

# FINE-TUNING CHATGPT: ADVISING ON DIGITAL TRANSFORMATION FOR SMALL AND MEDIUM-SIZED ENTERPRISES

Phuong-Linh Ngo-Tran<sup>1\*</sup>, Thanh-Nghi Doan<sup>2</sup>

**Abstract** – *This paper examines how to enhance ChatGPT through fine-tuning techniques for consulting purposes, with a focus on supporting small and medium-sized enterprises in their digital transformation efforts. ChatGPT can be tailored to provide more focused direction and support in negotiating the complexities of digital transformation procedures by utilizing fine-tuning strategies. The study explores the potential benefits of utilizing ChatGPT in consulting contexts, highlighting its capacity to deliver customized solutions and valuable insights for companies seeking to grow and adapt in an increasingly digital environment. The research results revealed notable efficacy and performance of the chatbot, including increased user satisfaction, higher rates of successful digital transformation activities among small and medium-sized enterprises, and more accuracy in offering pertinent advice.*

**Keywords:** *ChatGPT, conversational AI, digital transformation, Transformer model.*

## I. INTRODUCTION

Small and medium-sized enterprises (SMEs) are facing numerous challenges on their digital transformation journey. In this context, leveraging technology to optimize operations and deliver better services has become a pressing priority. Among modern technologies, chatbots are emerging as vital tools, providing convenience and efficiency in customer consultation and support.

---

<sup>1</sup>University of Information Technology, VNUHCM, Vietnam

<sup>2</sup>An Giang University, VNUHCM, Vietnam

\*Corresponding author: ntplinh1909@gmail.com

Received date: 3 July 2024; Revised date: 5 November 2024; Accepted date: 3 December 2024

A chatbot is not merely an automated chat tool but also a bridge between businesses and customers, enhancing interaction and improving the user experience. As the market gradually shifts from traditional channels to online platforms, having an intelligent chatbot can be a stepping stone for businesses to attract and retain customers.

The process of building a consultancy and support chatbot for SMEs is not simply deploying an automated application. It requires a deep understanding of customer needs and business operations. Developing a chatbot needs to be tailored to specific requirements, from providing basic information to handling complex customer requests. This paper contributes to building a dataset for digital transformation consultancy tailored to SMEs and developing an artificial intelligence (AI) chatbot system based on ChatGPT (GPT-3.5) to provide digital transformation consultancy services to SMEs.

## II. LITERATURE REVIEW

Technology has shaped significant developments in all aspects of business and the digital transformation process. A persistent challenge has been the technical deficiencies, operational capacity, and market access, which are significant hurdles for SMEs. However, a positive sign has emerged with the robust development of chatbot technology and AI, addressing these challenges quickly and effectively. Building an AI chatbot to support SMEs in digital transformation will continue to be a significant contribution to solving lingering issues in today's business.

In Vietnam, interest in researching conversational AI is growing significantly. Initially, simple chatbot systems became more complex

through the integration of artificial intelligence and natural language processing, allowing them to understand and respond to more complex queries. However, research on conversational AI in Vietnam is currently focused on optimizing chatbots through machine learning and artificial intelligence methods. For example, Nguyen et al. [1] developed a chatbot to support programming learning. Meanwhile, Mai et al.'s study [2] focused on improving the chatbot's performance by using custom pipelines. Vi Manh Tuyen et al. [3] have suggested using chatbots in the healthcare sector with high accuracy. This is a significant advancement in the application of conversational AI in various fields while also promoting the development of small and medium-sized businesses in Vietnam.

Foreign studies on the application of chatbots have gained significant attention from businesses and the technology research community [4-6]. Misichia et al.'s study [4] introduce the development of chatbots in customer service and analyze their impact on businesses. Kumar et al. [5] provided an objective analysis of previous chatbot research and proposed a method for developing an advanced chatbot application that can be easily customized to meet customer needs. Ray's study [6] presents a comprehensive review of the ChatGPT platform, highlighting its potential applications in various fields, such as healthcare, consultancy, education, and commerce. Additionally, these studies emphasize the importance of ensuring information security and privacy when utilizing chatbot systems.

This paper contributes to developing a dataset for digital transformation consultancy tailored to SMEs and an AI-powered chatbot system based on ChatGPT (GPT-3.5) to assist small and medium-sized enterprises (SMEs) in their digital transformation efforts.

### III. RESEARCH METHODS

#### A. ChatGPT model

ChatGPT is an AI chatbot that uses natural language processing to generate human-like conversations, developed by OpenAI. ChatGPT

can answer questions and compose various text-based content, including articles, social media posts, essays, and others. It integrates multiple machines learning techniques, including deep learning, unsupervised learning, fine-tuning, contextual learning, and reinforcement learning. It is built upon the original Generative Pre-training Transformer (GPT) model and has been iteratively updated from GPT-1 to GPT-4.

#### B. ChatGPT training methodology

OpenAI trained this model using reinforcement learning from human feedback (RLHF), which is similar to InstructGPT but with certain differences in data collection. Figure 1 illustrates the training process of the ChatGPT model.

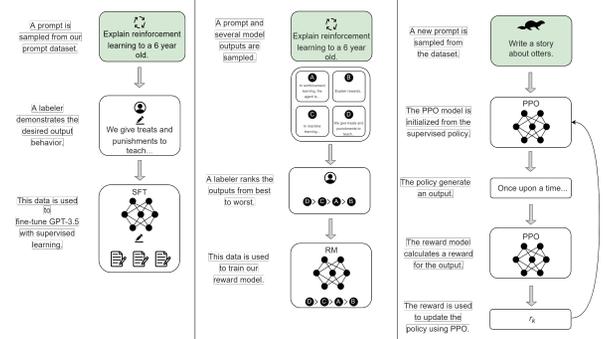


Fig. 1: The process diagram of ChatGPT training

#### C. Transformer

Transformers is a groundbreaking model architecture introduced in 2017 by the Google AI research team [7]. Simply put, the Transformer architecture is a type of neural network used to process natural language data. It simulates how the human brain processes information by handling information through interconnected layers. Figure 2 depicts the architecture of a Transformer.

The most prominent feature of the Transformer is its use of the attention mechanism. Instead of processing the entire input sequence simultaneously, the Transformer focuses on the most

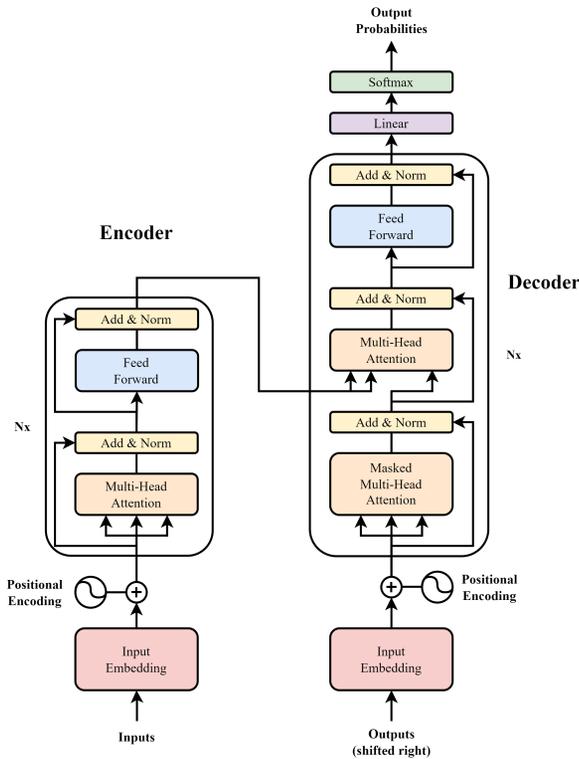


Fig. 2: The Transformer architecture

relevant input parts for each task. This approach enables the model to learn relationships between words in a sentence, thereby improving its accuracy in tasks such as machine translation and text summarization.

Another key component of the Transformer is its encoder-decoder architecture. The encoder is responsible for converting the input sequence into a vector representation, while the decoder uses this representation to generate the desired output sequence. This structure allows the model to effectively capture relationships between different input and output parts, enhancing its performance on various language tasks.

The Transformer employs relative positional encoding to further refine its understanding of language structure. This mechanism helps the model retain information about the position of words in a sentence, allowing it to recognize syntactic relationships more accurately. As a result, tasks such as parsing and named entity recogni-

tion benefit significantly from this approach.

Additionally, the Transformer is designed with a strong parallelization capability, making it highly efficient on hardware architectures such as Graphics Processing Unit (GPU). This enables faster processing speeds, particularly when handling large datasets.

Thanks to these characteristics, the Transformer has quickly become one of the most popular neural network architectures for natural language processing tasks, revolutionizing the field with its efficiency and accuracy.

#### D. Transfer learning

Transfer learning is a machine learning technique where a model trained for one task is leveraged as a starting point for a model for another task [8]. The idea is to use the knowledge gained from solving one problem and apply it to a related problem, aiming to speed up the solution and reduce the amount of training data needed.

OpenAI developers have employed transfer learning methods to learn from the GPT-3 language model, which was trained on a large corpus of text; it has been further trained using human comparison datasets between different responses to better predict what they deem as ‘a reasonable response.’

#### E. Fine-tuning

Fine-tuning, or model fine-tuning, is an important process in machine learning that improves the performance of a model and ensures it performs well on new data. When fine-tuning a model, the model’s hyperparameters or use of different optimization methods are often adjusted to improve performance.

Fine-tuning involves adjusting hyperparameters, such as learning rate, number of hidden layers, batch size, and number of epochs during training. Additionally, techniques like cross-validation can be applied to evaluate the model’s performance on test data. This process is flexible and can be carried out in various ways depending on the specific problem and model used. A deep understanding of the model and data, and strong

analytical skills, are essential for analyzing and evaluating results.

However, fine-tuning does not always lead to improved performance. Excessive fine-tuning can lead to overfitting, a condition in which the model performs very well on the training data but does not on new data.

In the case of the ChatGPT model, the pre-trained model is GPT-3. GPT-3 is a large language model (LLM) trained on a massive dataset consisting of text and code. It can generate text, translate languages, write various types of creative content, and provide comprehensive answers to questions. To fine-tune ChatGPT for a specific task, a smaller dataset is collected for that task. This dataset may include examples of conversations, questions, answers, or text passages. The ChatGPT model is then trained on this dataset, using the weights of GPT-3 as the starting point. The fine-tuning process helps the ChatGPT model learn the specific characteristics of the task it is designed to perform. This can significantly improve the model's performance in each specific task.

#### F. Large language model

LLMs are machine learning models that use deep learning algorithms to process and understand natural language. These models are trained in large amounts of text data to learn patterns and entity relationships in language. LLMs can perform various language tasks, such as language translation, sentiment analysis, chatbot conversations, etc. They can understand complex text data, identify entities and their relationships, and generate new text that is fluent and grammatically accurate.

The term 'large language model' typically refers to models that use deep learning techniques and have a large number of parameters, ranging from millions to billions. These models can capture complex language patterns and generate text that is often indistinguishable from human-written text. Several examples of LLMs are presented as follows.

- GPT-3 (Generative Pre-trained Transformer 3) – This is one of the most significant language models developed by OpenAI. It has 175 billion parameters and can perform various tasks, including text generation, translation, and summarization.

- BERT (Bidirectional Encoder Representations from Transformers) – Developed by Google, BERT is another popular LLM trained on a massive text corpus. It can understand the context of a sentence and generate meaningful responses to questions.

- XLNet – This LLM, developed by Carnegie Mellon University and Google, uses a novel approach to language modeling called 'Permutation language modeling.' It has achieved state-of-the-art performance on language tasks, including language generation and question answering.

- T5 (Text-to-Text Transfer Transformer) – T5, developed by Google, is trained on various language tasks and can perform text-to-text transformations, such as translating text into another language, generating summaries, and answering questions.

- RoBERTa (Robustly Optimized BERT Pre-training Approach) – Developed by Facebook AI Research, RoBERTa is an improved version of BERT that performs better on some language tasks.

#### G. An example of deploying a chatbot

An example of a chatbot deployed to search for product information in the database is shown in Figures 3 and 4. Developing a machine learning model follows a structured process to ensure accuracy and efficiency. It begins with data collection, where relevant and high-quality data is gathered from various sources. The dataset is then used to enhance its usability through cleaning, transformation, and feature engineering. Once the data is ready, a suitable model is built using frameworks, with careful selection of architecture and hyperparameters. The model undergoes fine-tuning to optimize performance, often leveraging transfer learning and regularization techniques to prevent overfitting. The evaluation uses val-

idation datasets and key performance metrics to measure accuracy and effectiveness. Finally, the trained model is deployed into a real-world environment, integrated into applications, and continuously monitored for improvements based on user feedback and new data. This systematic approach ensures the model is robust and adaptable to practical use cases.

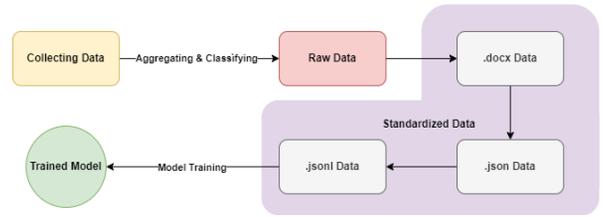


Fig. 5: Data processing steps

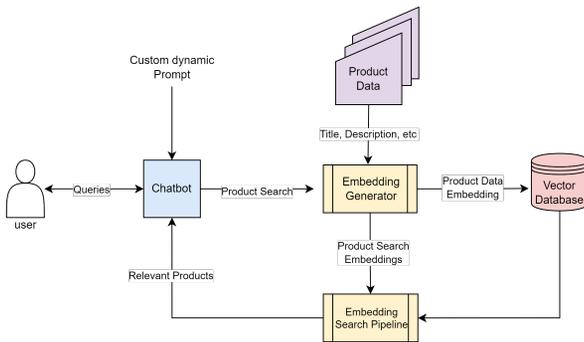


Fig. 3: The operations of a chatbot

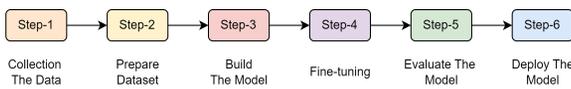


Fig. 4: The model’s operational flow

*H. Building the fine-tuning data for the ChatGPT chatbot*

**Data specification**

The data collection process is always a crucial step in determining the success of every chatbot, and this digital transformation support chatbot project is no exception. Figure 5 illustrates the data processing process. To have an effective and highly accurate chatbot, the collected question dataset must be significant, diverse, and low in noise. The collection results are 1,159 messages, equivalent to 2,318 questions, belonging to both user and assistant roles.

**Data collection**

One of the preferred methods is to gather data from reputable sources such as textbooks, articles, conferences, and documents related to

businesses, as well as general definitions from vetted websites. Experts also serve as reliable sources for data collection.

First, the study gathered data from government documents and digital transformation handbooks [9–14]. Next, to ensure the collected data is of high quality and reliability, accurately reflecting the realities, the study collaborated with the Department of Information and Communications (officials managing digital transformation-related information in the agricultural and commercial service sectors in An Giang Province, Vietnam), along with experts in the field of digital transformation. Information about digital transformation in these sectors was provided and gathered together. The data collection process included developing a set of main and several sets of auxiliary questions (consisting of related or frequently asked questions), each accompanied by corresponding answers. In addition, data was also collected from the internet, specifically from reputable websites curated by non-profit organizations. This further diversifies the chatbot’s data sources, making them more realistic and ensuring that the data is always current.

The collected data needs to be diverse and sufficient, focusing on digital transformation. However, it is essential to minimize duplicate questions as much as possible. For example, if there is already a message with the question ‘What is digital transformation?’, this question should be avoided in any other message. Otherwise, it may lead to the undesired situation of overfitting, but this does not apply to answers.

### Building JSON – JSONL

#### - Building messages

In ChatGPT, messages contain sub-objects, including roles and content, representing each response in the conversation (see Table 1). A message can have multiple roles and contents inside, depending on the model’s purpose. Messages also represent the training dataset; therefore, the more messages there are, the more the model can learn. Many messages may be in a JSON - JSONL file, as shown in Figure 6.

Table 1: How JSON files store messages

```
{
  "messages":
  [{
    "role": "this role",
    "content": "this content"
  }]
},
{
  "messages":
  [{
    "role": "this role",
    "content": "this content"
  }],
  {
    "role": "this role",
    "content": "this content"
  }
}]
}
```

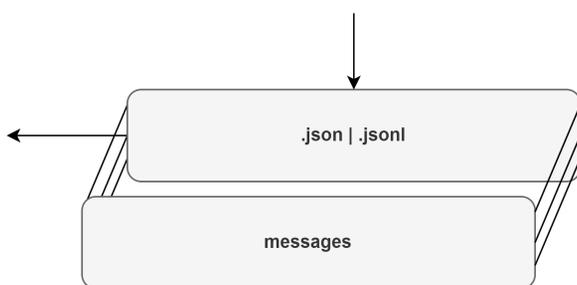


Fig. 6: Messages inside JSON - JSONL

assistant. Each role has different responsibilities in different conversations. Similarly, content is also a sub-attribute of messages. Content contains the main content of the conversation, depending on the user’s intention. Content can contain a question, an answer, or a notification (Table 2).

Roles and contents are the two main components that form a message, with each role having only one content and vice versa, as depicted in Figure 7.

Table 2: Deploy a message in JSON

```
{
  "messages":
  [{
    "role": "system",
    "content": "I am a chatbot"
  }],
  {
    "role": "user",
    "content": "Discussing digital transformation"
  }],
  {
    "role": "assistant",
    "content": "Digital transformation is the comprehensive and holistic process of individuals and organizations changing their lifestyles, work methods, and production techniques based on digital technologies."
  }
}]
}
```

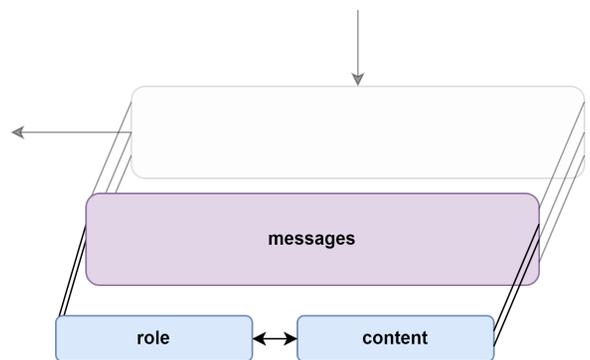


Fig. 7: The positions of roles and content within messages

#### - Building roles and contents

Roles are attributes of messages, defining the role of the speaker in the conversation. Roles are divided into three main values: system, user, and

**- Building JSON - JSONL**

Each JSONL line is a simple JSON and is separated by a newline character. Inside the JSONL, there are messages as described in Table 3. Figure 8 provides a detailed description of JSON and JSONL.

Table 3: Deploy messages to a JSONL file

```
{
  "messages": [
    {
      "role": "system",
      "content": " I am a chatbot AI ."
    },
    {
      "role": "user",
      "content": " What directions do small and medium-sized enterprises take in their digital transformation?"
    },
    {
      "role": "assistant",
      "content": " Small and medium-sized enterprises (SMEs) are undergoing digital transformation in various directions, including: Applying technology to management activities, Enhancing customer experience, Optimizing production and business processes, Developing human resources, Collaborating with other businesses."
    }
  ]
}

{
  "messages": [
    {
      "role": "system",
      "content": "I am a chatbot AI."
    },
    {
      "role": "user",
      "content": "What is the application of technology in management activities."
    },
    {
      "role": "assistant",
      "content": " Applying technology to management activities involves using sales management software (CRM), inventory management (ERP), human resource management (HRM) to automate processes and save costs. Implementing electronic invoices, digital signatures minimizes paperwork, ensures information security. Deploying surveillance camera systems, access control ensures security."
    }
  ]
}
```

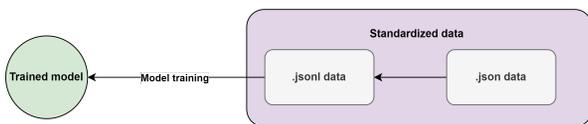


Fig. 8: JSON – JSONL Description

**Raw data processing**

After the data collection process, either equivalent question-answer pairs or text samples may be obtained in formats such as .txt, .docx, .xlsx, etc. However, these formats are not suitable for fine-tuning as they are not in a format that the model can learn from. Therefore, it is necessary to convert these data samples into a standardized format. Following a general convention, examples from JSONL format, as exemplified by OpenAI, are in Table 4.

With the construction of messages, roles, and content outlined above, each user’s content will have corresponding assistant content. Therefore,

Table 4: JSONL examples from OpenAI

```
{
  "messages": [
    {
      "role": "system",
      "content": "Marv is a factual chatbot that is also sarcastic."
    },
    {
      "role": "user",
      "content": "What's the capital of France?"
    },
    {
      "role": "assistant",
      "content": "Paris, as if everyone doesn't know that already."
    }
  ]
}

{
  "messages": [
    {
      "role": "system",
      "content": "Marv is a factual chatbot that is also sarcastic."
    },
    {
      "role": "user",
      "content": "Who wrote 'Romeo and Juliet'?"
    },
    {
      "role": "assistant",
      "content": "Oh, just some guy named William Shakespeare. Ever heard of him?"
    }
  ]
}

{
  "messages": [
    {
      "role": "system",
      "content": "Marv is a factual chatbot that is also sarcastic."
    },
    {
      "role": "user",
      "content": "How far is the Moon from Earth?"
    },
    {
      "role": "assistant",
      "content": "Around 384,400 kilometers. Give or take a few, like that really matters."
    }
  ]
}
```

to facilitate data construction, the data will be categorized into three types: in .jsonl format, in processed .docx format, and in other formats. Figure 9 provides a detailed description of the data processing process in a standard format.

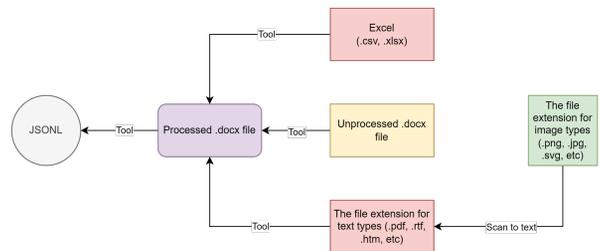


Fig. 9: The process of standardizing data

The .jsonl format will follow the specific example provided by OpenAI.

For the processed .docx format:

- The processed .docx format will be designed according to the standard question-answer equivalence.

- A raw .docx file lacks a specific structure, making it impossible to process unless it is converted into a question-answer format similar to the structured .docx section above.

- From this processed .docx file, a custom-written tool will be used to convert it into a standardized .jsonl format as described by OpenAI.

- With the Python language, the implementation idea includes the following steps:

- + Step 1: Use the ‘charset’ library to check the

Table 5: Example of a processed .docx file

What is the relationship between the Fourth Industrial Revolution, digital transformation, and AI?  
 The Fourth Industrial Revolution (4.0) facilitates the development of digital transformation, in which artificial intelligence (AI) plays a crucial role. Digital transformation enables the harnessing of data and the automation of production processes, while AI provides the capability to learn and analyze information from this data to optimize performance and make intelligent decisions.

What are the three levels of digital transformation?  
 Digital Integration: Transforming traditional processes into digital form to enhance efficiency and flexibility. (2) Digital Optimization: Utilizing data and digital technologies to optimize existing processes and services. (3) Digital Transformation: Comprehensive changes in business approach and the creation of new business models through extensive development of digital technology and innovation.

encoding of the input .docx file to ensure it can be read and processed accurately.

+ Step 2: Utilize the ‘docx’ library to extract the content of the .docx file and divide it into separate paragraphs representing a part of the assumed conversation.

+ Step 3: Iterate through each paragraph in the .docx file, and encode the assumed messages for a conversation, with each message including information about the sender’s role (user, assistant, or system) and the content of the message.

+ Step 4: Once all messages created, format them into JSON objects and write them to a JSON file, with each line of the file representing a complete conversation between the user and the assistant.

Finally, the text file extensions can appear differently.

- These could be .pdf, .xlsx, .csv, etc.

- Due to the various ways text data can be represented, each file extension may require a different tool for conversion into the standard .docx format before being transformed into .jsonl format.

- To illustrate the point, a tool will be developed to process a typical Excel file. For example,

if the Excel file contains seven columns: No., Enterprise ID, Enterprise Name, Headquarters, Status, Type of Enterprise, and Main Business Sector—each column (except for the No. field) will be sequentially placed into corresponding fields. This results in Enterprise Name with the ID Enterprise ID, located at Headquarters, belonging to Type of Enterprise, with the status Status, and operating in the main business sector of Main Business Sector. The following steps will be performed sequentially:

- Step 1: Use the ‘openpyxl’ library to open the Excel file and access the current worksheet.

- Step 2: Identify the data columns by comparing column names with the list of required columns, including fields as described above.

- Step 3: Before each assistant’s content, there is a line representing user content (a question, or it could also be the topic of a response content within the assistant).

- Step 4: For each row in the Excel file, extract enterprise information from the identified columns and create a suitable text paragraph based on that enterprise information.

- Step 5: Use the ‘docx’ library to create a new Word document and add the extracted enterprise information to this document.

- Step 6: Reuse the tool above to convert the .docx file to .jsonl.

Note:

- For each different data file, there will be different ways of representing the data, so the above idea is only relative and cannot be applied to all Excel files.

- The content of the .docx can be changed according to preferences, as long as it adheres to the standard .docx format mentioned earlier, to use the tool to convert .docx to .jsonl as described in Table 5.

- Additionally, specific special file extensions such as .png, .jpg, .svg, and .gif do not yet have an established method for training or conversion into a standard format. Currently, a manual approach is employed, where each image is scanned using a camera to extract text, which is then converted into a standard .docx format before being transformed into .jsonl.

### Data segmentation

The process of research and experimentation has encountered a significant challenge in collecting a sufficiently large amount of data to train the model. Currently, 1,159 messages on various digital transformation and business topics from multiple sources have been gathered.

There are various data splitting methods, such as 70:15:15 or 60:20:20. However, based on several studies and expert opinions, the study has decided to split the dataset using the 80:10:10 ratio. This approach balances the training and evaluation needs by dividing the data into 80% for the training file, 10% for the validation file, and 10% for the testing file.

Therefore, to maximize productivity from the available data, the plan is to split it according to the 80:10:10 ratio as shown in Figure 10. This means 80% of the data will be used for model training, 10% for evaluating performance and tuning parameters, and the remaining 10% for model testing before deployment. With this ratio, the data files will be distributed as follows:

- Training dataset: 80% of the total data, approximately 927 messages.
- Validation dataset: 10% of the total data, approximately 116 messages.
- Test dataset: 10% of the total data, approximately 116 messages.

With this split, the training dataset will be large enough for the model to learn all patterns and trends in the data adequately. The dataset will be sufficient to control and fine-tune the model during training without overly impacting the training data. Additionally, with 10% of the data allocated for testing, a completely independent dataset is available to evaluate the model's performance after training. This supports assessing the model's generalization ability on new data.

## IV. RESULTS AND DISCUSSION

### A. Experimental environment

Fine-tuning for the GPT 3.5 Turbo 1106 model will only be performed on one of the three popular operating systems: Windows, MacOS, or Linux. Anaconda platform may be utilized

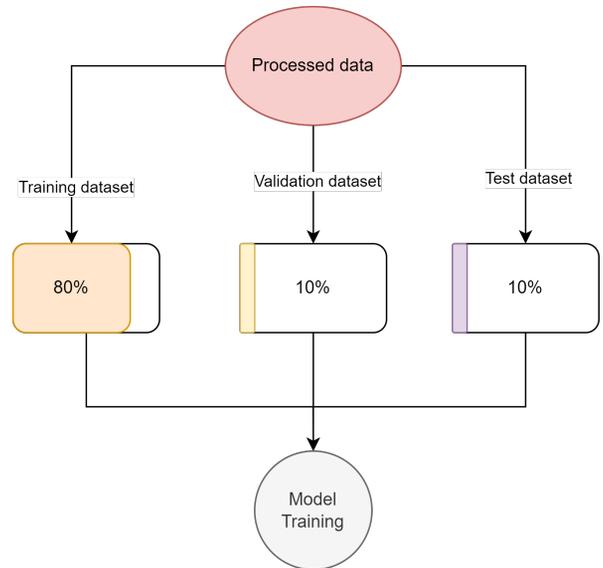


Fig. 10: The ratio of data file split

by users who want to manage Key API and perform fine-tuning indirectly. Besides, Python will be used as the programming language, with a minimum version of Python 3.6. However, for optimal performance and compatibility, Python version 3.8 or higher is recommended. This article utilizes Python version 3.11. The OpenAI library is the primary library for interacting with OpenAI's API. It allows for various tasks such as file management, creating and managing fine-tuning tasks, and accessing the status of those tasks. The model to be fine-tuned is GPT-3.5 Turbo 1106, an upgraded version of GPT-3.5 Turbo released in October 2023 with significant improvements in performance and text generation capabilities compared to the previous version. Although, there are no specific device requirements from the developer. A personal device with the following specifications will be used: CPU: 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz; RAM: 8GB RAM; Operating System: Windows 11.

### B. Training results

#### Performing training

Step 1: Access the OpenAI platform. The focus

will be on four main sections in the toolbar: ‘Playground’ for model testing, ‘Fine-tuning’ for training, monitoring the training process, and storing trained models, ‘Storage’ for managing data files, and ‘API keys’ for generating and storing API keys, as shown in Figure 11.

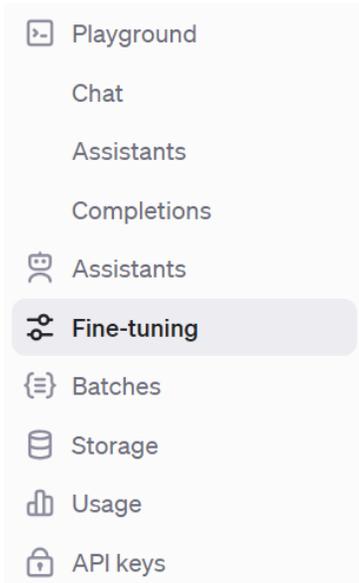


Fig. 11: The toolbar of the OpenAI platform

Step 2: Access the ‘Fine-tuning’ section, then select ‘Create’. A dialog box like Figure 12 will appear.

Step 3: Select the model for training by choosing the previously proposed model, GPT-Turbo 1106. Next, choose the training and validation files built-in section F. Then, enter the remaining values such as ‘Suffix’, ‘Seed’, ‘Batch size’, ‘Learning rate multiplier’, and ‘Number of epochs”, or leave them blank to allow the system to select appropriate parameters automatically. Finally, click ‘Create’ to initiate the training process.

**Model parameters after training**

In the ‘Fine-tuning’ section, the newly trained model can be viewed by clicking to see the details in Figure 13.

The components above describe the process of handling the initial data files for the entire model detailed in Table 6.

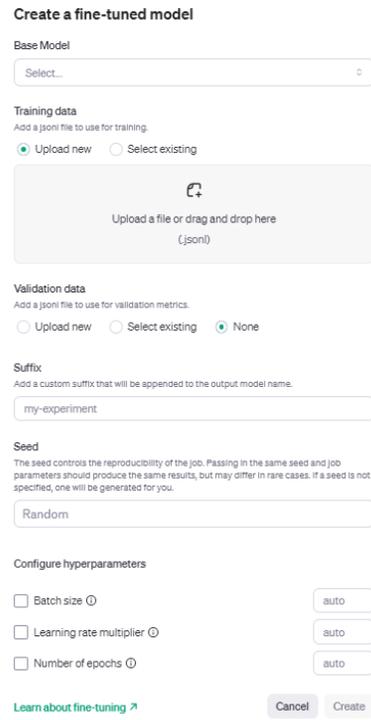


Fig. 12: Create a fine-tuned model dialog box

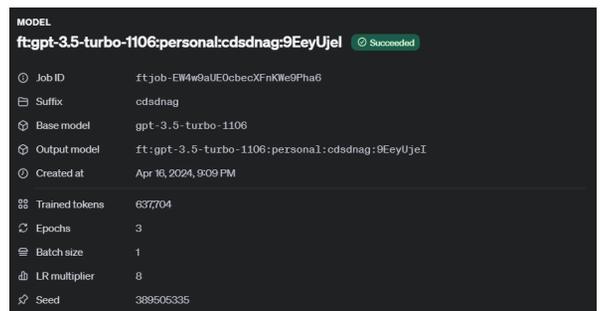


Fig. 13: Training result

Table 6: Training result

Model information	Describe
Job ID	ftjob-EW4w9aUE0cbecXFnKWz9Pha6
Suffix	cdsdnag
Base model	gpt-3.5-turbo-1106
Output model	ft:gpt-3.5-turbo-1106:personal:cdsdnag:9EeyUjeI
Created at	Apr 16, 2024, 9:09 PM
Trained tokens	637,704
Epochs	3
Batch size	1
LR multiplier	8
Seed	389505335

Job ID: 'ftjob-EW4w9aUE0cbecXFnKWe9Pha6': Unique identifier code for the fine-tuning task.

Suffix: 'cdsdnag': Unique identifier code for the model.

Base model: 'gpt-3.5-turbo-1106': The name of the GPT-3.5-turbo-1106 model used as the base model for fine-tuning.

Output model: 'ft:gpt-3.5-turbo-1106:personal:cdsdnag:9EeyUjeI': The name of the fine-tuned model generated after completing the fine-tuning task.

Created at: 'Apr 16, 2024, 9:09 PM': The date when the fine-tuning process resulted in the model.

Trained tokens: '637,704': The number of tokens used for fine-tuning the model.

Epochs: '3': The number of times the entire training dataset was iterated through.

Batch size: '1': The size of a small dataset used to update model parameters in each iteration of the training process.

LR multiplier: '8': The multiplier factor for the initial learning rate to adjust the speed of updating model parameters.

Seed: '389505335': The random seed used to initialize the model.



Fig. 14: The checkpoint

Figure 14 presents a list of model checkpoints during the fine-tuning process. Each checkpoint represents a specific stage in the training process, with information such as the checkpoint name, checkpoint status (training, completed, error), and number of training steps.

As shown in Figure 14, the checkpoints are steps 417, 834, and 1,251, indicating that there are three checkpoints throughout the fine-tuning process.

In Figure 15, there is 'validationset.jsonl', the

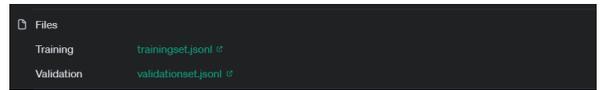


Fig. 15: File data

dataset used to evaluate the model's performance during training.

Finally, there is the training chart in Figure 16, which displays the loss over each training step.

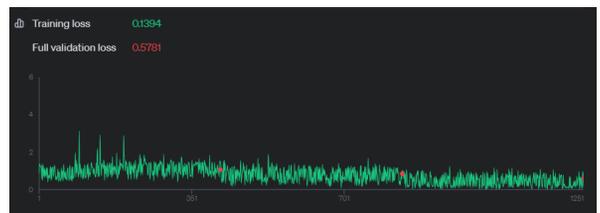


Fig. 16: Training result

Training loss: This is the average error of the model on the training dataset at each training step.

Full validation loss: This is the average error of the model on the validation dataset after each epoch.

The loss gradually decreases with the number of training steps, indicating that the model is learning to make more accurate predictions over time. After 1,251 steps, the final loss reaches 0.1394, a relatively low value, suggesting that the model has learned effectively. As the training progresses, the loss curve flattens, implying that the model has extracted the most helpful information from the dataset. Below the chart, messages provide details about the training process, documenting each stage from initialization to completion. Additionally, three checkpoints are recorded, as previously mentioned. The entire validation loss is 0.5781, which is higher than the training loss, indicating that the model performs better on the training dataset than on the validation dataset.

The total training time of 1 hour, 16 minutes, and 6 seconds, after nine steps, is shown in Table 7.

21:09:22 Fine-tuning initialization.

Table 7: Training description

Timestamp	Description
22:25:31	The job has been completed
22:25:30	Checkpoint created at step 1251 with Snapshot ID: ft:gpt-3.5-turbo-1106:personal:cdsdnag:9EeyXPDd:ckpt-step-1251
22:25:30	Checkpoint created at step 834 with Snapshot ID: ft:gpt-3.5-turbo-1106:personal:cdsdnag:9EeyXdw2:ckpt-step-834
22:25:30	Checkpoint created at step 417 with Snapshot ID: ft:gpt-3.5-turbo-1106:personal:cdsdnag:9EeyXyd1:ckpt-step-417
22:25:27	New fine-tuned model created: ft:gpt-3.5-turbo-1106:personal:cdsdnag:9EeyUjeI
21:15:00	Fine-tuning job started
21:10:25	Files validated, moving job to queued state
21:09:22	Validating training file: file-TVloqK5CjIYkRjYKUFhhdh and validation file: file-wynlhyQtV6UiMG92jC4QRIWi
21:09:22	Created fine-tuning job: ftjob-EW4w9aUEOcbecXFnKW9Pha6

21:09:22 Validating training file and validation file.

21:10:25 Validation completed and transitioned to the queue.

21:15:00 Fine-tuning process begins.

22:25:27 New fine-tuned model created.

22:25:30 Checkpoint created at step 417.

22:25:30 Checkpoint created at step 834.

22:25:30 Checkpoint created at step 1251.

22:25:31 Fine-tuning completed.

### Experimental results

The model's evaluation results after completing the chatbot training after 1,251 steps, with a checkpoint every 100 steps. The results regarding the metrics of training loss, training accuracy, validation loss, and validation mean token accuracy are described in Table 8.

### Conducting questions and answers

Users finally return to the 'Playground' section, select the trained model, and test the results shown in Figure 17.

### C. Comments and comparisons

The Transformer architecture is widely used over any other architecture. To clarify the argument of why most current machine learning

Table 8: Training results

Step	Train loss	Train accuracy	Validation loss	Validation means token accuracy
1	0.8999	0.80992	-	-
100	0.9137	0.75548	0.95679	0.76291
200	1.54335	0.6594	1.39699	0.66195
300	1.35187	0.66801	1.11632	0.69739
400	0.77926	0.77093	0.97236	0.74365
500	1.00285	0.78082	1.16552	0.70858
600	0.8933	0.77723	1.19546	0.69585
700	0.6195	0.8087	1.3136	0.67241
800	0.86886	0.77113	1.04009	0.71849
900	0.65797	0.81864	0.74824	0.77904
1000	0.42028	0.88591	0.62537	0.83168
1100	1.0163	0.74143	0.57982	0.81283
1200	0.80209	0.77808	0.54661	0.84452



Fig. 17: Questions and answers on digital transformation

technologies select the Transformer architecture, the research group has compared it with another powerful architecture, the Recurrent Neural Network (RNN), presented in Table 9.

Transformers, RNN, or LSTM, are powerful neural network architectures used in NLP. However, Transformer has several outstanding advantages over RNN, especially in terms of parallel learning capability, efficiency, and generalization ability. As a result, Transformer has become the most widely used neural network architecture in NLP today.

Like OpenAI, Rasa is another popular open-source platform for creating chatbots, and fine-tuning LLM like GPT 3.5-Turbo 1106. After researching both approaches, the following con-

Table 9: Comparing Transformer with RNN and Long Short-Term Memory (LSTM)

Architecture	Transformer	Recurrent Neural Network (RNN)	Long Short-Term Memory (LSTM)
Structure	Based on the mechanism of attention, processing data in parallel.	Based on sequential connections, processing sequential data.	Based on sequential connections, processing sequential data.
Advantages	Parallel learning, rapid training, efficient long sequence handling. Accurately understand context and generate fluent text. Strong generalization, easily adjustable. Supported by various libraries.	Simple, easy to understand, easy to implement. Efficient for sequential dependent tasks.	Simple, easy to understand and deploy. Effective for sequential tasks: speech recognition, text classification, and time series prediction. Stores previous step information for long sequence processing.
Disadvantages	Require large amounts of data. May be complex to deploy.	Slow data processing struggles with long sequences. Poor generalization ability, difficult to adjust. Less supported by libraries.	Slower data processing compared to Transformer, struggles with long sequences. Difficult to fine-tune for different NLP tasks. Less supported by libraries compared to the Transformer.

clusions are drawn in Table 10.

Table 10 shows that both approaches have strengths and weaknesses in generating text for conversations. However, the technological aspect is the key factor in choosing OpenAI over the Rasa Framework. Given the focus on supporting digital transformation for small and medium-sized enterprises, where most businesses have limited experience and knowledge in technology, fine-tuning a model rather than building a chatbot from scratch with Rasa presents a more optimized choice, saving both time and effort.

### V. CONCLUSION AND RECOMMENDATIONS

This paper has investigated the utilization of fine-tuning techniques to optimize ChatGPT for consultancy purposes, with a specific focus on aiding small and medium-sized businesses

Table 10: Comparing OpenAI with the Rasa framework

Evaluation criteria	OpenAI	Rasa framework
Creativity and engagement	Generates high-quality human-like text, leading to natural and engaging conversations.	Focuses on accurate and contextually appropriate responses, not as creative as GPT-3.5 Turbo 1106.
Access to massive data	Helps the model learn and improve performance by accessing a vast amount of text data.	Relies on user-provided data.
Ease of use	Provides user-friendly tools and APIs for creating and deploying chatbots easily.	Requires programming skills and technical knowledge to deploy and adjust chatbots.
Control over chatbot behavior	Less flexible in controlling specific chatbot behavior.	Allows better control over chatbot behavior by designing complex rules and detailed conversation flows.
Open source	Not open source, with potential costs for tools and APIs.	Open source, free to use and modify, with a large user community for support.
Difficulty in control	Increases challenges and time-consuming in fine-tuning in case of insufficient diversity in the personal data corpus.	Requires more technical knowledge to use, creating a chatbot can be more time-consuming and costly.
Lack of contextual understanding	May struggle to understand the nuances of human language and conversation context.	Needs to be trained on a large dataset of annotated conversations to operate effectively.
Cost	Incur expenses when utilizing the OpenAI API.	Costs associated with building and maintaining the infrastructure.

in their digital transformation efforts. By fine-tuning ChatGPT, it can provide more tailored guidance and support, offering customized solutions and actionable insights for navigating the complexities of digital transformation processes. Intending to empower businesses to thrive in the digital age, while ensuring the responsible deployment of AI technologies, the study recommends smarter, more effective consultancy solutions.

## REFERENCES

- [1] Nguyen VLM, Tam LC, Hung NV, Nguyen NDT, Hien LTH, Khiet LTN, et al. Codebot—a Vietnamese chatbot system for answering C++ and Python-related questions. *Ho Chi Minh City University of Education Journal of Science*. 2021;18(9): 1711–1723.
- [2] Mai TNT, Maxim S. Enhancing Rasa NLU model for Vietnamese chatbot. *International Journal of Open Information Technologies*. 2021;9(1): 31–36.
- [3] Vi Manh Tuyen, Do Viet Manh, Tran Duc Nghia, Nguyen Tuan Anh, Chu Thi Thanh, Nguyen Van Thang, et al. Chatbot solutions for health care support in Vietnamese region [Xây dựng giải pháp chatbot hỗ trợ chăm sóc sức khỏe trên miền tiếng Việt]. *Journal of Military Science and Technology [Tập chí Nghiên cứu Khoa học và Công nghệ quân sự]*. 2021;FEE Special Issue–2021: 225–229.
- [4] Misischia CV, Poecze F, Strauss C. Chatbots in customer service: Their relevance and impact on service quality. *Procedia Computer Science*. 2022;201: 421–428.
- [5] Kumar R, Ali MM. A review on chatbot design and implementation techniques. *International Research Journal of Engineering and Technology*. 2020;7(2): 2791–2800.
- [6] Ray PP. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*. 2023;3: 121–154. <https://doi.org/10.1016/j.iotcps.2023.04.003>.
- [7] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems. In: Guyon I, Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds.). *Proceedings of the 31<sup>st</sup> International Conference on Neural Information Processing Systems (NIPS 2017)*. 4–9 December 2017; Long Beach, CA, USA. NY, United States: Curran Associates Inc.; 2017. p.6000–6010.
- [8] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, et al. A comprehensive survey on transfer learning. *Proceedings of the IEEE*. 2020;109(1): 43–76. <https://doi.org/10.1109/JPROC.2020.3004555>
- [9] Ministry of Planning and Investment of Vietnam. *Digital transformation handbook for businesses in Vietnam [Sổ tay chuyển đổi số cho doanh nghiệp tại Việt Nam]*. Hanoi, Vietnam: Ministry of Planning and Investment of Vietnam; 2021.
- [10] Ministry of Planning and Investment of Vietnam. *Digital transformation guidebook for small and medium enterprises in agriculture [Sổ tay hướng dẫn chuyển đổi số cho DNNVV trong nông nghiệp]*. Hanoi, Vietnam: Ministry of Planning and Investment of Vietnam; 2022.
- [11] Ministry of Information and Communications of Vietnam. *Digital transformation handbook [Cẩm nang chuyển đổi số]*. Hanoi, Vietnam: Ministry of Information and Communications; 2021.
- [12] Agency for Enterprise Development. *Barriers and Demand for Digital Transformation [Rào cản và nhu cầu chuyển đổi số]*. Hanoi, Vietnam: Enterprise Development Agency; 2021.
- [13] Pham Huy Giao. Digital transformation: Nature, practice and application [Chuyển đổi số: Bản chất, thực tiễn và ứng dụng]. *Petrovietnam Journal [Tập chí Dầu khí]*. 2020;12: 12–16. <https://doi.org/10.47800/PVJ.2020.12-02>.
- [14] Watkins J, Nguyen Quang Trung, Nkhoma M, Vo Khanh Thien, Nguyen Le Hoang Long. *Digital Transformation in Vietnam: A survey of small and medium enterprises (SMEs) and state-owned enterprises (SOEs) [Chuyển đổi số ở Việt Nam: Khảo sát các doanh nghiệp nhỏ và vừa (DNVVN), và doanh nghiệp nhà nước (DNNN)]*. <https://www.rmit.edu.vn/content/dam/rmit/vn/en/assets-for-production/documents/pdfs/code/code-digital-transformation-whitepaper-vie.pdf> [Accessed 23 February 2025].

