



# Kannada Chatbot : A Transformer Based Approach

Authors: Angad Sandhu, Isha Nema, Arisetty Sree  
Ashish, Musica Supriya

# Introduction

- Indian Subcontinent is home to about 121 separate languages, each with its own inner workings and structures.
- Chatbots have been trained in various languages, such as English, French, and Russian. Multilingual chatbots trained in niche Indic languages, the language barrier obstructing native speakers from accessing important information can be overcome.
- The purpose of this chatbot is to assist Kannada-speaking farmers to have a better connect with vendors by eliminating the need for a middleman for translation.

# Literature Survey

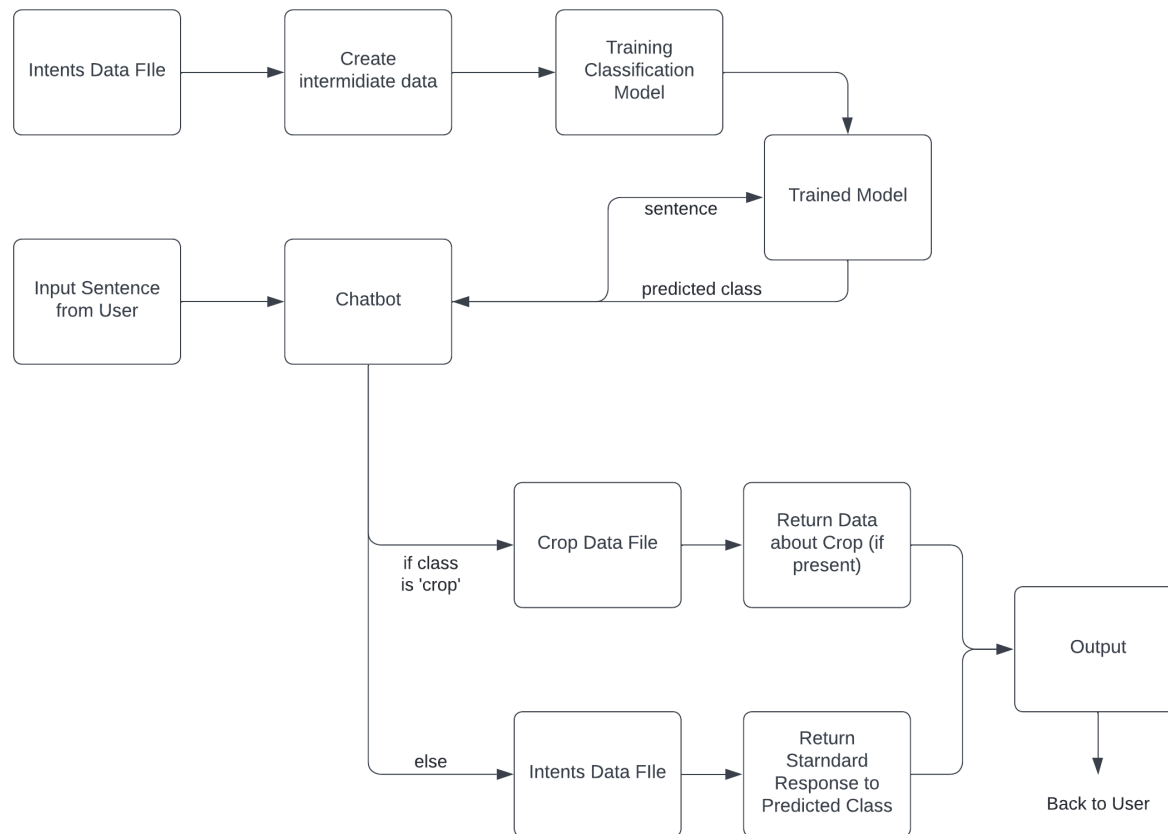
- In Kiruthika et al., the system collects details of the farmer, land, and cultivated crops and maintains a database. A function queries the database, retrieves the information, and suggests it to the user based on a condition [1].
- NLP techniques used by Vijayalakshmi et al. conversational assistants to comprehend user inquiries in their native tongue. [3]
- The system by Yashaswini et al. gets information from government websites and repositories. This information is cleaned to remove any noise and undesired, unrelated information before being safely stored in a database. It is trained thoroughly using machine learning in TensorFlow architecture using the KNN algorithm [5].

# Methodology

- Two Techniques for creating chatbots are: Simple Chatbot, Intelligent Chatbot
- Hybrid Chatbot implementation has been proposed in this paper.
- Hybrid Chatbots take advantages of both simple and intelligent chatbots



# Methodology Flow

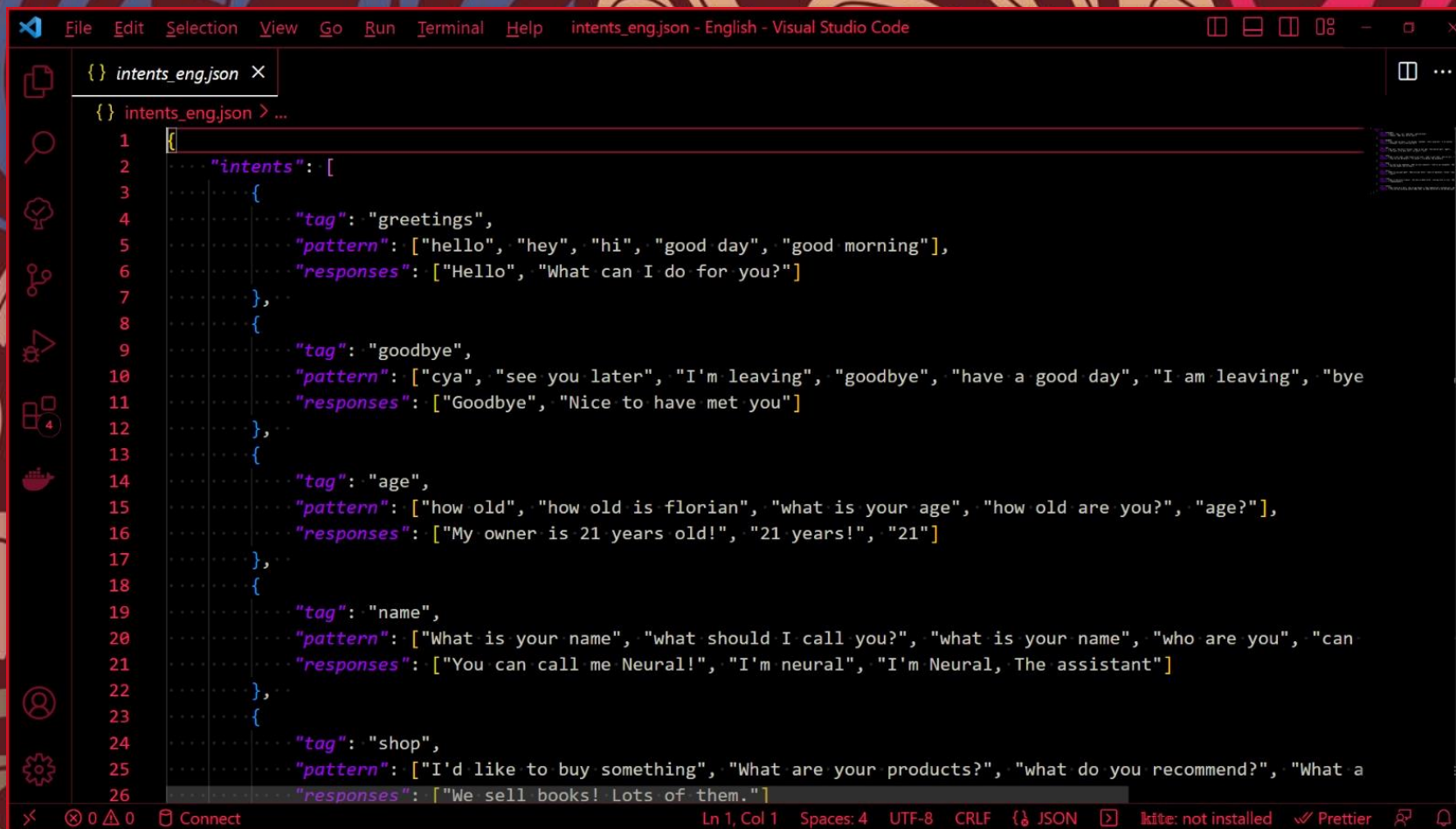


# Data Collection

- The first step of the data collection process used in the present study was to retrieve or create a set of fixed questions and answers as used by classical chatbots.
- All sets of questions and answers were distributed within a predefined number of classes.
- After all such classes were created, the data was put into a JSON file
- The file is a list of dictionaries with 3 attributes - tag, patterns and responses.



# Intents.json

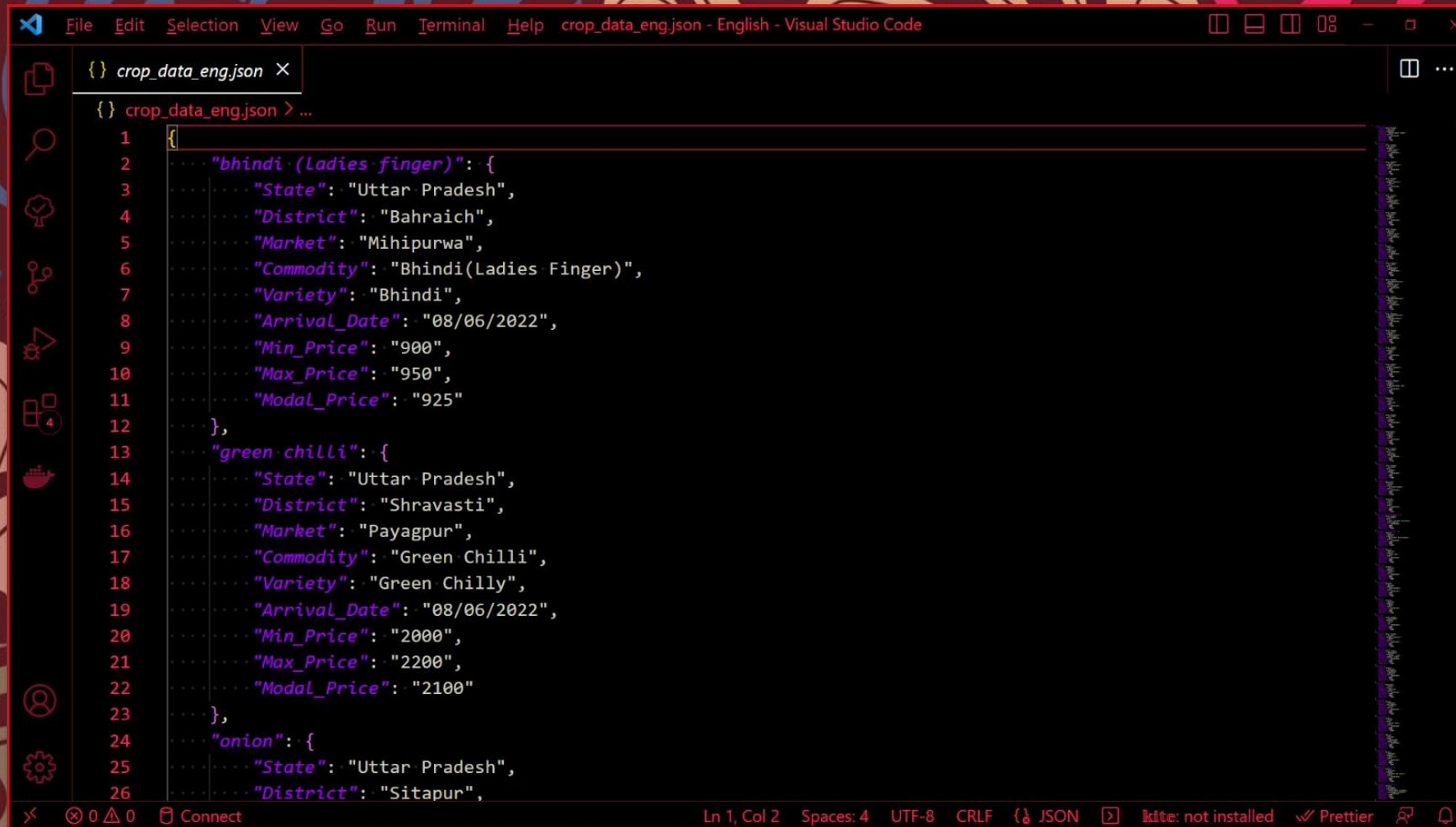


```
{
  "intents": [
    {
      "tag": "greetings",
      "pattern": ["hello", "hey", "hi", "good day", "good morning"],
      "responses": ["Hello", "What can I do for you?"]
    },
    {
      "tag": "goodbye",
      "pattern": ["cya", "see you later", "I'm leaving", "goodbye", "have a good day", "I am leaving", "bye"],
      "responses": ["Goodbye", "Nice to have met you"]
    },
    {
      "tag": "age",
      "pattern": ["how old", "how old is florian", "what is your age", "how old are you?", "age?"],
      "responses": ["My owner is 21 years old!", "21 years!", "21"]
    },
    {
      "tag": "name",
      "pattern": ["What is your name", "what should I call you?", "what is your name", "who are you", "can"],
      "responses": ["You can call me Neural!", "I'm neural", "I'm Neural, The assistant"]
    },
    {
      "tag": "shop",
      "pattern": ["I'd like to buy something", "What are your products?", "what do you recommend?", "What a"],
      "responses": ["We sell books! Lots of them."]
    }
  ]
}
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} JSON linter: not installed ✓ Prettier



# crop-data.json



```
1  {  
2    "bhindi (Ladies finger)": {  
3      "State": "Uttar Pradesh",  
4      "District": "Bahraich",  
5      "Market": "Mihipurwa",  
6      "Commodity": "Bhindi (Ladies Finger)",  
7      "Variety": "Bhindi",  
8      "Arrival_Date": "08/06/2022",  
9      "Min_Price": "900",  
10     "Max_Price": "950",  
11     "Modal_Price": "925"  
12   },  
13   "green chilli": {  
14     "State": "Uttar Pradesh",  
15     "District": "Shravasti",  
16     "Market": "Payagpur",  
17     "Commodity": "Green Chilli",  
18     "Variety": "Green Chilly",  
19     "Arrival_Date": "08/06/2022",  
20     "Min_Price": "2000",  
21     "Max_Price": "2200",  
22     "Modal_Price": "2100"  
23   },  
24   "onion": {  
25     "State": "Uttar Pradesh",  
26     "District": "Sitapur",
```

Ln 1, Col 2 Spaces: 4 UTF-8 CRLF {} JSON kite: not installed Prettier



# Analysis

- Create intermediate data structures (would be used later in training):
  - “words” is a list of all patterns broken down into individual words and appended together
  - “classes” is a list of all 7 class names
  - “docs” is a list of tuples with individual patterns and their corresponding classes.
- stopwords are removed, data frame is shuffled, split into “Training” and “Validation” sets.
- The text data is passed and tokens are converted to one-hot encoding for representation.

# Evaluation

- After the model is trained and ready to use, the chatbot is implemented.
- The chatbot scans the input sentence and passes through the model to generate a list of classes with their corresponding probabilities.
- The class with the highest probability is considered.
- Response is returned to the predicted class as the answer to the input sentence.



# MLFF Network

Metric	Value
Loss	0.587
Accuracy	0.8213
Precision	0.900
Recall	0.619
F1 Score	0.7335
Mean Absolute Error	0.097

Table 1: Metrics from MLFF Network



# Transformer Network

Class	Accuracy
Greetings	1.0
Goodbye	0.9923
Age	1.0
Name	0.9722
Shop	0.9629
Hours	0.9880
Crop	1.0
<b>Average Accuracy</b>	<b>0.9879</b>

Table 2: Class-wise Prediction Accuracy of Transformer

Metric	Value
Accuracy	0.9879
Precision	0.9437
Recall	0.8708
F1 Score	0.9057

Table 3: Metrics from Transformer Network



# Accuracy Difference

Metric	Increase (%)
Accuracy	20.28
Precision	4.85
Recall	40.67
F1 Score	23.47

Table 4: Percentage increase in performance

# Conclusion

- Based on a quantitative and qualitative analysis of current processes to create hybrid chatbots, it can be concluded that such chatbots can be constructed using technologies such as MLFF and Transformer networks. The results indicate that attention-based implementations are much better suited for synthesizing such chatbots.
- Based on these conclusions, practitioners should consider using these models for websites or applications with significant traffic from people belonging to diverse cultures.
- The next iteration of this research could be focused on creating an intelligent system that identifies the users' preferred language and automatically loads up a chatbot of that language.



# References

- [1] Simran Khanuja et al. “Muril: Multilingual representations for indian languages”. In: arXiv preprint arXiv:2103.10730 (2021).
- [2] Usha Kiruthika et al. “E-agriculture for direct marketing of food crops using chatbots”. In: 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS). IEEE. 2020, pp. 1–4.
- [3] Gustavo Marques Mostaco et al. “AgronomoBot: a smart answering Chatbot applied to agricultural sensor networks”. In: 14th international conference on precision agriculture. Vol. 24. 2018, pp. 1–13.
- [4] S Gracia et al. “Connect Farmer”. In: International Journal of Engineering and Technology(UAE) 7 (Sept. 2018), pp. 181–183. doi: 10.14419/ijet.v7i3.34.18942.
- [5] J Vijayalakshmi and K Pandimeena. “Agriculture talk bot using AI”. In: Int. J. Recent Technol. Eng 8 (2019), pp. 186–190.
- [6] Brahmananda Reddy D. “Agriculture Chatbot Application Using Python”. In: International Journal of Interdisciplinary Innovative Research and Development (2020).
- [7] P Kaviya et al. “Artificial Intelligence Based Farmer Assistant Chatbot”. In: International Journal of Research in Engineering, Science and Management 4.4 (2021), pp. 26–29.
- [8] DK Yashaswini, R Hemalatha, and G Niveditha. “Smart chatbot for agriculture”. In: Int J Eng Sci Comput 9 (2019), pp. 22203–22205.
- [9] Jayalath Ekanayake and Luckshitha Saputhanthri. “E-AGRO: Intelligent Chat-Bot. IoT and Artificial Intelligence to Enhance Farming Industry”. In: AGRIS online Papers in Economics and Informatics 12.665-2020-1221 (2020), pp. 15–21.
- [10] Naman Jain et al. “AgriBot: agriculture-specific question answer system”. In: (2019).
- [11] H Manoj T Gadiyar Dr Thyagaraju and GS Poornima. “AI AND CLOUD-BASED SMART FARM ASSIST IN KANNADA FOR COCONUT FARM-ERS”. In: ().
- [12] Duygu Altinok. Mastering spaCy: An end-to-end practical guide to implementing NLP applications using the Python ecosystem. Packt PublishingLtd, 2021.