CHATBOT

HOSPITAL CUSTOMER CARE ASSISTANT

Alaaeddin Osta

TABLE OF CONTENTS

I. INTRODUCTION

PROJECT OVERVIEW

The project aims to build a hospital chatbot to assist the customer care department.

OBJECTIVES

- Clean and preprocess data
- Build machine learning models to categorize queries into different tags
- Evaluate the trained models
- Select the best model and deploy it into a chatbot

DATA DESCRIPTION

The data consists of a few samples of queries and responses collected in JSON format.
It has four total keys:

- **Tag**: Represents the category to which the query and response belong, such as greeting or goodbye. (This is the target variable that the model will predict)

- **Patterns**: Contains lists of user queries. (This feature serves as the input feature to the model)

- **Responses**: Contains lists of responses for corresponding queries. (This data is not used for training it is only for chatbot deployment)

- **Context**: Provides contextual information of the query and provides association to other tags when invoked. (Same as the Responses key, this data is utilized post-training)

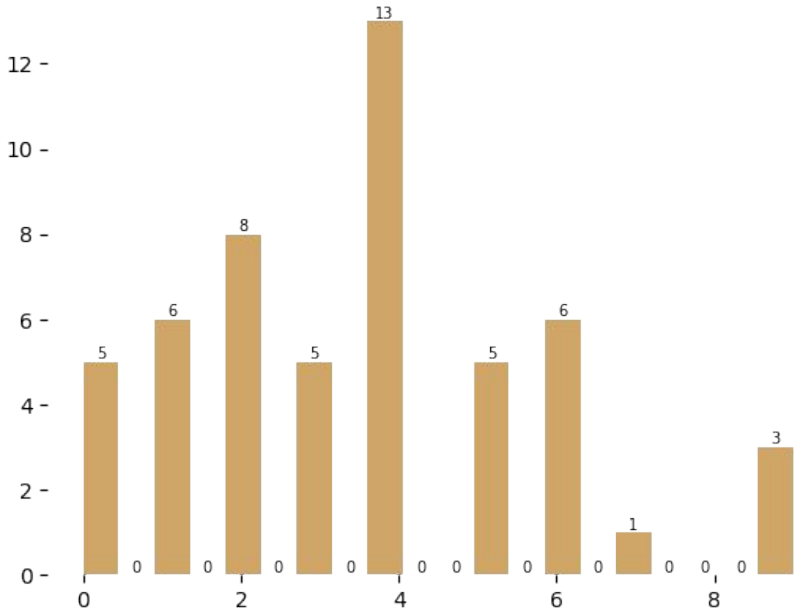
Sample from the dataset


```
{'tag': 'greeting', 'patterns': ['Hi there', 'How are you', 'Is anyone there?', 'Hey', 'Hola', 'Hello', 'Good day'], 'responses': ['Hello, thanks for asking', 'Good to see you again', 'Hi there, how can I help?'], 'context': ['']}...
```

II. DATA ANALYSIS

PATTERNS COLUMN (INPUT FEATURE)

Distribution of Number of Words in the Sentences of Patterns Column

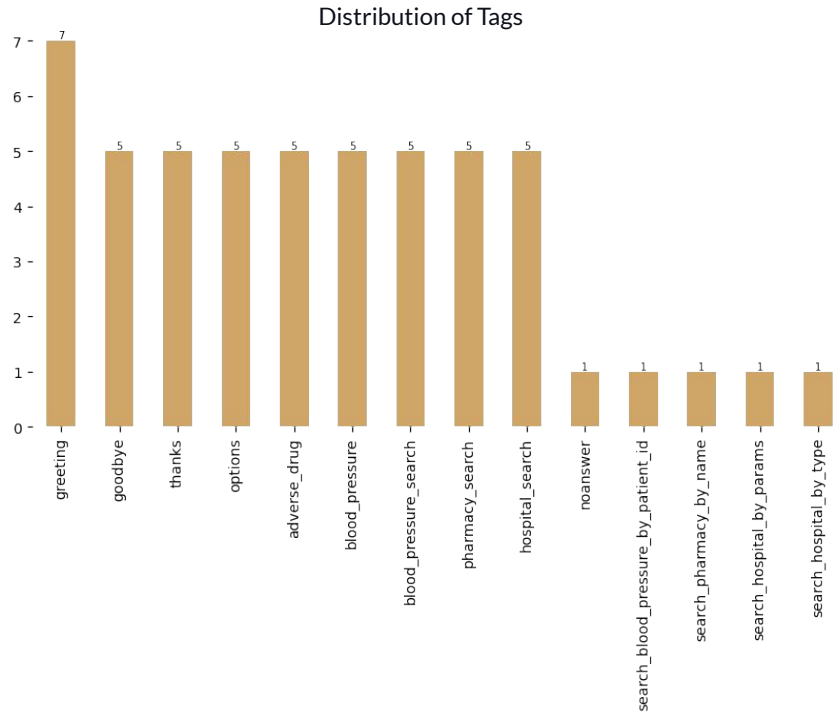


The dataset shows a range of sentence lengths within the patterns feature, ranging from 0 to 9 tokens.

The longest sentences, containing 9 tokens, occur 3 times.

Sentences with 4 tokens are the most frequent in the dataset with 13 occurrences.

TAG COLUMN (TARGET VARIABLE)



There are a total of 14 tags in the dataset. The "Greeting" tag has the highest count, occurring 7 times.

Most other tags have a relatively high count of 5.

The "Noanswer" tag, along with tags starting with the word "search", appears only once.

Note: Tags with only one occurrence were excluded from the training data. Further details will be provided in subsequent slides.

III. DATA PREPROCESSING

RESTRUCTURING THE DATA

The array of dictionaries was turned into a pandas dataframe as follows:

	tag	patterns	responses	context
0	greeting	[Hi there, How are you, Is anyone there?, Hey,...	[Hello, thanks for asking, Good to see you aga...	[]
1	goodbye	[Bye, See you later, Goodbye, Nice chatting to...	[See you!, Have a nice day, Bye! Come back aga...	[]
2	thanks	[Thanks, Thank you, That's helpful, Awesome, t...	[Happy to help!, Any time!, My pleasure]	[]

The elements of the pattern column were transformed from list-like elements to rows, replicating index values.

tag	patterns
greeting	[Hi there, How are you, Is anyone there?, Hey,...

tag	patterns
greeting	Hi there
greeting	How are you
greeting	Is anyone there?
greeting	Hey

DEALING WITH NULL VALUES

All the tags that start with the word "search" don't have pattern input. Instead, they are referenced in the context of other tags. So, rows with these tags were excluded from the training data and only employed to generate responses.

tag	patterns	responses	context
search_blood_pressure_by_patient_id	NaN	[Loading Blood pressure result for Patient]	[]
search_pharmacy_by_name	NaN	[Loading pharmacy details]	[]
search_hospital_by_params	NaN	[Please provide hospital type]	[search_hospital_by_type]
search_hospital_by_type	NaN	[Loading hospital details]	[]

The "noanswer" tag means that the user has not inputted anything and has an empty pattern input. Thus, it does not require prediction. An if statement is utilized instead: if the user input is an empty string, the response corresponding to the "noanswer" tag is triggered.

noanswer	NaN	[Sorry, can't understand you, Please give me m...	[]

DEFINING X AND Y: HANDLING LIMITED DATA SIZE WITH CROSS-VALIDATION

The dataset consists of only 47 samples, which is too small. Due to this limited size, there isn't enough data to split into separate training and test sets. Thus, the 47 samples were utilized for training, and StratifiedKFold cross-validation was employed to evaluate the models.

After removing empty values, the 'patterns' column was assigned to X_train, while nine unique tag labels were assigned to y_train.

LABEL ENCODING TAGS

The tag labels were encoded into numerical representations (from 0 to 8).

I initially intended to use one-hot encoding for the labels. However, the StratifiedKFold split function was not accepting one-hot encoded labels, so I stuck with the encoded numerical representations instead.

TEXT VECTORIZING AND EMBEDDING

To convert the training sentences from the patterns column into numbers for the machine learning models, three different techniques were tested: TF-IDF vectorizer, token embedding, and sentence embedding employing various models for evaluation. Both pipelines incorporate stop word removal and lowercase conversion as preprocessing steps.

IV. PREDICTIVE MODELING

MODEL EVALUATION

Model	Training Accuracy	Cross Validation Accuracy	Training Loss	Cross Validation Loss
MultinomialNB with Tfidf	95.74	74.00	1.3453	1.6931
DNN with Tfidf	95.74	70.22	0.1406	1.0337
DNN with token embedding	100	44.67	0.0084	1.8589
Conv1D with token embedding	100	70.22	0.1302	0.9039
Bi-LSTM	100	74.22	0.0082	1.1238
Universal Sentence Encoding	100	93.33	0.0796	0.4012
DistilBERT	100	54.44	0.1011	2.2962

All the DNN models were trained with 50 epochs and a batch size of 12, with Sparse categorical cross-entropy as the loss function. The MultinomialNB model was utilized as a baseline and showed a relatively good performance on the cross-validation set. StratifiedKFold with k = 5 was used for cross-validation.

Universal Sentence Encoding and Bi-LSTM outperform other models like DNN with Tfidf and DNN with token embedding in accuracy and loss metrics on the cross-validation set, where the model with Universal Sentence Encoding took the lead.

Interestingly, the DestilBERT transformer model did not perform well on the cross-validation set.

As a result, the Universal Sentence Encoding model is generalizing the best, so it was selected for deployment into the chatbot.

As anticipated, the models are overfitting due to the data size. To address the issue, further experimentation with hyperparameter tuning and regularization techniques could be conducted. Additionally, adding more data may enhance the model's ability to generalize and yield better results.

V. CHATBOT DEPLOYMENT

DUMMY DATA

Dummy data for blood pressure, hospitals, and pharmacies was generated to test the chatbot.

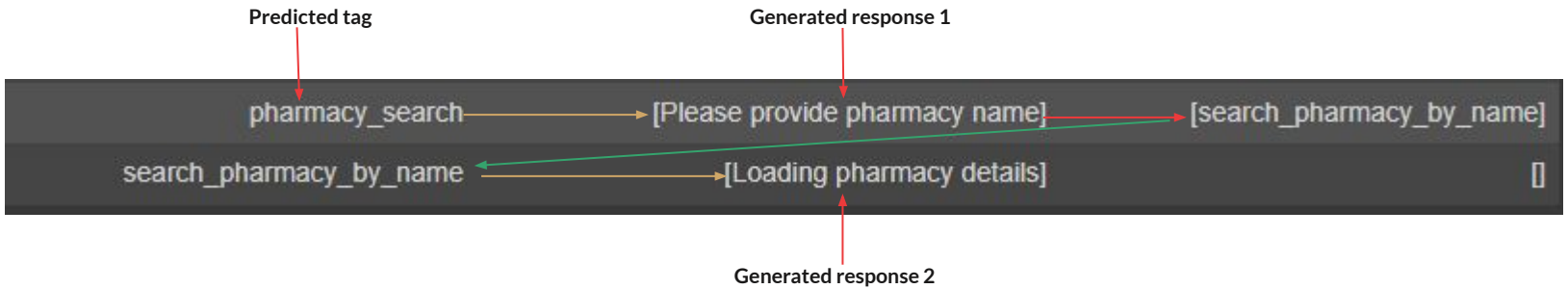
```
# Creating dummy data
blood_pressure_data = {
    'Patient ID': [1, 2, 3, 4, 5],
    'Systolic Pressure (mmHg)': [120, 130, 125, 140, 118],
    'Diastolic Pressure (mmHg)': [80, 85, 82, 90, 78],
    'Heart Rate (bpm)': [72, 75, 70, 80, 68],
    'Date': ['2024-03-29', '2024-03-29', '2024-03-29', '2024-03-29', '2024-03-29']
}
```

```
pharmacy_data = {
    'Pharmacy ID': [1, 2, 3, 4, 5],
    'Pharmacy Name': ['ABC Pharmacy', 'XYZ Pharmacy', 'Sunshine Pharmacy', 'City Pharmacy', 'Ocean Pharmacy'],
    'Address': ['123 Main Street', '456 Elm Street', '789 Oak Avenue', '101 Pine Street', '202 Cedar Street'],
    'City': ['Anytown', 'Anycity', 'Somewhere', 'Anyville', 'Anybeach'],
    'State': ['NY', 'CA', 'TX', 'FL', 'FL'],
    'Zip Code': ['12345', '98765', '54321', '67890', '09876'],
    'Phone': ['(555) 123-4567', '(555) 987-6543', '(555) 321-6789', '(555) 678-1234', '(555) 890-5678']
}
```

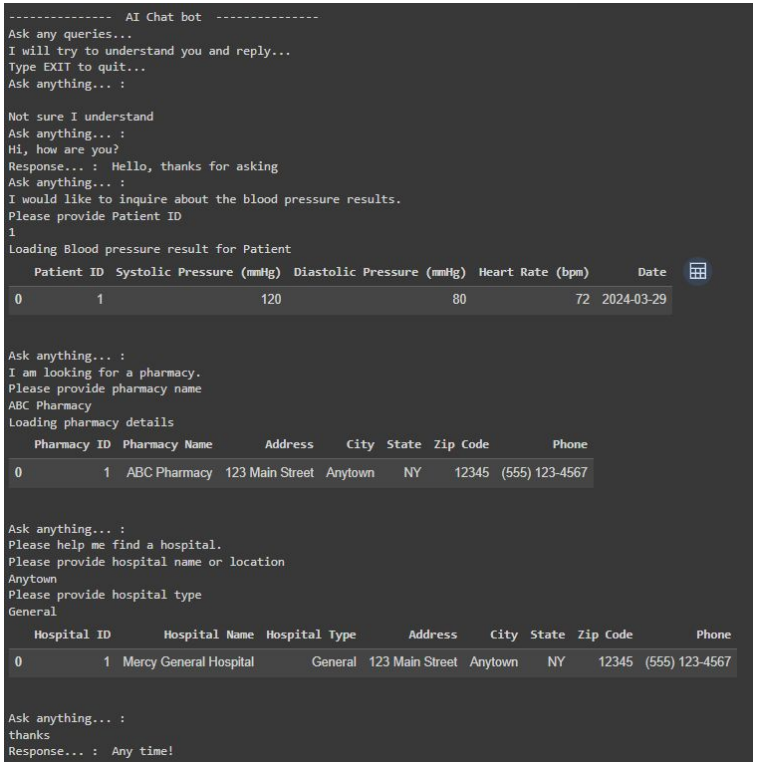
```
hospital_data = {
    'Hospital ID': [1, 2, 3, 4, 5],
    'Hospital Name': ['Mercy General Hospital', 'St. Mary's Hospital', 'Sunshine Hospital', 'City Hospital', 'Ocean View Hospital'],
    'Hospital Type': ['General', 'General', 'Community', 'General', 'Specialty'],
    'Address': ['123 Main Street', '456 Elm Street', '789 Oak Avenue', '101 Pine Street', '202 Cedar Street'],
    'City': ['Anytown', 'Anycity', 'Somewhere', 'Anyville', 'Anybeach'],
    'State': ['NY', 'CA', 'TX', 'FL', 'FL'],
    'Zip Code': ['12345', '98765', '54321', '67890', '09876'],
    'Phone': ['(555) 123-4567', '(555) 987-6543', '(555) 321-6789', '(555) 678-1234', '(555) 890-5678']
}
```

RESPONSE GENERATION

The predicted tag is used to generate a response from the responses column, while the context column is used to call other responses.



EXAMPLE OUTPUT



```
--------------  AI Chat bot  --------------
Ask any queries...
I will try to understand you and reply...
Type EXIT to quit...
Ask anything... :

Not sure I understand
Ask anything... :
Hi, how are you?
Response... :  Hello, thanks for asking
Ask anything... :
I would like to inquire about the blood pressure results.
Please provide Patient ID
1
Loading Blood pressure result for Patient
```

	Patient ID	Systolic Pressure (mmHg)	Diastolic Pressure (mmHg)	Heart Rate (bpm)	Date
0	1	120	80	72	2024-03-29

```
Ask anything... :
I am looking for a pharmacy.
Please provide pharmacy name
ABC Pharmacy
Loading pharmacy details
```

	Pharmacy ID	Pharmacy Name	Address	City	State	Zip Code	Phone
0	1	ABC Pharmacy	123 Main Street	Anytown	NY	12345	(555) 123-4567

```
Ask anything... :
Please help me find a hospital.
Please provide hospital name or location
Anytown
Please provide hospital type
General
```

	Hospital ID	Hospital Name	Hospital Type	Address	City	State	Zip Code	Phone
0	1	Mercy General Hospital	General	123 Main Street	Anytown	NY	12345	(555) 123-4567

```
Ask anything... :
thanks
Response... :  Any time!
```

We can see that the chatbot is functioning as intended, providing meaningful responses and effectively searching for the requested details.

VI. WRAP UP

CONCLUSION

- In this project, we successfully built a chatbot for hospital customer care assistance. The process included data preprocessing, model building, evaluation, and deployment.
- The data underwent various preprocessing steps, including data restructuring, NaN removal, label encoding, and text vectorization with token & sentence embeddings.
- StratifiedKFold cross-validation was employed for model evaluation due to the limited dataset size for better model evaluation despite the small sample size.
- Multiple machine learning models were trained and evaluated. The Universal Sentence Encoding model outperformed other models in accuracy and loss metrics in both the training and cross-validation sets.
- The Universal Sentence Encoding model was selected for deployment into the chatbot, ensuring optimal performance and functionality in assisting hospital customers.
- The chatbot effectively generated responses using predicted tags and context associations, providing meaningful assistance to users.

Thank You!

Credits: This presentation template was created by Slidesgo and includes infographics and images from Freepik.