technical analysis

Contents

"Chatbots are important because you won't feel stupid asking important questions. Sometimes talking to someone can be a bit intimidating. Talking to a chatbot makes that a lot easier!"
— Petter Bae Brandtzaeg, project leader, Social Health Bots project

1 Introduction

Key Concept

When a user sends a message to a chatbot, the system executes a sophisticated multi-stage process:

[leftmargin=*]**Natural Language Understanding**: Analyzing the text to extract meaning, intent, and context. **Response Generation**: Using AI models to select or generate the appropriate response. **Knowledge Integration**: Accessing external information or medical databases via APIs.

When a user sends a message to a chatbot, they receive a response almost instantly. However, behind this seemingly simple interaction, a sophisticated process takes place where several technologies work in synergy. Upon receiving the message, the chatbot analyzes the text using advanced Natural Language Processing (NLP) techniques. Then, using artificial intelligence (AI) models, it selects or generates the most appropriate response. Finally, integrations via APIs allow access to external information or medical databases to provide accurate and up-to-date answers. Understanding these technical aspects is essential to unraveling the inner workings of the chatbot and designing a system capable of effectively responding to user needs.

2 AI Models

Several AI models are used in the development of medical chatbots, mainly BERT and GPT.

2.1 BERT (Bidirectional Encoder Representations from Transformers)

Technical Detail

Architectural Components:

- Bidirectional context processing.
- Stack of Transformer encoders (multiple layers).
- Multi-head self-attention mechanism.
- Pre-training objectives: Masked LM and Next Sentence Prediction.

BERT is a deep learning model that processes text bidirectionally, considering both previous and subsequent tokens to better understand complex medical queries.

After tokenization, a user query is represented as:

$$X = [x_1, x_2, \ldots, x_n]$$

where x_i represents the i^{th} token.

The attention mechanism in BERT is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V$$

with Q (Query), K (Key), and V (Value) derived from the tokens, and d_k being the dimension of the key vectors.

Multi-head attention is computed via:

$$MH(Q, K, V) = \text{Concatenate}(h_1, h_2, \ldots, h_h) \cdot W_O$$

where $h_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ is the output of the i^{th} head, and W_O is the output projection matrix.

BERT produces a contextualized embedding H for an input X:

$$H = \text{BERT}(X)$$

These embeddings are essential for understanding medical queries.

A Transformer encoder layer is formalized as:

$$\text{EncoderLayer}(X) = \text{LayerNorm}\Big(X + \text{MultiHead}(X, X, X)\Big)$$

For task-specific outputs (e.g., entity recognition or intent classification), H is mapped to:

$$\text{Entities} = \text{OutputLayer}(H)$$

and

$$\text{Intent} = \text{softmax}(pool(H))$$

The fine-tuning process minimizes the loss:

$$Loss = -\sum_i Y_i \cdot \log(\hat{Y}_i)$$

by solving:

$$\min_{\theta_{\text{BERT}}} Loss(X, Y; \theta_{\text{BERT}})$$

using the Adam optimizer.

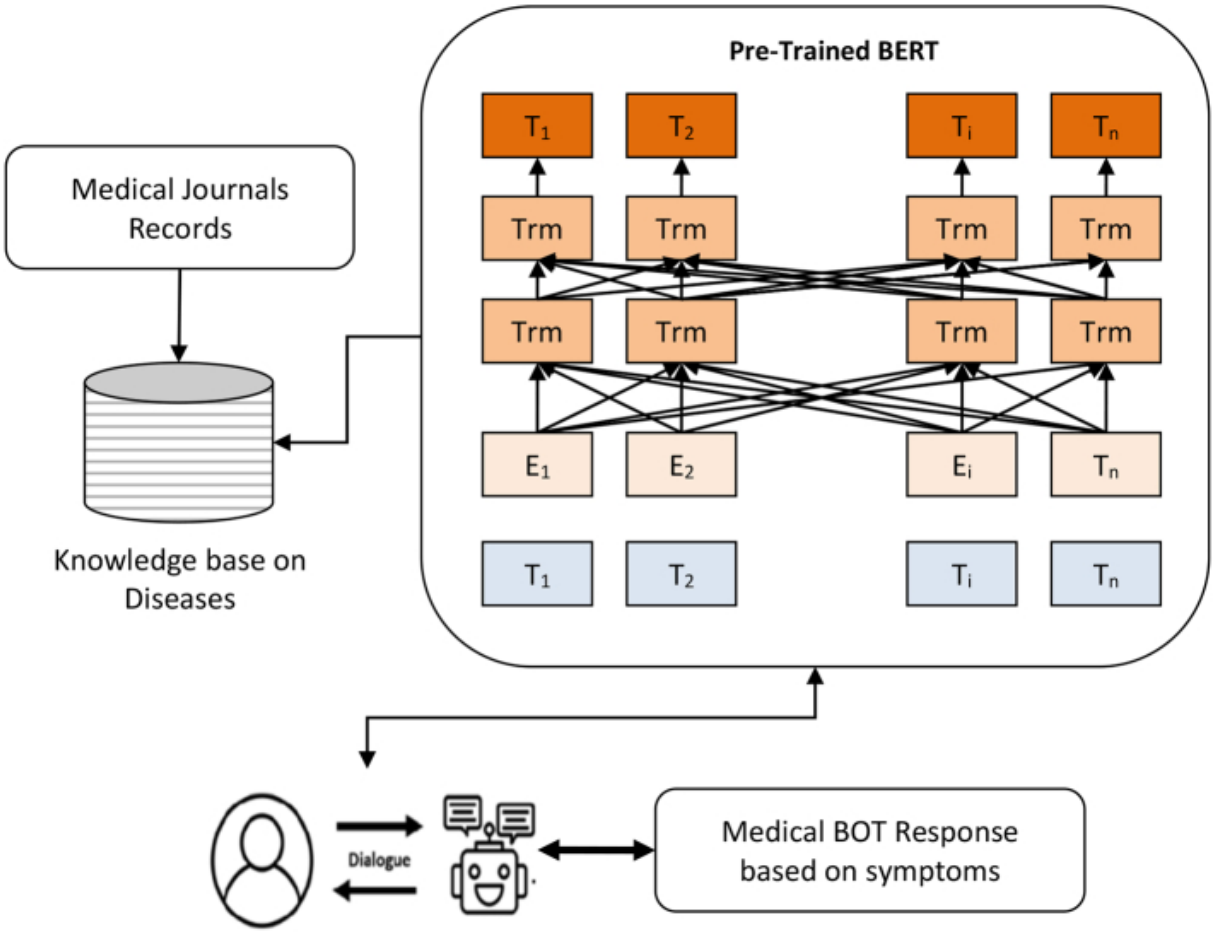

Figure 1: An example image

2.2 GPT (Generative Pretrained Transformer)

Technical Detail

Key Differentiators from BERT:

- Unidirectional context processing.
- Autoregressive text generation.
- Causal attention masking.
- Large number of parameters (e.g., 175B in GPT-3).

Unlike BERT, GPT processes text unidirectionally (left-to-right) and predicts one word at a time:

$$P(w_t \mid w_1, w_2, \ldots, w_{t-1}) = \text{softmax}(W \cdot h_t)$$

where w_t is the word to predict at position t and h_t is the hidden state.
The attention mechanism in GPT, with causal masking, is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V$$

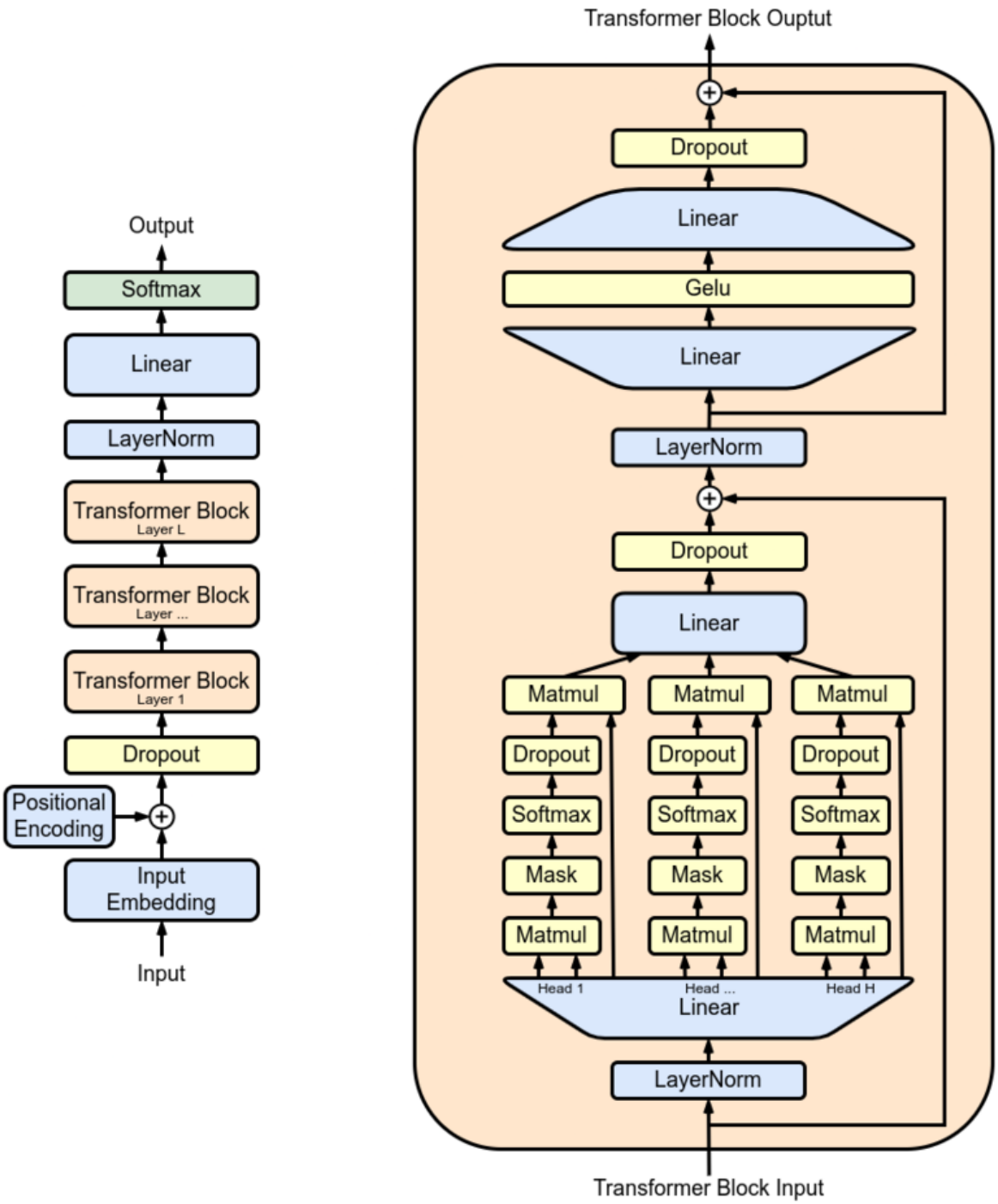

Figure 2: An example image

GPT uses autoregressive decoding, generating text based on previously predicted words. For task-specific tuning, it is optimized with the loss:

$$\text{Loss} = -\sum Y_i \cdot \log(\hat{Y}_i)$$

3 Models for Specific Tasks in Medical Domains

For specialized medical applications, domain-specific pre-trained models are used to better handle technical vocabulary and context.

3.1 PaLM (Pathways Language Model)

PaLM is a model developed by Google AI with up to 540 billion parameters. It performs various tasks such as logical reasoning, arithmetic, joke explanation, code generation, and translation using its chain-of-thought capabilities. Launched in April 2022 and later made available via API, a specialized version, Med-PaLM, was trained on medical data—surpassing previous systems and even passing US medical exams.

Google later expanded PaLM with PaLM-E for multimodal tasks and, in May 2023, launched PaLM 2 with 340 billion parameters. AudioPaLM, introduced in June 2023, enables automatic translation of spoken language.

The loss function for Med-PaLM is:

$$\mathcal{L}_{\text{med}} = -\sum_{i=1}^{N} \log P(y_i \mid x, y_{<i})$$

3.2 ChatDoctor

Technical Detail

Enhancements over Base Models:

- Fine-tuning on 100K medical dialogues.
- Based on Meta AI's LLaMA architecture.
- Real-time information retrieval mechanism.
- Integration of a symptom-checking system.

ChatDoctor is an advanced medical chatbot designed to provide accurate, personalized medical advice. It has been fine-tuned on a large dataset of 100,000 dialogues between patients and doctors from an online consultation platform, thereby enhancing its ability to understand patient needs and deliver informed advice. Additionally, it incorporates an autonomous information retrieval mechanism for real-time access to reliable sources (e.g., Wikipedia and medical databases).

4 NLP Techniques in Medical Chatbots

Natural Language Processing (NLP) transforms human language—often imprecise and varied—into structured data that can be used by artificial intelligence models. In a medical context, precision and the ability to extract specific information (such as symptoms, medications, or diagnoses) are essential to provide relevant and reliable responses. Tools like the Google Cloud Natural Language API and representation models such as Word2Vec or BERT (developed by Google) exemplify this technological approach.

We now detail the main NLP techniques used in medical chatbots, illustrating how they enable the understanding and processing of human language to improve patient experience and safety.

4.1 Tokenization

Tokenization is the process of breaking down entire sentences into individual words or tokens. This method, known as word tokenization (or sentence tokenization for sentences), is a data processing technique. The extracted tokens are used to build a vocabulary—a collection of unique tokens—which helps the AI system understand the context of a conversation.

4.2 Lemmatization & Stemming

These techniques reduce words to their base or root form. Lemmatization considers the context, while stemming applies rules to remove suffixes. This normalization ensures that variations such as "pain," "pains," or "acute pain" are treated as a single entity, thereby improving the consistency of the analysis.

4.3 Part-of-Speech Tagging

Part-of-Speech (POS) tagging identifies the grammatical role of each word in a sentence (e.g., noun, verb, adjective). For example, in the sentence "I have a headache," the system determines the roles of each word, aiding in understanding that the statement describes a symptom. This step is crucial for comprehending sentence structure and is supported by tools such as Google Cloud.

4.4 Named Entity Recognition (NER)

Named Entity Recognition extracts specific information such as medications, symptoms, or diseases. For instance, in "I take paracetamol for my migraine," the system identifies "paracetamol" as a medication and "migraine" as a symptom. The Google Cloud Natural Language API incorporates this functionality to analyze medical text data.

4.5 Sentiment Analysis

Sentiment analysis detects the emotional state expressed in the text. For a medical chatbot, this allows the system to tailor its responses based on the patient's feelings (e.g., detecting frustration or distress). Google's tools provide metrics to quantify sentiment, enabling more adaptive responses.

4.6 Word Embeddings

Word embeddings transform each word into a numerical vector in a high-dimensional space, capturing semantic relationships between terms. Models like Word2Vec (developed by Google Research) or BERT enable the chatbot to understand that terms such as "pneumonia" and "pulmonary infection" are semantically similar even if not identical.

4.7 Text Classification

Text classification categorizes messages based on their content. For example, a message describing "abdominal pain" can be classified into a specific category to guide the chatbot's response. Google Cloud AutoML Natural Language facilitates the training of classification models tailored to specific medical contexts.

4.8 Machine Translation

In a global medical environment, machine translation is essential for communication between patients and healthcare professionals speaking different languages. Google Translate, which uses advanced NLP techniques, enables a medical chatbot to interact effectively in multiple languages.

4.9 Text Generation

Text generation produces natural, coherent responses based on the context provided by the user. Models such as T5, developed by Google Research, allow the chatbot to generate appropriate responses, ask follow-up questions, or offer suggestions based on the described symptoms.

4.10 Coreference Resolution

Coreference resolution links pronouns or referential expressions to their antecedents in dialogue. For example, in an exchange where a patient says "I have pains" followed by "This is becoming unbearable," the system understands that "this" refers to the "pains." This technique enhances dialogue coherence.

4.11 Topic Modeling

Topic modeling identifies dominant themes within a set of texts. In a medical chatbot, it can detect the primary areas of concern for patients—whether related to symptoms, treatments, or other health issues. Techniques such as Latent Dirichlet Allocation (LDA) can be integrated using frameworks provided by Google.

4.12 Summarization

Automatic summarization condenses lengthy conversations or medical documents to extract the essential information. This technique is particularly useful for generating post-consultation summaries, facilitating the review of key information by the patient. Models like T5 contribute to this summarization capability.

5 Integration of APIs in Medical Chatbots

Medical chatbots are increasingly used to enhance healthcare access, automate patient responses, and facilitate medical information management. These chatbots rely on **APIs (Application Programming Interfaces)** to communicate with external systems (hospitals, laboratories, pharmacies) while ensuring security and compliance with legal standards.

Each API serves a specific function: retrieving data, updating records, ensuring confidentiality, or translating conversations into multiple languages. APIs can be categorized based on their functionalities.

5.1 Communication with External Entities

Medical chatbots use **RESTful APIs** to exchange information with hospital management systems (EHR/EMR), pharmaceutical databases, or appointment booking platforms.

Technical Detail

Representational State Transfer (REST) APIs allow chatbots to **retrieve** and **update** information via HTTP requests (GET, POST, PUT, DELETE). These APIs are lightweight, fast, and compatible with most medical systems.

Examples of commonly used APIs:

- **FHIR (Fast Healthcare Interoperability Resources)**: Standard API to access electronic health records (EHR).
- **Epic API**: Retrieves medical history and manages appointments.
- **Cerner API**: Integrates with hospital systems to view lab results and diagnoses.

Use Cases:

- A patient asks, *Ẅhat medications have been prescribed to me?¨* The chatbot sends a GET request to the **Epic API** or **FHIR API** to retrieve and display prescription data.
- For scheduling, the chatbot uses a POST request to the **Zocdoc API** to record the appointment.

5.2 Security and Compliance

Data security in medical chatbots is critical. Medical APIs use specific protocols to ensure confidentiality and comply with international regulations.

Technical Detail

OAuth 2.0 is a secure authentication protocol allowing chatbots to access medical data without exposing user credentials. It is widely used for EHR system connections.

Example: Microsoft Azure API for FHIR uses OAuth 2.0 to protect access to medical data.

Technical Detail

JWT is a standard to create secure digital tokens for authentication, ensuring only authorized entities interact with the API.

Example: Google Healthcare API uses JWT to manage patient data securely. Medical chatbots must comply with:

- **HIPAA (Health Insurance Portability and Accountability Act)**: U.S. standard ensuring medical data protection.
- **GDPR (General Data Protection Regulation)**: European standard for personal data privacy.

A chatbot using **Google Cloud Healthcare API** must adhere to HIPAA and GDPR regulations when handling patient records.