
   - The model generates a response based on the input.

```python
def generate_response(patient_text):
    # Tokenize the input text
    input_text = f"Patient: {patient_text} Doctor:"
    input_ids = tokenizer.encode(input_text, return_tensors='pt').to(device)

    # Generate a response from the model
    with torch.no_grad():
        output_ids = model.generate(
            input_ids,
            max_length=150,
            num_beams=5,
            no_repeat_ngram_size=3,
            early_stopping=True,
            pad_token_id=tokenizer.eos_token_id,
            temperature=0.7,
            top_k=50,
            top_p=0.95,
            do_sample=True  # Enable sampling
        )
```

FUTURE DEVELOPMENT

- Advanced AI Techniques
- Dataset Expansion
- User Feedback Integration
- Multilingual Support
- Enhanced Security and Privacy
- Integration with Additional Platforms
- Utilizing Advanced GPU Resources
- Continuous Performance Monitoring and Improvement
- Collaboration with Medical Professionals

THANK YOU FOR YOUR ATTENTION