GPT4MIA: Utilizing Generative Pre-trained Transformer (GPT-3) as A Plug-and-Play Transductive Model for Medical Image Analysis

Yizhe Zhang[1], Danny Z. Chen[2]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China
yizhe.zhang.cs@gmail.com
[2] Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
dchen@nd.edu

Abstract. In this paper, we propose a novel approach (called GPT4MIA) that utilizes Generative Pre-trained Transformer (GPT) as a plug-and-play transductive inference tool for medical image analysis (MIA). We provide theoretical analysis on why a large pre-trained language model such as GPT-3 can be used as a plug-and-play transductive inference model for MIA. At the methodological level, we develop several technical treatments to improve the efficiency and effectiveness of GPT4MIA, including better prompt structure design, sample selection, and prompt ordering of representative samples/features. We present two concrete use cases (with workflow) of GPT4MIA: (1) detecting prediction errors and (2) improving prediction accuracy, working in conjecture with well-established vision-based models for image classification (e.g., ResNet). Experiments validate that our proposed method is effective for these two tasks. We further discuss the opportunities and challenges in utilizing Transformer-based large language models for broader MIA applications.

Keywords: Medical Image Classification · Generative Pre-trained Transformer · GPT-3 · Large Language Models · Transductive Inference

1 Introduction

Modern large language models (LLMs) are built based on the Transformer architecture and are trained to produce a sequence of text output given a sequence of text input such that the output is expected to be semantically **coherent** to the input. For example, for a text completion task, the input text is a sequence of text from a text resource, and the model is trained to produce the next character, word, or sentence of the input text. Open AI's GPT-3 has 175 billion parameters, and was trained on hundreds of billions of words. Brown et al. [4] showed that GPT-3 is capable of few-shot learning: Given a few examples/demonstrations to GPT-3, it can generalize considerably well to new samples with similar characteristics. The input and output coherency and the strong generalization

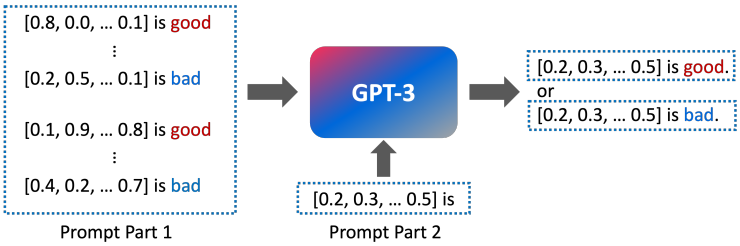

Fig. 1. Illustrating our high-level idea: Using GPT-3 for transductive inference on a binary classification task. Feature texts in Prompt Part 1 are from a set of samples with known labels. Prompt Part 2 contains feature text of a test sample.

capability indicates that pre-trained LLMs such as GPT-3 are potentially capable as general tools for **transductive inference** tasks with limited data.

The notion of transductive inference was first introduced by Vapnik [9]. Given training samples (with labels) and test samples, transductive inference predicts the labels of the test samples using either a parametric model (e.g., a transductive support vector machine (SVM) [7]) or a non-parametric model (e.g., a nearest neighbor based classifier [5]). Different from inductive inference, transductive inference does not aim to induce a prediction function from known samples; instead, its goal is to obtain the labels of test samples via propagating the information from known samples (e.g., training samples).

In this paper, we propose a novel approach, called GPT4MIA, which utilizes GPT-3 as a plug-and-play transductive model to improve medical image analysis (MIA). For an MIA task (e.g., medical image classification), we give information of **known** samples as part of GPT-3's input and ask GPT-3 to infer a new sample's label (see Fig. 1). We expect GPT-3 to infer a test sample's label by using transductive information from the known samples on the test sample. We give theoretical analysis on why this approach is feasible by drawing connections between attention mechanism and nearest neighbor inference. To make this approach more efficient and effective, we optimize the prompt construction, aiming to choose the most representative samples/features and order them in the prompt based on their importance. We present two practical use cases of utilizing our proposed method in medical image classification. We then validate the effectiveness of our method on medical image classification benchmarks.

Our method utilizes a generative pre-trained Transformer for performing transduction from known medical image samples (e.g., training samples) to new test samples. The GPT-3 used in this work has billions of parameters. However, these parameters were pre-trained with language (text) data, and are not being updated during the transduction process for medical image classification. To our best knowledge, this is the first study to utilize a large pre-trained Transformer-based language model for performing transductive inference for image classification tasks (computer vision tasks), which are out of the data domain of the pre-training (language) domain. Our contributions are summarized as follows.

(1) We propose to utilize a large pre-trained language model (e.g., GPT-3) as a plug-and-play transductive inference method for improving MIA. We show that GPT-3 can serve as a general tool for performing transduction with an appropriate setup. Our approach is novel and flexible, suggesting a new direction of research for improving medical AI's accuracy and reliability.

(2) We develop techniques to improve the efficiency, effectiveness, and usability of our proposed GPT4MIA. Two use cases are proposed for GPT4MIA, and strong empirical results validate that GPT4MIA outperforms conventional and state-of-the-art methods in both inductive and transductive method categories.

(3) Our work offers a new way of utilizing a small set of additional labeled data in medical AI: Given a trained deep learning (DL) model and a small set of labeled data (e.g., a validation set), utilizing GPT-3 as a transductive inference method in conjunction with a DL model can achieve better prediction reliability (use case #1) and higher prediction accuracy (use case #2).

2 Approach

In this section, we first provide theoretical analysis on the connection between the attention mechanism and transductive inference mechanism. Then we show details on how to design prompts for using GPT-3 as a transductive inference method. Finally, we present two use cases with workflow to demonstrate how to use GPT-3 as a plug-and-play transductive inference method for MIA.

2.1 Theoretical Analyses

A fundamental component of GPT-3 is the scaled dot-product attention. Typically, three pieces of input are fed to an attention layer: queries Q, keys K, and values V. The scaled dot-product attention can be described as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{s})V, \tag{1}$$

where s is a scaling factor. Below, we show a special case of Eq. (1) can be viewed as a nearest neighbor (NN) classifier under a cosine distance metric system[3].

Setup 1: Suppose the key component K contains features of a set of m known samples, and each feature is of a unit length. The value component V contains these m samples' corresponding labels, and each label is a one-hot vector. The query component Q contains a feature vector (of a unit length), which represents a new test sample whose label is yet to be determined.

Proposition 1: When the scaling factor s is approaching 0 (e.g., s is a very small positive number), the attention function in Eq. (1) is approaching an NN classifier in the cosine distance metric system.

[3] Nearest neighbor classifiers are a typical type of transductive methods for prediction problems.

The above is not difficult to show. QK^T computes the pair-wise similarities between the test sample's feature and the features in the keys K. A small s would enlarge the numerical gap between similar pairs and dissimilar pairs. This then leads to a one-hot-like result after applying the $softmax$ operation. The one-hot-like result is then multiplied with the values V, which chooses the label of a known sample that is most similar to the test sample.

Generative Pre-trained Transformer uses a special type of attention called "self-attention", where the K, V, and Q components are all the same. We will show that in a slightly different setup from Setup 1, the self-attention mechanism can also serve a role as an NN classifier for inferring a new sample's label given known samples' information.

Setup 2: For each known sample, we concatenate its feature vector with the corresponding label vector to form a feature-label vector. We repeat this process for every known sample, and put all the thus-obtained feature-label vectors into K (row by row). In addition, we construct the test sample's feature-label vector by concatenating its feature vector with a label vector containing all zeros. We put this feature-label vector into K as well. Since we are considering the self-attention mechanism, V and Q are constructed in the same way as for K.

Proposition 2: Under Setup 2, self-attention (i.e., Eq. (1) with $K = V = Q$) generates the same label vector as the one that is generated from the attention in Setup 1 for the test sample. With s approaching a small value, self-attention can serve as an NN classifier for inferring the test sample's label.

Since the label vector for the test sample has all zeros at the input, the similarity measures between the test sample and known samples are influenced only by their features. This leads the inference process for the label of the test sample to be essentially the same as shown in Proposition 1. Transformer architecture used in modern large language models, including GPT-3, consists of multiple layers of self-attentions. Below we give more results on stacking self-attentions.

Proposition 3: Under Setup 2, a single layer of self-attention (Eq. (1) with $K = V = Q$) performs one iteration of clustering on feature-label vectors (including those for the known samples and test sample). L layers of self-attention perform L iterations of clustering. There exists a number L^* for the number of layers of self-attention for which the clustering process converges.

Guided by the above theoretical analysis, below we proceed to design the prompt (input) of GPT-3 for transductive inference. We use Setup 2 to guide the prompt construction since GPT-3 uses self-attention: The features and labels of the known samples and the feature of the test sample are put together to feed to GPT-3. According to Proposition 3, stacking self-attentions is functionally more advanced than a nearest neighbor-based classifier. GPT-3 uses not only stacking self-attentions but also numerous pre-trained parameters to augment these attentions. Hence, we expect GPT-3 to be more robust than the conventional methods (e.g., KNN) for transductive inference.

2.2 Prompt Construction

A set of m known samples is provided with their features $F = \{f_1, f_2, \ldots, f_m\}$ and corresponding labels $Y = \{y_1, y_2, \ldots, y_m\}$. A feature vector f_{test} of a test sample is given. The task in this section is to construct a prompt text representation that contains information from F, Y, and f_{test}, which is fed to GPT-3 for inferring the label of the test sample.

Selecting and Ordering Known Samples. As a language model, the original goal of training GPT-3 was to train the model to generate output that is semantically coherent with its input. The data used for training GPT-3 implicitly imposed a prior: The later a text appears in the input prompt (the closer the text to the output text), the larger impact it would impose on the output generation process. Hence, it is essential to put the more representative feature-label texts near the end of the prompt for inferring the test sample's label.

We compute pair-wise similarities between the features in the set F of the known samples and obtain an affinity matrix S, in which each entry $\mathrm{S}_{i,j}$ describes the similarity between samples i and j and is computed as $sim(f_i, f_j)$. A cosine similarity function is the default choice for $sim(.,.)$.

For a feature vector $f_i \in F$, we define a simple measure of how well f_i represents the other known samples: $\mathrm{rep}_i = \sum_{j=1}^{m} \mathrm{S}_{i,j}$. To select the top k representative samples, one can compute rep_i for each $i = 1, 2, \ldots, m$, and choose the largest k representative samples: $\mathbf{index} = argsort(\mathrm{rep}_1, \ldots, \mathrm{rep}_m, \text{“}descend\text{”})$, and **index** is represented as $\mathbf{index}[1, 2, \ldots, k]$. The order of the samples in the prompt for GPT-3 should be in the reverse order of that in the **index** list, where the most representative sample ($f_{index[1]}$) should be put at the end of the prompt in order to give more influence on the output generation. When dealing with imbalanced classification problems, we perform the above process for the samples in each class, and join them in an interleaved fashion.

Converting Features and Labels to Feature-label Texts. For all the feature vectors f_i where i is in the **index** list computed above, we convert these features to texts in an array-like format. For each feature text thus obtained, we put its corresponding label together with the feature text to form a feature-label text. We then put these feature-label texts together into a long text. More details can be found in the Python-like pseudo-code in Listing 1.1 below.

```
def Prompt_Construct_Part1(F,Y,selection_ratio=0.25): #only run once
    ot1=""; m=len(F); k = selection_ratio * m; rep=np.zeros(m,1)
    for i in range(m):
        for j in range(m):
            rep[i]=rep[i]+cosine_sim(f[i],f[j]))
    ind=argsort(rep,"descend"); ind=ind[0:k];
    for i in reversed(range(k)):
        ot1 = ot1+str(f[ind[i]]) + " is in class "
        + str(argmax(y[ind[i]]) + "\n")
    return ot1

def Prompt_Construct_Part2(f_test): #for each test sample
    ot2= str(f_test) + "is in class \n"
    return ot2
```

Listing 1.1. Generating prompts for GPT4MIA.

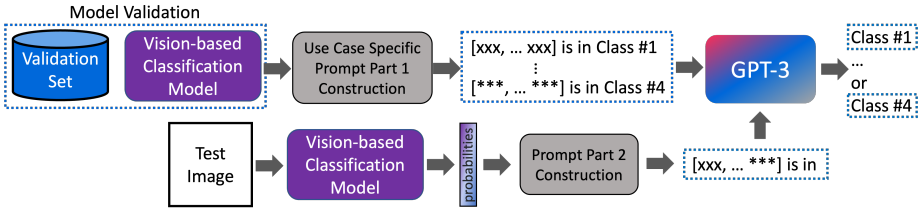

Fig. 2. The workflow of GPT4MIA. A validation set provides references for transductive inference.

2.3 Workflow and Use Cases

In this section, we propose two use cases for improving an already-trained vision-based classification model with our proposed GPT4MIA method. The main workflow is illustrated in Fig. 2.

Use Case #1: Detecting Prediction Errors. The first use case of utilizing GPT-3 as a transductive inference method is for detecting prediction errors by a trained vision-based classifier. Conventionally, a validation set is commonly used for comparing and selecting models. Here, we utilize a validation set to provide known samples for transductive inference. Feature vectors in F are obtained from the output probabilities of the vision-based classification model, and labels in Y are obtained by checking whether the classification model gives the correct prediction on each validation sample.

Use Case #2: Improving Classification Accuracy. The second use case aims to improve an already-trained classifier by directly adjusting its predictions. This is a more challenging scenario in which the method not only seeks to detect wrong predictions but also acts to convert them into correct ones. Feature vectors in F are obtained from the output probabilities of the vision-based classification model, and labels in Y are obtained from the validation set for each validation sample.

3 Experiments

In this section, we empirically validate the effectiveness of our proposed GPT4MIA. Inductive methods (e.g., Linear Regression (LR) [10], Multi-Layer Perception (MLP) [6], and Support Vector Machine (SVM) [2]) and transductive methods (e.g., K-Nearest Neighbor (KNN) [8] and Underbagging KNN (UbKNN) [5]) are applicable to the two use cases presented above. We compare these methods with GPT4MIA in the experiments below.[4]

Configurations: We use the OpenAI API [1] for querying the GPT-3 service for all the experiments related to GPT4MIA. More specifically, the text-Davinci-003 model is used, which can process up to 4000 tokens per request. The hyper-

[4] LR, MLP, SVM, and KNN are conducted using the scikit-learn library at https://scikit-learn.org/, and UbKNN is with our implementation.

parameter k (for top k) is chosen to be a quarter of the number of the total available known samples (m). Inference for one test sample costs about $0.05 USD (charged by OpenAI). For the compared methods, we test their default settings as well as other hyper-parameter settings to report their best results.

3.1 On Detecting Prediction Errors

We utilize the RetinaMNIST and FractureMNIST3D datasets from the MedMNIST dataset [11] for these experiments. We apply a ResNet-50 model trained with the training set as the vision-based classifier, for which the weights can be obtained from the official release.[5] We then collect the model's output probabilities for each validation sample and label it based on whether the prediction is correct. An error detection method is then built based on the information from the validations for classifying the predictions into two classes (being correct or incorrect). The error detection model is then evaluated using the test set working with the same prediction model which was used on the validation (ResNet-50 in this case). We compare our proposed GPT4MIA method on this task with a set of well established inductive methods and transductive methods. From Table 1, one can see that GPT4MIA significantly outperforms the known competing methods for detecting prediction errors from a CNN-based classifier.

3.2 On Improving Classification Accuracy

We utilize the RetinaMNIST and FractureMNIST3D datasets from MedMNIST [11] for these experiments. ResNet-50 is used as the trained vision-based classification model. The model weights are obtained from the MedMNIST official release. In Table 2, we observe that GPT4MIA performs similarly when comparing with the state-of-the-art transductive inference method Underbagging KNN in balanced accuracy. In Tabel 3, we observe that GPT4MIA performs considerably better in balanced accuracy.

3.3 Ablation Studies

We validate the effect of performing sample selection and ordering described in Section 2.2. In Table 4 and Table 5, we show the performances for the setting without the step of sample selection and ordering. From these results, it is clear that sample selection and ordering is important for better performance when utilizing GPT-3 as a transductive inference tool.

Acknowledgement. This work was supported in part by National Natural Science Foundation of China (62201263) and Natural Science Foundation of Jiangsu Province (BK20220949).

[5] The model weights are obtained from https://github.com/MedMNIST/experiments.

Table 1. Experiments for Use Case #1: Detecting Prediction Errors.

Method	RetinaMNIST				FractureMNIST3D			
	Precision	Recall	F-score	Bal-Accu	Precision	Recall	F-score	Bal-Accu
LR	**0.617**	0.631	0.624	0.486	0.492	**0.776**	0.604	0.517
MLP	0.616	0.637	0.626	0.488	0.571	0.482	0.523	0.572
SVM	0.607	0.648	0.627	0.489	0.527	0.414	0.464	0.533
KNN	0.608	0.407	0.488	0.458	0.488	0.716	0.580	0.507
UbKNN	0.574	0.859	0.689	0.551	0.673	0.673	0.673	0.564
GPT4MIA	0.581	**0.860**	**0.693**	**0.679**	**0.706**	0.673	**0.689**	**0.603**

Table 2. Experiments for Use Case #2. Dataset: RetinaMNIST.

Method	Class #1	Class #2	Class #3	Class #4	Class #5	Bal-Accu
N/A	0.813	0.063	0.400	0.563	0.0	0.368
LR	0.753	0.0	0.663	0.29	0.0	0.342
MLP	0.736	0.0	0.456	0.50	0.0	0.338
SVM	0.729	0.0	0.369	0.75	0.0	0.370
UbKNN	0.747	0.130	0.478	0.603	0.0	0.392
GPT4MIA	0.672	0.437	0.207	0.529	0.150	**0.399**

Table 3. Experiments for Use Case #2. Dataset: FractureMNIST3D.

Method	Class #1	Class #2	Class #3	Bal-Accu
N/A	0.778	0.375	0.326	0.493
LR	0.600	0.596	0.304	0.500
MLP	0.556	0.673	0.283	0.504
SVM	0.533	0.596	0.391	0.507
UbKNN	0.644	0.394	0.478	0.505
GPT4MIA	0.522	0.510	0.543	**0.525**

Table 4. Ablation study for Use Case #2. Dataset: RetinaMNIST.

Method	Class #1	Class #2	Class #3	Class #4	Class #5	Bal-Accu
w/o Selection & Ordering	0.724	0.086	0.435	0.456	0.1	0.360
GPT4MIA (Full)	0.672	0.437	0.207	0.529	0.150	**0.399**

Table 5. Ablation study for Use Case #2. Dataset: FractureMNIST3D.

Method	Class #1	Class #2	Class #3	Bal-Accu
w/o Selection & Ordering	0.311	0.769	0.283	0.454
GPT4MIA (Full)	0.522	0.510	0.543	**0.525**

4 Discussion and Conclusions

In this paper, we developed a novel method called GPT4MIA that utilizes a pre-trained large language model (e.g., GPT-3) for transductive inference for medical image classification. Our theoretical analysis and technical developments are well-founded, and empirical results demonstrated that our proposed GPT4MIA is practical and effective. Large language models (LLMs) such as GPT-3 and, recently, ChatGPT [3] have shown great capability and potential in many different

AI applications. In this work, we showed that GPT-3 can perform transductive inference for medical image classification with better accuracy than conventional and state-of-the-art machine learning methods. LLMs are great new technologies that can push the boundaries of AI research; on the other hand, new concerns are raised in using these generative models. Reliability and privacy are among the top priorities for medical image analysis, and more efforts should be put into this frontier when working with LLMs. In addition, further improving LLMs for medical image analysis, including better robustness and accuracy, lower costs, and more use cases, are all exciting and important future research targets.

References

1. OpenAI API. https://openai.com/api/.
2. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.
3. ChatGPT. https://openai.com/blog/chatgpt/.
4. Tom Brown et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.
5. Hanyuan Hang, Yuchao Cai, Hanfang Yang, and Zhouchen Lin. Under-bagging nearest neighbors for imbalanced classification. *Journal of Machine Learning Research*, 23(118):1–63, 2022.
6. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.
7. Thorsten Joachims et al. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.
8. Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.
9. Vladimir Vapnik. *Statistical Learning Theory*. Wiley, 1998.
10. Sanford Weisberg. *Applied Linear Regression*, volume 528. John Wiley & Sons, 2005.
11. Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2 – a large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, 2023.