UAS APPLIED DEEP LEARNING

MedAlpaca Technical Report

Sayyid Abdullah

2206130800

Abstract

MedAlpaca is a collection of open-source medical AI conversation models and training data built on top of the LLaMA (Large Language Model Meta AI) base model. These models are trained on a large dataset of medical text, including research articles, medical records, and discussion forums. MedAlpaca can be used for a variety of tasks, including answering medical questions, generating summaries of medical topics, translating medical text, and writing various types of creative medical content. MedAlpaca is still under development but has shown promising results in a variety of tasks. These models have the potential to be used to improve access to healthcare and improve the quality of healthcare.

1. Introduction

The rapid advancements in artificial intelligence (AI) have significantly impacted various sectors, and healthcare is no exception. One of the most promising areas within healthcare AI is medical AI conversation, and MedAlpaca emerges as a leading force in this domain. MedAlpaca is a collection of open-source medical AI conversation models and training data built upon the LLaMA (Large Language Model Meta AI) base model. These models are trained on a massive dataset of medical text, encompassing research articles, medical records, and discussion forums. This extensive training equips MedAlpaca models with the ability to perform various tasks, including:

- Answering medical questions: MedAlpaca can provide informative and accurate answers to a wide range of medical queries, from simple inquiries about symptoms to complex concerns regarding diagnosis and treatment options.
- Generating summaries of medical topics: Whether it's a specific disease, treatment approach, or medical procedure, MedAlpaca can concisely summarize relevant information, facilitating effective knowledge comprehension.
- Translating medical text: Breaking down language barriers, MedAlpaca can translate medical text from one language to another, promoting accessibility and communication in healthcare settings.
- Writing various types of creative medical content: Beyond its practical applications, MedAlpaca can even generate creative medical content, such as educational stories, poems, or songs, potentially enhancing patient engagement and understanding.

Several factors contribute to the significance of MedAlpaca within the medical AI conversation landscape:

- Open-source nature: Unlike many proprietary AI models, MedAlpaca's open-source code allows for greater transparency, collaboration, and customization. This fosters a vibrant community of researchers and developers who can contribute to the model's ongoing improvement.
- Focus on medical language: MedAlpaca is specifically trained on medical text, equipping it with a deep understanding of medical terminology, concepts, and nuances. This ensures the accuracy and reliability of its responses compared to general-purpose AI models.
- Promising results: Initial evaluations of MedAlpaca have demonstrated its potential for various tasks. For instance, the model has exhibited high accuracy in answering medical questions and generating informative summaries of medical topics.

The versatility of MedAlpaca presents numerous potential applications within the healthcare domain:

- Virtual medical assistants: MedAlpaca models can power virtual medical assistants capable of answering patients' basic medical questions, scheduling appointments, and providing preliminary health information.
- Clinical decision support systems: By analyzing medical data and patient records, MedAlpaca can assist healthcare professionals in making informed clinical decisions, potentially improving patient outcomes.

- Medical education and training: MedAlpaca can generate personalized learning materials and conduct interactive medical simulations, enhancing the learning experience for medical students and practitioners.
- Patient education and communication: MedAlpaca can create tailored educational content and facilitate clear communication between healthcare providers and patients, empowering patients to actively participate in their own care.

Despite its promise, MedAlpaca, like any AI model, faces certain challenges:

- Bias and fairness: Ensuring that MedAlpaca's responses are free from bias and promote equitable healthcare access remains a critical concern. Mitigating biases present in the training data and continuously monitoring the model's outputs are crucial steps.
- Explainability and trust: Building trust in AI models requires transparency regarding their decision-making processes. Developing methods to explain MedAlpaca's reasoning behind its responses will be essential for fostering user trust and acceptance.
- Data privacy and security: Protecting sensitive medical data used to train and operate MedAlpaca is paramount. Implementing robust data security measures and adhering to ethical guidelines are essential for responsible AI development in healthcare.

Looking ahead, the future of MedAlpaca holds immense potential. Continued research and development efforts focused on addressing the aforementioned challenges can further refine the model's capabilities and pave the way for its broader integration into healthcare practices. As MedAlpaca evolves, it has the potential to revolutionize medical care delivery, promoting improved access to information, enhanced patient education, and ultimately, better health outcomes for all.

2. Method

a. Neural architecture model of Alpaca and MedAlpaca

Both models are likely built upon the foundation of transformers, a powerful neural network architecture adept at handling sequential data like text. Transformers leverage attention mechanisms to prioritize relevant portions of the input sequence when making predictions. This allows them to capture intricate relationships between words and sentences, crucial for understanding and generating human language. Imagine a multi-layered structure where each layer processes the input further, refining its understanding. Here's a simplified breakdown of potential components:

- Encoder-Decoder Stack: This core duo forms the backbone of the model.
  - Encoder: Analyzes the input text, extracting meaning and context. Think of it as comprehending what's being said.
  - Decoder: Generates the output text, guided by the encoder's insights. Imagine it as formulating a response or creating new content.
- Attention Layers: These layers within the encoder and decoder employ attention mechanisms. They assess the relationships between words, determining which ones hold the most weight for understanding or generating the desired output.
- Positional Encoding: Since transformers lack inherent understanding of word order, positional encoding helps them differentiate the sequence and context of words within the input.
- Embedding Layer: This layer maps individual words in the vocabulary to numerical vectors, allowing the model to process them mathematically.

While this provides a foundational understanding, additional elements likely contribute to the models' capabilities:

- Residual Connections: These skip connections directly connect layers, facilitating efficient information flow and preventing the vanishing gradient problem that can hinder training.
- Layer Normalization: This technique stabilizes the training process by normalizing the activations within each layer, addressing issues like internal covariate shift.
- Pre-training on Large Datasets: Both models are likely trained on massive amounts of text data, fine-tuned for their specific domains. Alpaca might leverage general text and code, while MedAlpaca would focus on medical literature and resources.

The primary distinction lies in their training data:

- Alpaca: Trained on a broader range of text and code, enabling versatility in tasks like general language processing, code generation, and creative writing.
- MedAlpaca: Focused on medical text, specializing in medical-related tasks like answering medical questions, summarizing medical topics, and translating medical documents.

Another key differentiator is their accessibility:

- Alpaca: Closed-source model, restricting access to its code and inner workings.
- MedAlpaca: Open-source model, allowing researchers and developers to contribute to its improvement and explore its functionalities in detail.

b. Fundamental

Med Alpaca is built from LLaMA (Large Language Model Meta AI) which is the basic model of Med Alpaca. LLaMA represents the latest large language model released by Meta, demonstrating their commitment to open science. Available in various sizes, including 7 billion, 13 billion, 33 billion, and 65 billion parameters.

LLaMA uses the BPE (Byte Pair Encoding) algorithm to carry out tokenization. The whole number is divided into individual digits, and back into bytes to decipher unknown UTF-8 characters. Meanwhile, Med Alpaca uses SentencePiece for tokenization. SentencePiece divides words into several parts called subwords. Sentencepiece is divided into 4 (four) components, namely normalizer, encoder, decoder, and trainer. A new subword segmentation algorithm based on a language model, which this paper calls 'n-best decoding'. This simply refers to when given 'n best segmentations', we choose the best one to maximize a particular score.

The sentencepiece tokenization equation is as follows:

$$P(x|y) \ = \frac{e^{E(x,y)}}{\sum_{c\in V} e^{E(x,c)}}$$

$P(x|y)$: the probability that token x will not follow token y.

$E(x,y)$: a score indicating how likely it is that token x will follow token y.

V : the set of all possible tokens.

Score $E(x,y)$ counted by conditional probability model, i.e:

$$score\ E(x,y) \ = \ log\ P(y|x)/|y|^{\lambda}$$

The higher the score value $E(x,y)$, the more likely it is that token x will follow token y. After carrying out tokenization, continue by representing the tokenized word vector using GloVe. The GloVE model is trained to minimize the following loss function

$$\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}} h(x_{ij})\left(\mathbf{u}_j^\top \mathbf{v}_i + b_i + c_j - \log x_{ij}\right)^2$$

from the equation, it shown that the function is a loss function.

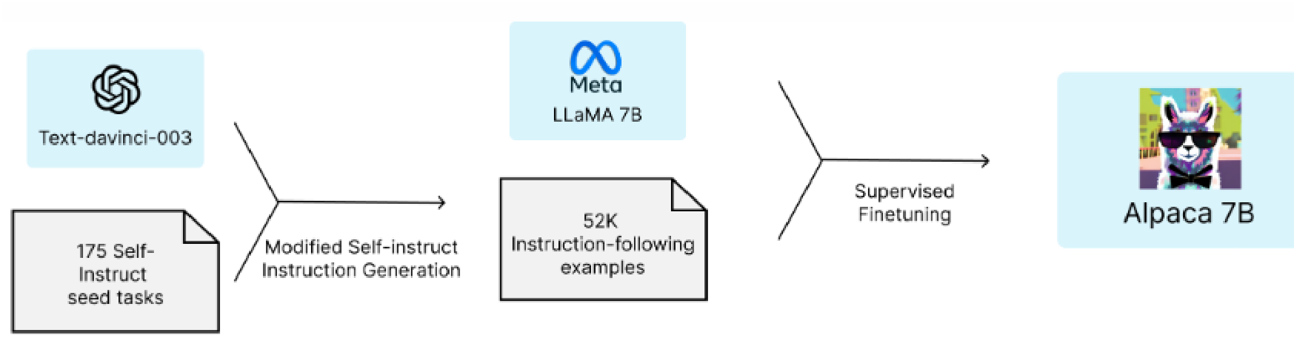
$\boldsymbol{u}_j^T \boldsymbol{v}_i$ are the word vector of x and y, where b_i is center word bias and c_i is the context word bias. Both are the parameters of the model. The equation is the equation for GloVe with sliding window and bias vector. This equation is effective in capturing the semantic relationships between words, which is important for predicting the risk of medical complications.

To facilitate the adaptation of pre-trained language models to particular taks while ensuring efficient weight updates, Low-Rank Adaptation (LoRA) is employed. This technique entails preserving the pre-trained model weights in their original state and integrating trainable rank decomposition matrices into each layer of the Transformer architecture.

3. Practical

There are two important challenges to training a high-quality instruction-following model under an academic budget: a strong pretrained language model and high-quality instruction-following data. The first challenge is addressed with the recent release of Meta's new LLaMA models. For the second challenge, the self-instruct paper suggests using an existing strong language model to automatically generate instruction data. Alpaca is a language model fine-tuned using supervised learning from a LLaMA 7B model on 52K instruction-following demonstrations generated from OpenAI's text-davinci-003.

The figure below illustrates how to obtain the Alpaca model. For the data, generated instruction-following demonstrations by building upon the self-instruct method, and then started with the 175 human-written instruction-output pairs from the self-instruct seed set. Then prompted text-davinci-003 to generate more instructions using the seed set as in-context examples. Then improved over the self-instruct method by simplifying the generation pipeline (see details in GitHub) and significantly reduced the cost. Data generation process results in 52K unique instructions and the corresponding outputs, which cost less than $500 using the OpenAI API.


Text-davinci-003
175 Self-Instruct seed tasks
Modified Self-instruct Instruction Generation
Meta
LLaMA 7B
52K Instruction-following examples
Supervised Finetuning
Alpaca 7B

Medalpaca-7b is a large language model specifically fine-tuned for medical domain tasks. It is based on LLaMA (Large Language Model Meta AI) and contains 7 billion parameters. The primary goal of this model is to improve question-answering and medical dialogue tasks.

Training data

The researchers collected the training data for this project from various resources. First, Anki flashcards were used to automatically generate questions from the front of the cards and answers from the back of the cards. Second, medical question-answer pairs were generated from Wikidoc. Paragraphs with relevant headings were extracted, and Chat-GPT 3.5 was used to generate questions from the headings and use the corresponding paragraphs as answers. This dataset is still under development, and the researchers estimate that approximately 70% of these question-answer pairs are factually correct. Third, StackExchange was used to extract question-answer pairs, selecting the top-rated question from five categories: Academia, Bioinformatics, Biology, Fitness, and Health. Additionally, a dataset from ChatDoctor consisting of 200,000 question-answer pairs was used, available at https://github.com/Kent0n-Li/ChatDoctor.

Model Usage

To evaluate the performance of the model on a specific dataset, you can use the Hugging Face Transformers library's built-in evaluation scripts. Please refer to the evaluation guide for more information. You can use the model for inference tasks like question-answering and medical dialogues using the Hugging Face Transformers library. Here's an example of how to use the model for a question-answering task:

```
from transformers import pipeline
pl = pipeline("text-generation", model="medalpaca/medalpaca-7b",
tokenizer="medalpaca/medalpaca-7b")
```

```
question = "What are the symptoms of diabetes?"
context = "Diabetes is a metabolic disease that causes high blood sugar. The symptoms include increased thirst, frequent urination, and unexplained weight loss."
answer = pl(f"Context: {context}\n\nQuestion: {question}\n\nAnswer: ")
print(answer)
```

Here's a comprehensive explanation of the code:

1. Importing the Necessary Tool:

- from transformers import pipeline: This line imports the pipeline function from the transformers library. This function acts as a convenient way to access and use pre-trained language models for various natural language processing (NLP) tasks.

2. Setting Up a Text Generation Pipeline:

- pl = pipeline ("text-generation", model="medalpaca/medalpaca-7b", tokenizer="medalpaca/medalpaca-7b"): This line creates a text-generation pipeline, specifically designed to generate text based on provided prompts or contexts.
    - It specifies the task as "text-generation".
    - It chooses the "medalpaca/medalpaca-7b" model, a large language model trained for text generation.
    - It uses the matching "medalpaca/medalpaca-7b" tokenizer, which prepares text input for the model.

3. Preparing the Question and Context:

- question = "What are the symptoms of diabetes?": This line stores the question that you want the model to answer.
- context = "Diabetes is a metabolic disease that causes high blood sugar. The symptoms include increased thirst, frequent urination, and unexplained weight loss.": This line provides relevant background information to help the model generate a more accurate and informative answer.

4. Generating the Answer:

- answer = pl(f"Context: {context}\n\nQuestion: {question}\n\nAnswer: "): This line calls the text-generation pipeline to produce an answer.
    - It feeds the context and question into the pipeline in a structured format.
    - The f-string allows for easy integration of variables into the prompt.

- print(answer): This line would display the generated answer on the screen if the code were executed.

4. Reference

Han, T. et al. (2023). MedAlpaca – An Open-Source Collection of Medical Conversational AI Models and Training Data. https://arxiv.org/abs/2304.08247v2.

Frieder, S. et al. (2023). Large Language Models for Mathematicians. https://arxiv.org/abs/2312.04556v1.

Kumar, A. (2023). LLaMA: Concepts Explained (Summary). https://akgeni.medium.com/llama-concepts-explained-summary-a87f0bd61964.

Ji, Z. et al. (2023). Towards Mitigating Hallucination in Large Language Models via Self-Reflection. Association for Computational Linguistics.

Alammar, J. (2020). The Illustrated Transformer. https://jalammar.github.io/illustrated-transformer/.