**AgroGPT**

**Abstract:**

Agriculture has always been a significant part of the economy worldwide. But there are several issues that farmers face from day to day life and they hardly find solutions to it. Many researches and tools were built to solve such problems and some of them were quite successful. But those contain some limitations. Agro-GPT is a tool that was brought to solve issues related to farming practices. In this paper, we will discuss how Agro-GPT can be a power tool for agriculture related queries and how it can contribute in this sector to a great extent. Agro-GPT was basically created using NLP techniques with a dataset related to farmer queries. It was fine-tuned using LLm models which are generative in nature and provided a satisfactory output.

**Introduction:**

Agriculture is one of the most important sectors in the world. However, traditional farming methods are not sustainable in the long run. AgroGPT is a revolutionary technology that can help farmers increase their yields, reduce costs, and minimize environmental impact. AgroGPT is an AI-powered agricultural platform that utilizes machine learning algorithms to solve queries that can help the farmers to be easily informed and optimize their farming practices. In the orthodox process experts have to repeatedly advise the same thing to different farmers. Usually it is hard for farmers to find a solution to their problems instantly. The whole process of finding the solution to their problems usually ends up wasting a lot of time. Sometimes the information received by the farmers by traditional process can also be misleading. The lack of a proper platform for agricultural assistance with queries and concerns of the farmers.

E-Agro is a chat room interface and chat bot which was developed to discuss the lasting farming problems and help farmers to make proper decisions on farming practices (1). The farmers, expertise and stakeholders were surveyed and a set of questions was taken into account. Artificial intelligence Markup Language (AIML) was used for training the model. A specialized markup language called Artificial Intelligence Markup Language (AIML) is used to build conversational agents and chatbots that follow a certain pattern-matching and response-generation methodology. It predicts intents based on the provided examples. It is built on XML and enables programmers to design conversational agents that have more human-like comprehension and response to human inputs. The intents which users want to know are generated following the pattern of the input examples. The information base for AIML is categorized. Every category has two patterns and templates in it. Templates hold the answers that the chatbot should provide when it finds a matching pattern, while patterns are utilized to match user input. The chatbot was tested and it was found that it could answer the atomic questions and the farmers accepted it as a user friendly tool.

Whereas Farmer Assistant Chatbot is built using the Naive Bayes algorithm that analyzes the user's queries and understands the message (2). This system is a web application that provides answers to the questions of the farmer. It was modeled in such a way as the Information Repository by a connected graph where the nodes contain information and links interrelate the information nodes. The design semantics includes AIML (Artificial Intelligence Markup Language) specification language for authoring the information repository such that chat robot design separates the information repository from the natural language interface component. The system uses built-in Naive Bayes to answer the query. The answers are appropriate to what the user queries. The user can query any farmer-related activities through the system. The user does not have to go to the office for inquiry personally. The system analyzes the question and then answers to the user. The system answers the query as if the person responds to it. With the help of Naive Bayes, the system answers the query asked by the farmer. The system replies using an effective Graphical user interface, which implies that a real person is talking to the user. The user can query about farmer-related activities online with the help of this web application—a talk bot, a virtual conversational assistant. Farmers can interact with the bot in a very simple manner. The focus is on developing the bot more intellectually so that it can even understand not-so-well-grammatically defined sentences. The bot uses the Natural Language Processing technique to parse the user queries.

However, it has some limitations. The fact that the E-Agro was only trained on atomic questions failed to answer composite questions. That is if the needed data is not entered into the Chat-Bot's knowledge base, it cannot respond. As a result, new examples and intentions must be added to the Chat-Bot's knowledge base. They don’t generate new context based on the input as they rely on predefined patterns and responses. They deliver preset responses linked to predefined patterns found in their AIML scripts by matching user input to those patterns. The answers are kept in the AIML knowledge base and are basically pre-written. Unlike AIML, GPT3 or GPT4 which are generative AI models can generate more human-like text based on inputs without relying on predefined responses. In the Farmer assistant chatbot, the chatbot was designed with a minimal dataset. The accuracy would be much lower, and the answer it would give would be inaccurate most of the time. The Naive Bayes algorithm was used to create this chatbot. It is known for its simplicity, computational efficiency, and ease of interpretation. However, the Naive Bayes algorithm can not handle complex problems. So, in complicated situations, this algorithm must be affected more to provide accurate results.

We have already got a couple of limitations by analyzing those two research papers. Now, it is important to know how we solve their limitations. We found a large dataset of 189K question & answer of farmer's frequently asked questions. After having the dataset, we

pre-possessed the dataset with the help of pandas. After cleaning the dataset, we jump into the coding section. In the coding section, we tried out 2 platforms to train our dataset. One is [Lamini.ai](http://lamini.ai) and another is Happy Transformers.

Happy Transformers is built on top of HuggingFace trainer. As we're working with Large Language Models (LLMs) to get generative answers. We trained our models on Pythia-70M, GPT-Neo-125M, Pythia-410M parameters in happy transformers.

Besides, [Lamini.ai](http://lamini.ai) is the LLM platform for enterprises and developers to build customized, private models: easier, faster, and higher-performing than any general LLMs. To fit our dataset into a Lamini based framework, we convert our dataset into JSON format. Which was in CSV format earlier. After getting the JSON format, we again convert it to JSONL format. After that, we use GPT-Neo-125, Pythia-410 parameters on [lamini.ai](http://lamini.ai) to train our dataset.

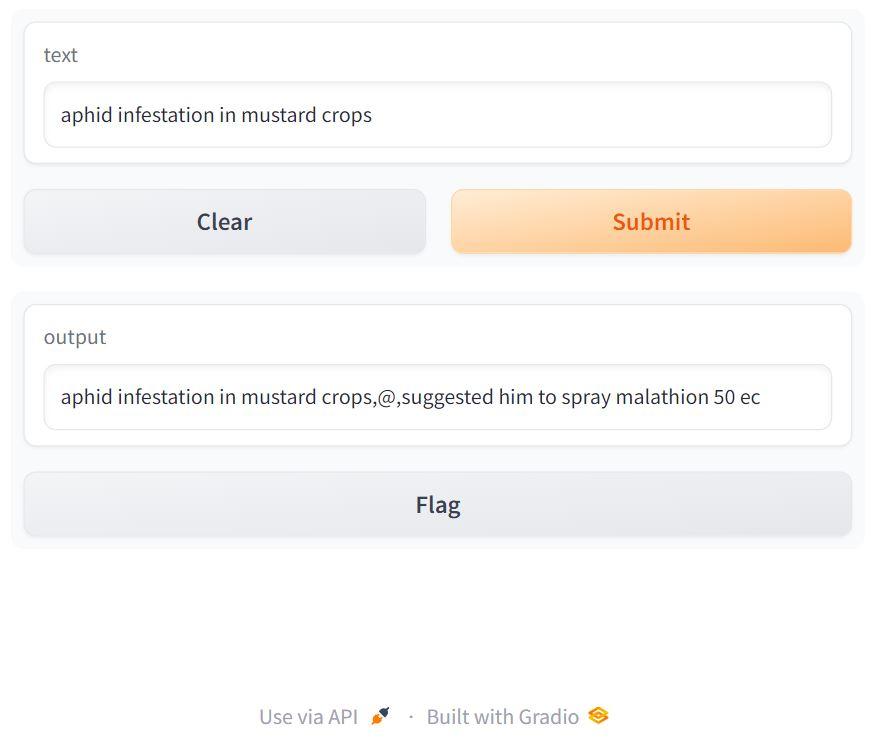
Our unique contribution -

• Large dataset

• Use Large Language Models

• Fine-tuned LLMs to generative answers based on our custom dataset

**Input - Output:**



**Related Work:**

**NLP & ML Algorithm in Agricultural Domain:**

Agribot was implemented using a simple Neural network-based model to answer the queries of the farmer (3). Their dataset is from Kissan Call Centre. The dataset is processed in the neural network and it extracts an entity from it. Sen2Vec Model, which can transform sentences to vectors was used to categorize similar sentences. The most relevant answers to the queries were generated. The accuracy of a sentence embedding model was improved from 56% to 86% with this implementation.

E-AGRO: Intelligent Chat-Bot. IoT and Artificial Intelligence to Enhance Farming Industry (March 2020)

E-Agro is a chat bot which was developed to discuss the lasting farming problems. Artificial intelligence Markup Language (AIML) was used for training the model (1). It predicts intents based on the provided examples which are built based on XML. The chatbot was tested and it was found that it could answer the atomic questions and the farmers accepted it as a user friendly tool.

The Naive Bayes algorithm was used in the construction of the Farmer Assistant Chatbot, which interprets user inquiries and comprehends their meaning (2). This system is an online application that responds to inquiries from farmers. The Information Repository was modeled as a connected graph with nodes representing information and links connecting the nodes. In order to author the information repository and keep the information repository and the natural language interface component apart in chat robot design, the design semantics incorporate the AIML (Artificial Intelligence Markup Language) specification language. Built-in Naive Bayes is used by the system to respond to the question. The responses match the questions posed by the user. The bot uses the Natural Language Processing technique to parse the user queries.

Agroxpert, a chatbot, was built using NLP techniques. For farmers, the technology will function as an interactive virtual assistant, providing answers to all of their questions about agriculture (4). For farmers, the technology will function as an interactive virtual assistant, providing answers to all of their questions about agriculture. The chatbot was built using chatterbot, a Python conversational dialogue engine called ChatterBot uses machine learning to generate responses based on collections of previously recorded talks. ChatterBot's language-independent nature makes it possible to train it to speak any language. When asked a question that corresponds to the training data set, the chatbot can correctly respond to it.The threshold similarity index can be adjusted to the best match, and the chatterbot employs the best match model. The answer is 100% accurate if the query is exactly the same as the one that is in the dataset. However, if the query differs slightly from the dataset, the best match method and threshold similarity index will determine how accurate the answer is. If the sentence is not exactly as it appears in the dataset, there is 90% accuracy in obtaining the needed response.

In a multi user chat room, an AgroBot that integrates natural language processing and machine learning responds to inquiries regarding farming (5). The application will employ the Deep Learning CNN algorithm to predict disease from that crop leaf and present possible treatments once the chatbot asks the farmer to share an image of their crop. When the user asks a question about a crop, such as what kind of crop it is, the Chatbot will respond with information about soil, rainfall, and other factors.

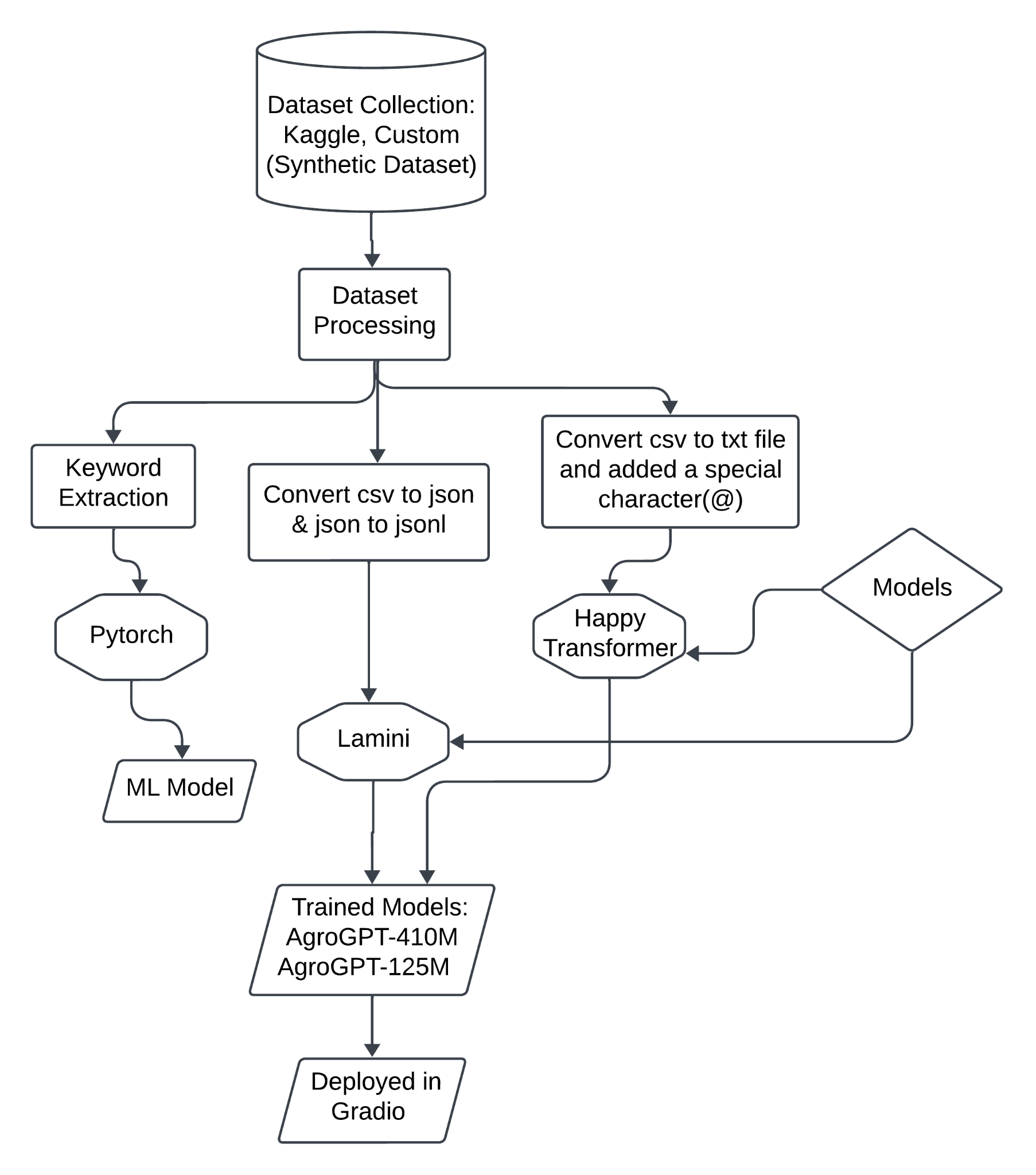
**LLMs:**

The Hua Tuo model is the first open-source Chinese biomedical LLM tuned with knowledge-based instruction data. This integrates structured and unstructured medical knowledge from CMeKG, ensuring our model has accurate and domain-specific knowledge (6). We constructed a test set of potential questions in Chinese dialogue scenarios and compared the generated responses of our HuaTuo model with three other baseline models. To evaluate the model performance, we recruited five notators with medical backgrounds who assessed the randomly mixed responses of the models using a three-point scale for each dimension of Safety, Usability, and Smoothness (SUS).

“ChatDoctor” aims to improve the medical knowledge of large language models (LLMs) like ChatGPT by creating a specialized model with enhanced medical advice accuracy (7). The model was refined using a dataset of 100,000 patient-doctor dialogues from an online medical consultation platform, anonymized for privacy, and incorporated a self-directed information retrieval mechanism. The model's fine-tuning with real-world interactions improved its ability to understand patient needs and provide informed advice. The proposed ChatDoctor represents a significant advancement in medical LLMs, demonstrating an improved understanding of patient inquiries and accurate advice.

**Methodology:**

Flowchart:

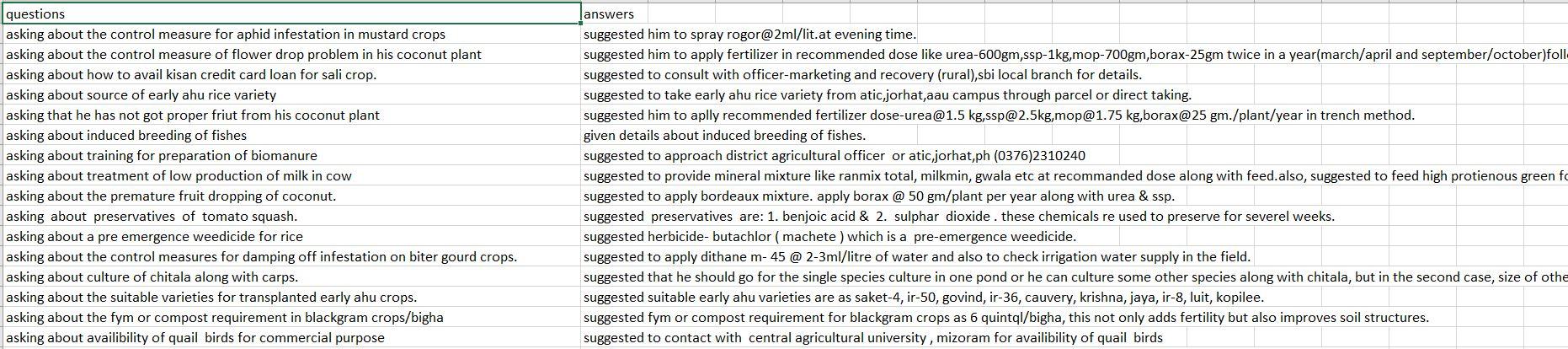
****

**Dataset:**

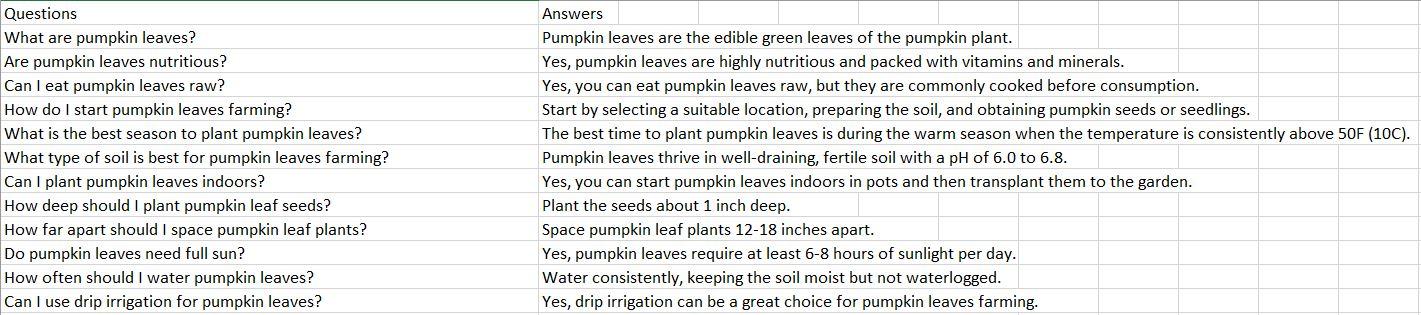
There are two types of dataset we are using. First one, we collected a dataset from kaggle which is based on Farmers Call Query Data (source – Kaggle). This data set was generated using data from data.gov.in, an open data platform by Govt. of India. Data is of Kisan Call Centre where farmers called for some query over phone call, and experts replied with some answers. Dataset has two columns: ‘questions:’ asked by farmers ‘answers:’ reply from experts. This dataset has almost nearly 1.7 lakh data.

Another one, Synthetic dataset (AI that has been educated on real-world data samples creates synthetic data) made by our group mates using ChatGPT. We select 60 topics (vegetables, fruits, animals, fishes, crops etc). Each topic has at least 80 questions. So, it’s around (60\*80 = 4800 pairs of questions answers).

**Farmers Call Query Data (source – Kaggle):**

****

**Synthetic dataset using ChatGPT (source – made by our group mates)**

****

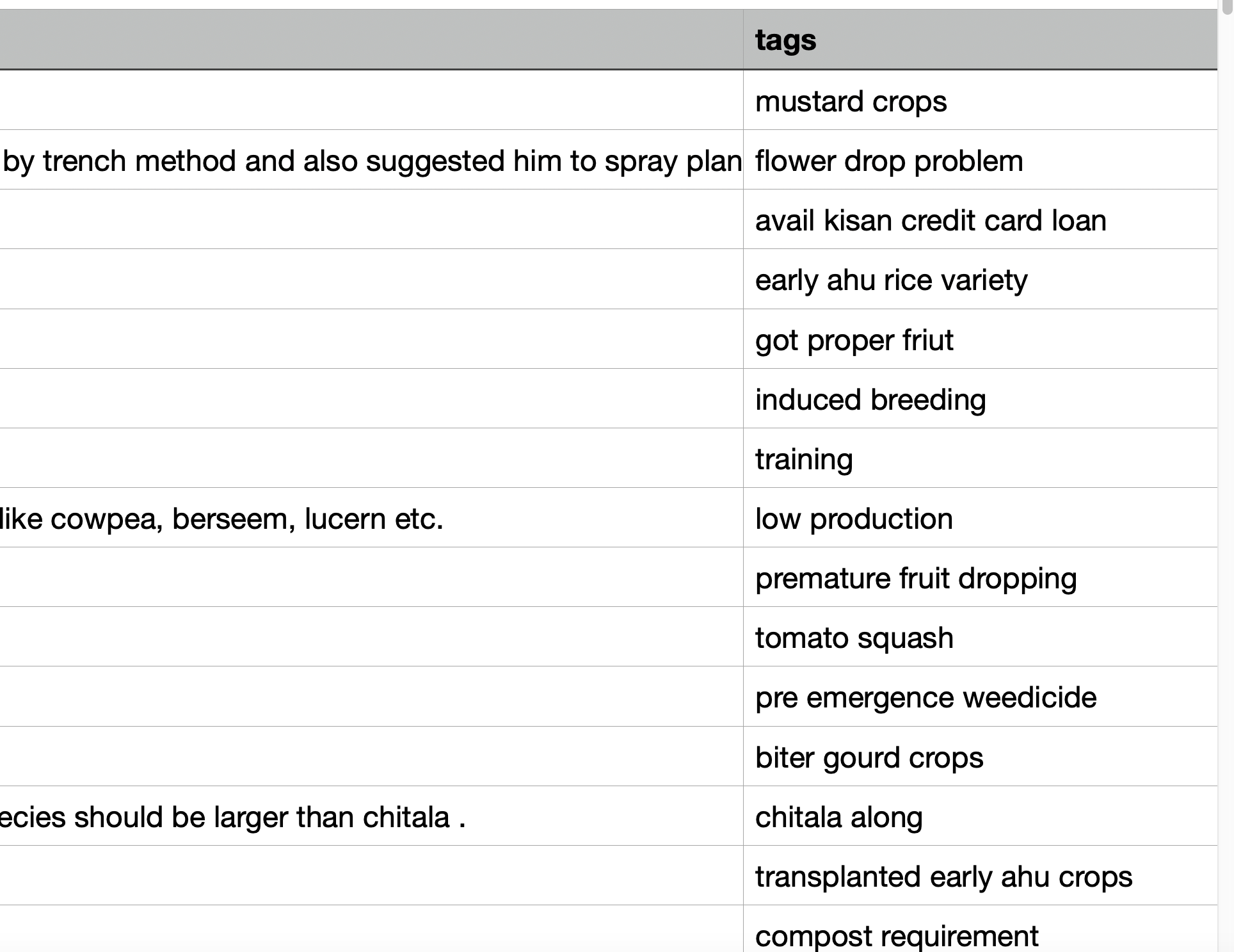
**Language/Framework/platform: ​**

* Pytorch **​**

**Keyword Extraction:**

Since we wanted to generate tags that are a unique keyword that will be generated after summarizing the questions and answers column of the dataset, we started to look for different methods of keyword extraction. They found some models like ‘rake nltk’, ‘yake’, ‘KEA’, ‘TextRank’. Out of all of these, The rake nltk gave better results.

Keywords extracted for each questions and answers and named as tags column:

****

**Happy Transformer:**

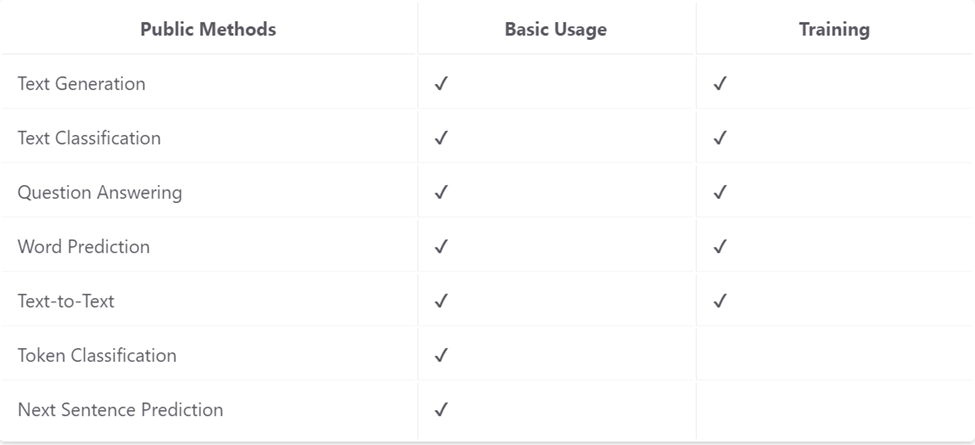
**What is Happy Transformer?**

Happy Transformer makes it easy to fine-tune and perform inference with NLP Transformer models. *Happy Transformer* is PyPi Python package built on top of Hugging Face's transformer library that makes it easy to utilize state-of-the-art NLP models.

**Installation process:**

pip install happy transformer

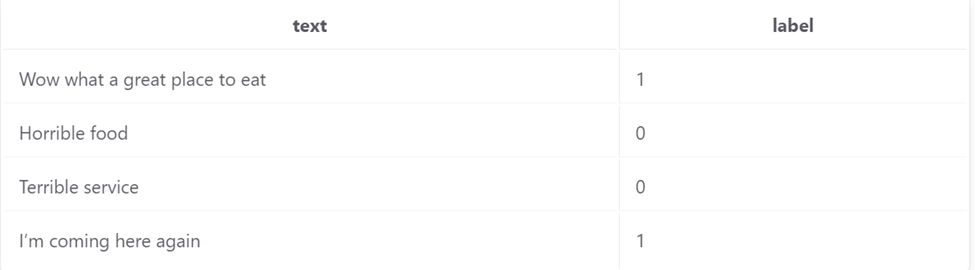
**Features:**

****

From the following features we select Text Generation. Because other methods are not useful for our project. But why?

In Text Classification, it only helps us to identify whether a sentence is negative or positive. So, it needs 2 types of parameters (Text, Label).

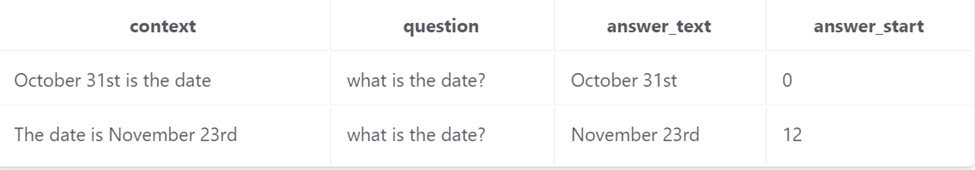
Example is shown below:

****

This process is not applicable for our project due to the formation of dataset.

Then we move to the Question Answering Fine tuning method. This process needs 4 types of parameters.

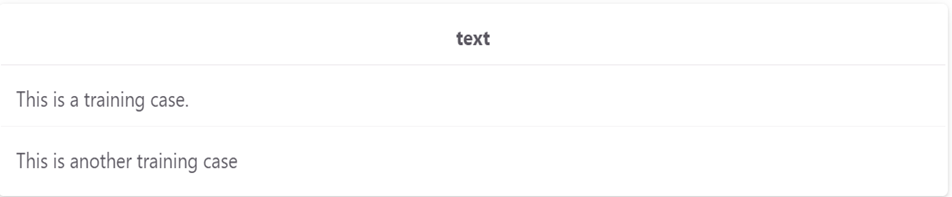
Like this picture –

****

This type of dataset format is too hard to set up. The dataset we are working on has two sets of parameters (Questions, Answers), but in this method, the dataset must have four sets of parameters, which is unsuitable for our work.

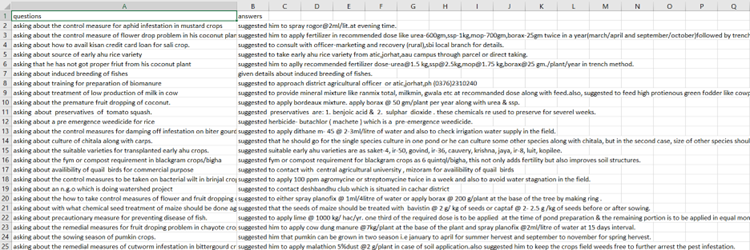
In short, other methods are not applicable for Text generation. As far as we figure out that

Text Generation method is the best option for our project. For fine tuning with this method the dataset should be like this one -

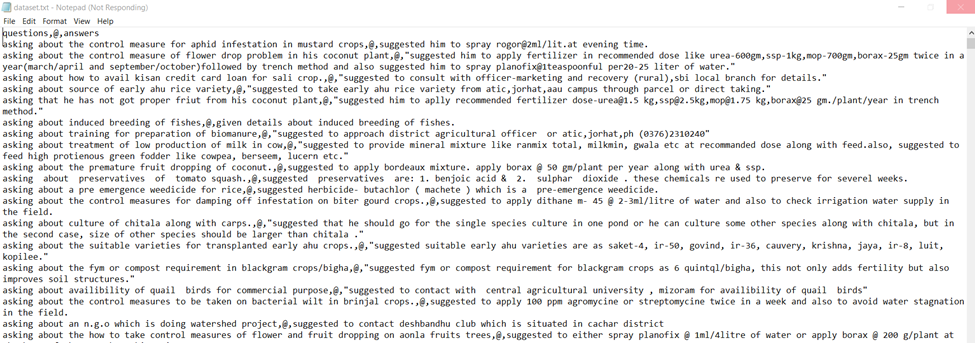
****

So, we have to convert our dataset into this format.

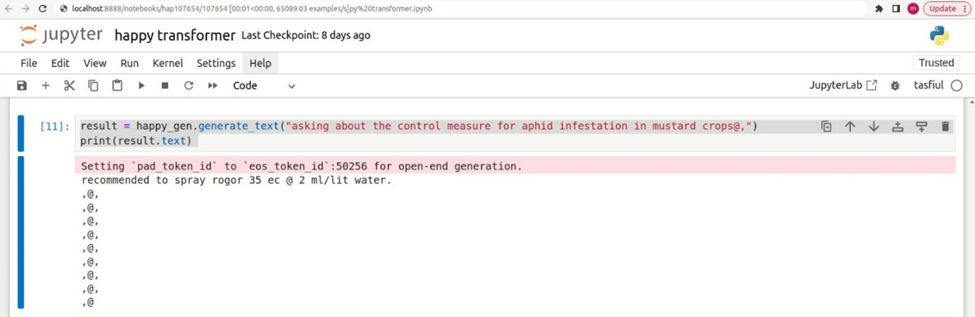
**Before converting our dataset:**

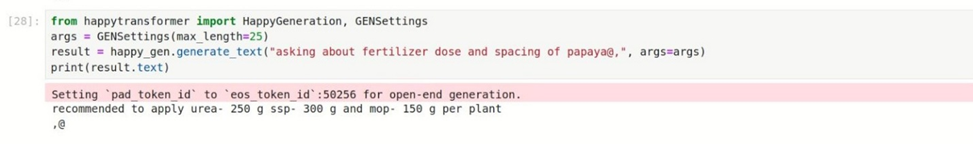
****

**After converting that format:**

****

We use a special character (@) to separate and help the model to learn which part is the question and which one is the answer. We get better results using this special sign. As shown below:

****

****

**Limitations of Happy Transformer (Text-Generation):**

1. In happy transformer there is no option for saving the checkpoints.

2. Even we work with a larger dataset (180k), after fine tuning it fail to generate larger response. As we shown here.

3. To better response we have to use an special character.

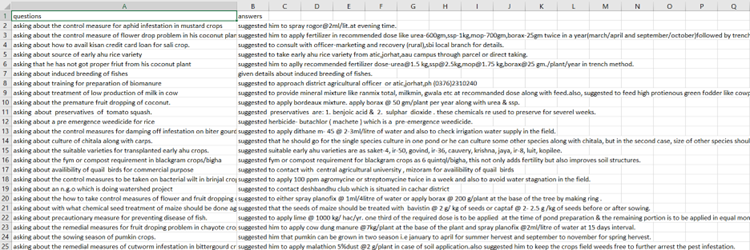
4. It’s dataset should be a single line for text or a single column for csv file. So, if this is harder to implement and get better response.

**Lamini:  
  
What is Lamini?**Lamini is the LLM platform for enterprises and developers to build customized, private models: easier, faster, and higher performing than any general LLMs.  
 List of finetuning demos:  
 I) Finetuning your custom LLM  
 II) Question Answer LLM finetuning  
 III) Llama V2 Parameter Efficient finetuning  
 IV) Finetuned vs. Non finetuned LLMs  
 **Installation process:**

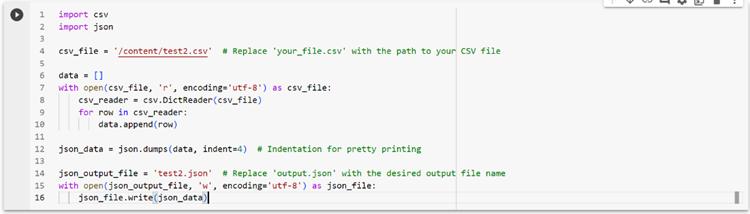
Step 1: Authenticate with Google

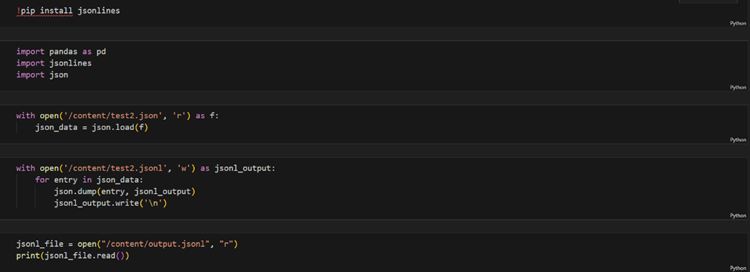
Step 2: Installation of Lamini:

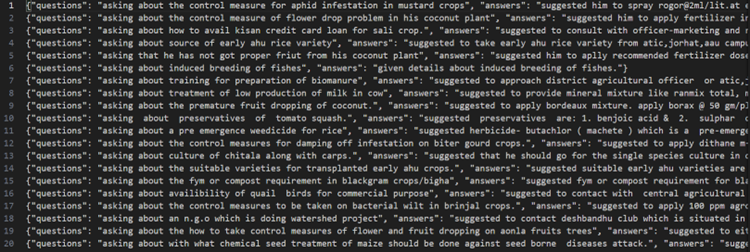
!pip install --upgrade --force-reinstall --ignore-installed lamini  
  
 Our dataset:

****Lamini only supports JSONL format in their dataset. But our dataset was in CSV format. So, we convert the dataset from CSV to JSON and then JSON to JSONL.

**CSV to JSON:**

**  
  
 JSON to JSONL:**

**  
  
  
 Dataset after conversion (JSONL formatted):**

****

**Limitations of Lamini:**1.Fine Tuning only with small models (~400M); < 250K tokens.

2.Have to train only on their servers.

3. Have no facility to export weights.

4.We cannot set the arguments (learning rate, number of training epochs, batch size etc )

5. Dataset should be large enough to get a proper output.

**Large Language Models:**

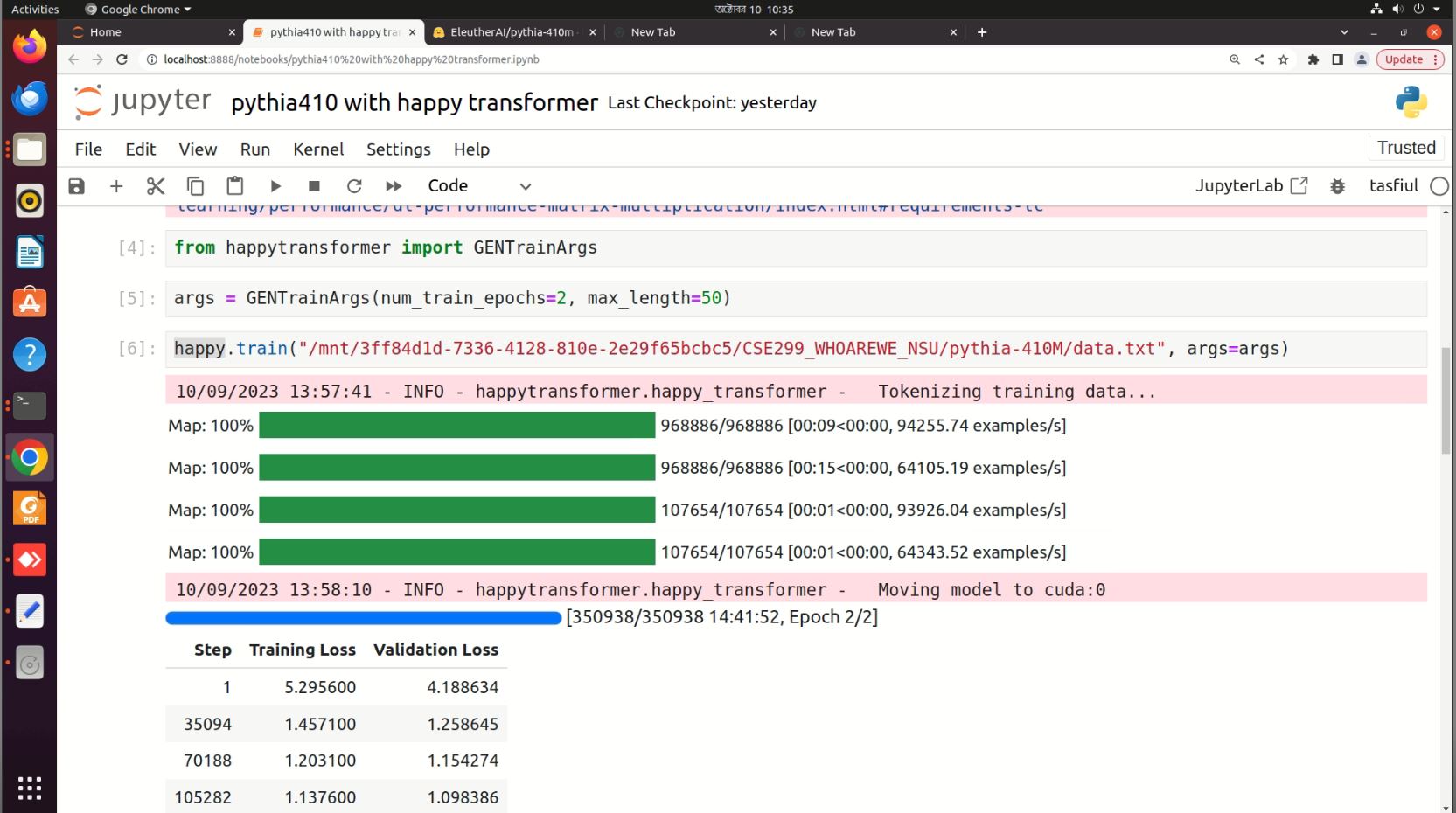
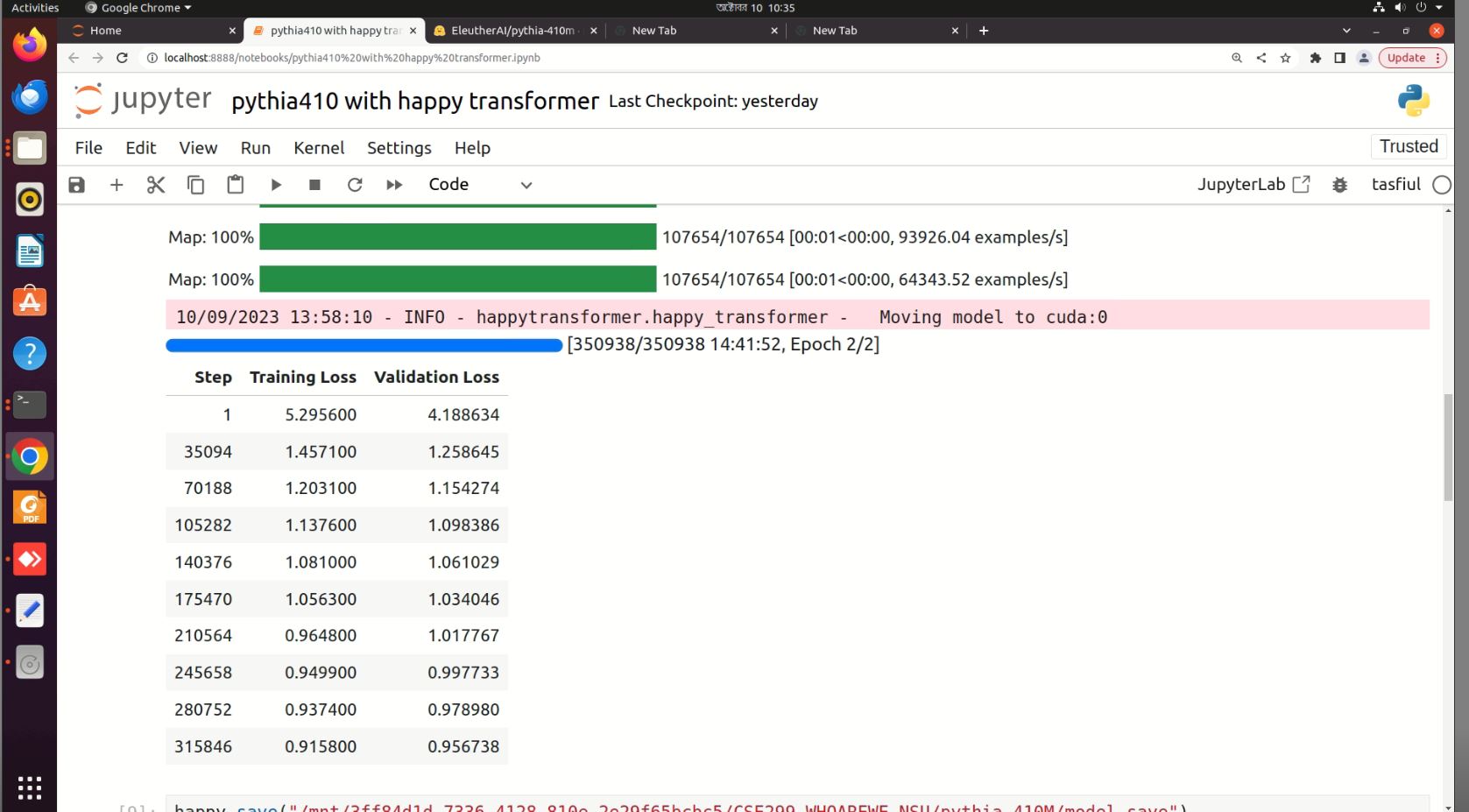
* **EleutherAI/pythia-70m,​**
* **EleutherAI/gpt-neo-125m,​**
* **EleutherAI/pythia-410m, ​**
* **EleutherAI/gpt-neo-1.3B**

**Training parameters:**

****

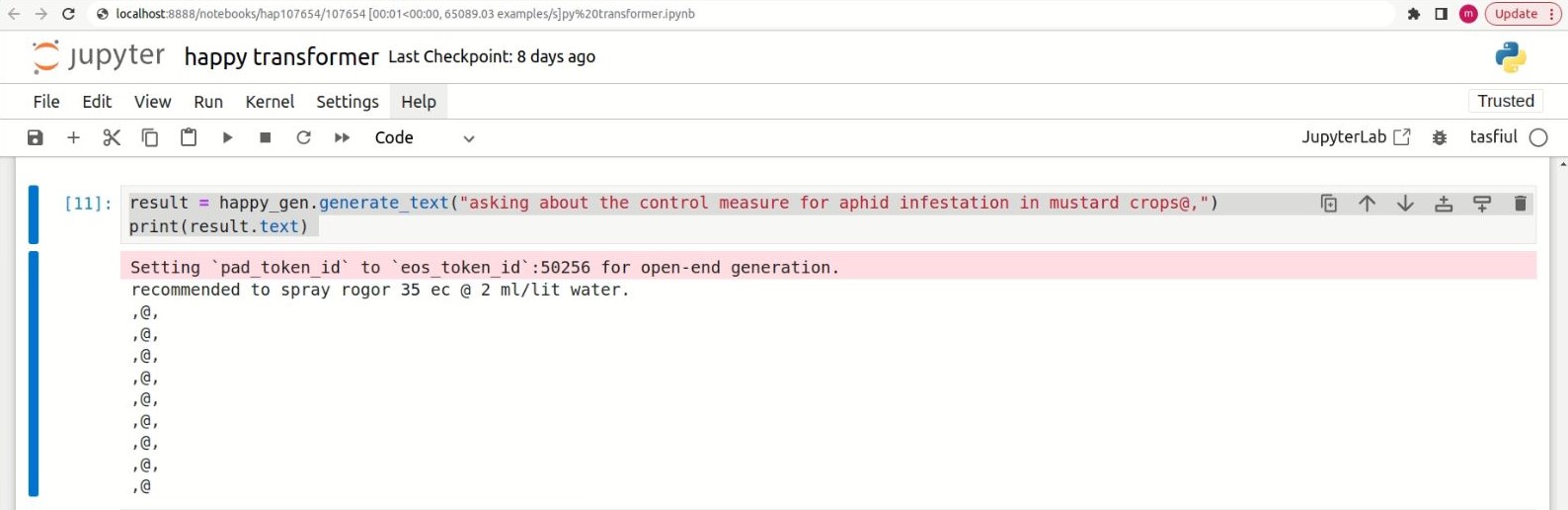
**Result:**

**Training section:**

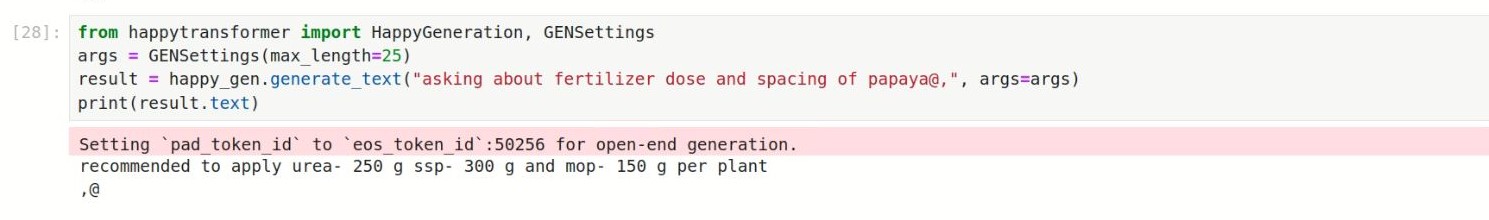
****

**Testing Output:**

**Testing part (1):**

****

**Testing part (2):**

****

**Conclusion & future work:**

Global economy has always depended heavily on agriculture. However, farmers deal with a number of problems on a daily basis for which they seldom find solutions. To address these issues, numerous studies and instruments were developed, some of which proved to be highly effective. However, those have certain restrictions. Agro-GPT is a tool designed to address problems pertaining to agricultural techniques. In this paper, we'll talk about how Agro-GPT can be a powerful tool for answering questions about agriculture and how it can significantly advance this field. Essentially, NLP techniques were used to create Agro-GPT using a dataset pertaining to farmer queries. It was refined with the help of generative LLm models, which produced a good result.

In future we will train our dataset on a few different models. Then if we get good results. Then we will think about working in the Bangla language. As we’re working on an English dataset, we assume this work will impact people worldwide.

**Reference:  
  
[**1] J. Ekanayake and S. Luckshitha, “E-AGRO: Intelligent Chat-Bot. IoT and artificial intelligence to enhance farming industry,” *AGRIS On-line Papers in Economics and Informatics*, vol. 12, no. 01, pp. 15–21, Mar. 2020, doi: 10.7160/aol.2020.120102.

[2] P. Kaviya, “Artificial intelligence based Farmer Assistant Chatbot,” Apr. 14, 2021.<https://journal.ijresm.com/index.php/ijresm/article/view/631>  
  
[3] N. Jain *et al.*, “AgriBot: Agriculture-Specific Question Answer System,” *AgriBot: Agriculture-Specific Question Answer System*, Jun. 2019, doi: 10.35543/osf.io/3qp98.

[4] V. Nayak, P. Nayak, Sampoorna, Aishwarya, and N. H. Sowmya, “Agroxpert - Farmer assistant,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 506–512, Nov. 2021, doi: 10.1016/j.gltp.2021.08.016.

[5] B.A. Abhale, Jayshri Bagul, Shital Chavan, Tejashri Parjane, and Payal Tribhuvan, “Agriculture Assistant Chatbot using Artificial Neural Network,” *IJIRMPS*, May 2023, [Online]. Available: https://www.ijirmps.org/research-paper.php?id=230108

[6]“Papers with Code - HuaTuo: Tuning LLaMA Model with Chinese Medical Knowledge,” Apr. 14, 2023. [https://paperswithcode.com/paper/huatuo-tuning-llama-model-with-chinese](https://paperswithcode.com/paper/huatuo-tuning-llama-model-with-chinese?fbclid=IwAR1xlJ_VuLar4wG3FVDM0XBa6CWY0Amv2-lJna4CmbIvl2cvE9Bs5u0oE-o)

[7] “Papers with Code - ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge,” Mar. 24, 2023. [https://paperswithcode.com/paper/chatdoctor-a-medical-chat-model-fine-tuned-on](https://paperswithcode.com/paper/chatdoctor-a-medical-chat-model-fine-tuned-on?fbclid=IwAR1zxqGk8BDecAokaUwRjNCHa0sQG-2OE8EaDsdWElAl-v1o0WAY-fxMaGU)