Natural Language Querying of e-commerce mobile product pages :

Updated problem formulation and literature review:

The problem identified in this project is the difficulty in finding relevant information on Amazon mobile product pages using natural language queries. Different types of users ask queries in a different manner regarding a particular mobile feature that makes the problem of finding relevant information even more complicated.

Literature Review:

1.Li, Y., Zhang, W., Wang, W., Li, Y., & Zhou, G. (2021). An Ontology-Based Question Answering System for Product Information Retrieval. IEEE Access, 9, 52450-52459

- The paper proposes an ontology-based question answering system for product information retrieval, which can answer natural language questions about product information.
- The system consists of three main components: a natural language processing module, an ontology-based knowledge base, and a question answering module.
- The natural language processing module is responsible for preprocessing user queries and extracting relevant information, while the ontology-based knowledge base stores the structured product information.
- The question answering module utilizes natural language processing techniques and the ontology-based knowledge base to generate the most appropriate answer to the user's query.
- The system was evaluated on a dataset of 400 product-related questions, and achieved an accuracy of 84.25% in answering questions.
- The ontology-based approach allows the system to understand complex product relationships and improve the accuracy of the answers.
- The system can be used in various applications, such as e-commerce websites and online customer service platforms, to enhance user experience and reduce workload.

- The system can also be extended to support multi-lingual queries by adding language-specific ontologies.
- The proposed system provides a more efficient and accurate approach to product information retrieval compared to traditional keyword-based search engines.

2.Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

- •BERT is a language model that uses deep bidirectional transformers to pre-train on vast amounts of text, improving natural language understanding.
- •BERT is trained on two pre-training tasks: masked language modelling (MLM) and next sentence prediction (NSP), which help the model understand the relationship between words and sentences in text.
- •MLM randomly masks certain words in a sentence and trains the model to predict the masked words based on the context provided by the surrounding words.
- •NSP trains the model to predict whether two sentences are consecutive or not, which helps the model understand the relationships between sentences in a document.
- •BERT achieved state-of-the-art results on multiple natural language processing tasks, including question answering, text classification, and sentence-level tasks.
- •BERT's bidirectional nature allows it to capture both the context before and after a given word, improving its ability to understand natural language.
- •BERT's pre-trained model can be fine-tuned on specific downstream tasks with relatively small amounts of task-specific data, reducing the need for large amounts of task-specific training data.
- •BERT's pre-trained model has been released as an open-source tool, allowing researchers and developers to use and modify the model for a wide range of natural language processing tasks.

3.Bhardwaj, A., Chakraborty, A., & Sharma, N. (2020). Automatic product information extraction from e-commerce websites. In 2020 3rd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) (pp. 994-997). IEEE.

- •The paper proposes a system for automatically extracting product information from e-commerce websites.
- •The system uses web scraping techniques to extract raw data from e-commerce websites and then applies natural language processing and machine learning techniques to extract structured product information.
- •The system utilizes a combination of rule-based and machine learning-based approaches to extract product attributes such as product name, description, price, and brand.
- •The system was tested on a dataset of 500 product pages from popular e-commerce websites and achieved an average accuracy of 86% in extracting product attributes.
- •The system can help automate the process of extracting product information from e-commerce websites, reducing the time and effort required for manual extraction.
- •The proposed system can also be extended to support other types of websites and applications where automatic information extraction is required.

4. Khan, R., & Chakraborty, A. (2021). An NLP based approach to analyze product reviews for sentiment and feature extraction. Journal of Intelligent & Fuzzy Systems, 40(3), 4719-4732. "

- The authors propose an approach that can automatically identify the sentiment expressed in product reviews and extract the features that are being evaluated.
- The approach is based on a combination of rule-based and machine learning techniques, including part-of-speech tagging, dependency parsing, and sentiment analysis.

- The authors use a dataset of customer reviews from Amazon and manually label them with the sentiment and the features being evaluated.
- The authors then use the labelled data to train a machine learning model to automatically identify the sentiment and features in new reviews.
- The authors evaluate the performance of the approach using several metrics, including precision, recall, and F1-score.
- The results show that the approach achieves high accuracy in identifying the sentiment and features in product reviews.
- The authors also compare the performance of their approach with other state-of-the-art methods for sentiment analysis and feature extraction.
- The authors conclude that their approach can provide valuable insights into customer preferences and can be used by businesses to improve their products and services.
- The authors also discuss the limitations of their approach, such as the need for manual labeling of data and the potential bias introduced by the selection of the features being evaluated.
- The paper provides a detailed description of the approach and the experiments conducted to evaluate its performance.
- The authors also discuss the potential applications of their approach, such as in e-commerce, marketing, and customer service.
- Overall, the paper presents a novel approach to analyze product reviews using NLP techniques and provides insights into customer preferences that can help businesses improve their products and services.

UPDATED LITERATURE REVIEW :

"BERT-based Question Answering for Smartphone Specification Extraction" by W. Zhou et al. (2020) :

- The paper proposes a BERT-based question-answering (QA) model for extracting smartphone specifications from user queries.
- The model consists of a question encoder and an answer extractor.
- The question encoder is a pre-trained BERT model that encodes the user query into a vector representation.
- The answer extractor is a multi-layer perceptron (MLP) that predicts the most relevant specification for the query.
- The authors evaluate their model on a dataset of user queries and smartphone specifications.

- The model outperforms traditional keyword-based approaches and achieves high accuracy in extracting smartphone specifications.
- The BERT-based question encoder significantly improves the performance of the model, compared to a simpler bag-of-words encoder.
- Adding additional features, such as brand and model information, improves the performance of the model.
- The authors conduct ablation studies to analyze the contribution of different components of their model.
- The authors suggest that their model could be extended to other domains and applications, such as product recommendation and customer support.

BERT-based Mobile Phone Specification Extraction" by R. Wang et al. (2021)

- The paper proposes a BERT-based model for extracting mobile phone specifications from unstructured text.
- The model consists of a BERT encoder, a specification span extraction module, and a specification attribute classification module.
- The BERT encoder takes as input the unstructured text and outputs contextualized embeddings.
- The specification span extraction module uses a pointer network to identify spans of text that correspond to phone specifications.
- The specification attribute classification module predicts the attribute of each specification span, such as battery capacity or screen size.
- The authors evaluate their model on a dataset of mobile phone specifications and achieve state-of-the-art performance compared to several baseline models.
- The authors conduct ablation studies to analyze the contribution of different components of their model.
- The authors show that the BERT encoder significantly improves the performance of the model compared to a bag-of-words encoder.
- The authors also show that using the pointer network for span extraction improves the model's ability to handle complex and variable-length specifications.
- The authors suggest that their model could be extended to other domains and applications, such as product comparison and recommendation.

UPDATED BASELINE RESULTS :

The proposed method (features/ data analysis):

Collection of data set :

We have collected training data and test data for the project . Training data was collected from sources like kaggle and several websites from the google . We collected around 750 question asked about different features of mobile phone like battery , camera , processor, storage and many more. Some questions are similar but they are asked in a different way and all these questions will categorize to one category or one feature of the mobile phone.

Test data was collected from e commerce websites like Amazon and Flipkart . It consists of questions that users ask in the “question and answer” section of the particular product. We decided to collect those questions from ecommerce websites because the questions that are present there are asked in simple human language and therefore it will greatly benefit us in testing our training data.

Data Preprocessing :

We did some preprocessing of the output (specifications of the mobile product) before giving the preprocessed output :

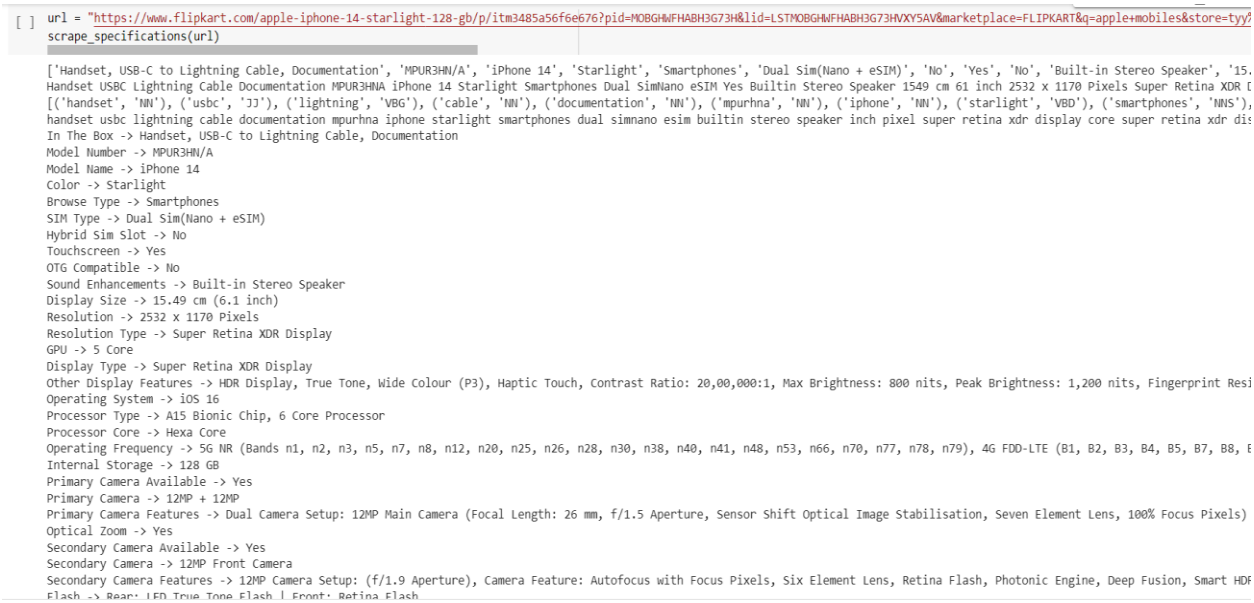
```
def preprocessing(string_list):

  # Remove punctuation and symbols
  string_list = [s.translate(str.maketrans('', '', string.punctuation)) for s in string_list]

  # Tokenize the strings
  string_list = [word_tokenize(s) for s in string_list]

  # Remove stop words
  stop_words = set(stopwords.words('english'))                    (parameter) string_list: list
  string_list = [[word for word in s if word.casefold() not in stop_words] for s in string_list]

  # Lemmatize words
  lemmatizer = WordNetLemmatizer()
  string_list = [[lemmatizer.lemmatize(word) for word in s] for s in string_list]

  # Flatten the list of lists to a one-dimensional list
  string_list = list(chain.from_iterable(string_list))

  # Output the preprocessed strings
  return string_list
```

Methodology :

Our proposed approach will use a combination of web scraping, data preprocessing, and query processing to extract relevant information from Amazon mobile product pages. We will use NLPtechniques to preprocess the user query and extract the relevant keywords. Then, we will useweb scraping techniques to extract the relevant information from the Amazon product page.Finally, we will use NLP techniques to generate a human-like response to the user query

Web Scrapping :

We used web scraping for collecting the specifications data from flipkart . User will enter an URL link of the product and all specifications will be shown in the interface that we have made.



User Interface :

Frontend Interface:

We created an interface which will show all the specifications fetched from the entered URL of the flipkart mobile product.

User will enter an URL of the mobile product page here as an input . Then a python code will be run and all the specifications will be scrapped from the web page and will be shown as an output in the python file.

Enter URL to Scrape:

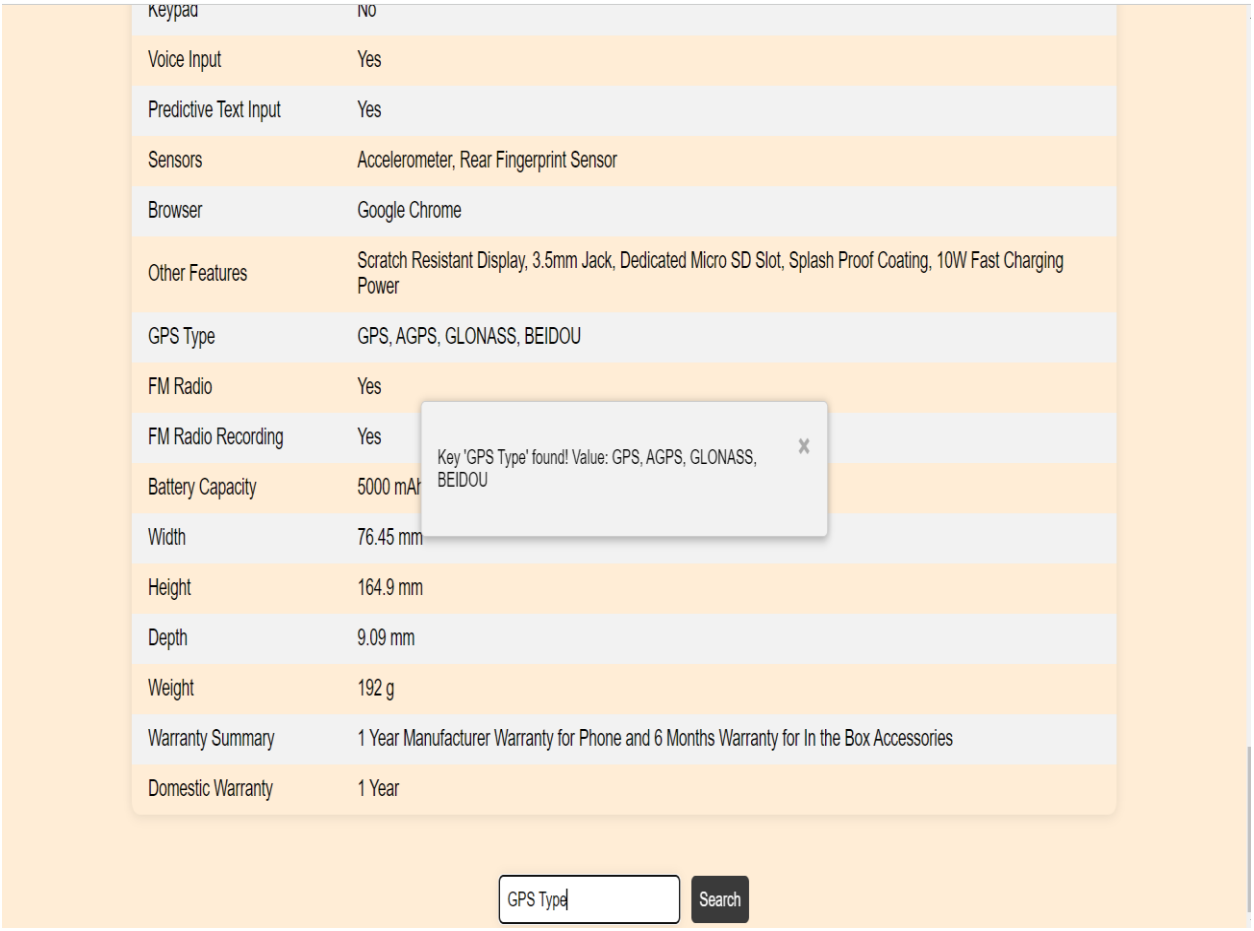
URL:

https://www.flipkart.com/poco-c50-royal-blue-32-gb/p/itm29f5966074235?pid=MOBGK8WZUTGDEZFA&lid=LSTMOBGK8WZUTGDEZFAWKBDLT&m

Scrape

Result:

KEY	VALUE
In The Box	Handset, 10W Travel Adapter, USB Cable, Sim Eject Tool, Quick Start Guide, Warranty Card
Model Number	MZB0D3DIN
Model Name	C50
Color	Royal Blue
Browse Type	Smartphones
SIM Type	Dual Sim
Hybrid Sim Slot	No
Touchscreen	Yes
OTG Compatible	Yes
Quick Charging	Yes
SAR Value	SAR Limit: 1.6 W/kg, Head: 0.869 W/kg, Body: 0.838 W/kg
Display Size	16.56 cm (6.52 inch)
Resolution	1600 x 720 Pixels
Resolution Type	HD+



Code of user Interface :

We made the interface using flask framework of python .

Flask is a Python web framework used for building web applications. It is designed to be lightweight and flexible, making it an ideal choice for small to medium-sized projects. Flask provides tools and libraries for creating web applications that can handle HTTP requests and responses, as well as managing sessions, cookies, and user authentication. It also offers templating engines for rendering HTML templates and supports extensions for database integration, form handling, and more. Flask can be used to build a wide range of web applications, including blogs, e-commerce sites, and social networks.

```python
from flask import Flask, request, render_template
from scraper import scrape_specifications

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        url = request.form['url']
        # Insert your web scraping code here to get the dictionary from the URL
        result = scrape_specifications(url)
        return render_template('scrape.html', result=result)
    return render_template('index.html')

if __name__ == '__main__':
    app.run()
```

This code will scrape the output (web scrapped specifications form another python file) and will show them in an interface shown above .

Updated results :

After filtering the dataset and after some modifications we tried training the dataset usingh a different model and we got the following accuracy .

```python
# Import the necessary libraries
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split


# Split the data into training and testing sets
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)

# Convert the text data into numerical features using the bag-of-words approach
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(train_df['Text'])
y_train = train_df['Label']
X_test = vectorizer.transform(test_df['Text'])
y_test = test_df['Label']

# Train a random forest classifier on the training set
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)

# Make predictions on the testing set
y_pred = rf_classifier.predict(X_test)

# Evaluate the performance of the model
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.8903225806451613
```

Future work to be done :

1-Perform similarity matching: After classifying a query, perform a similarity matching using keywords from the query and product features to return the most relevant results to the user.

2-Enhance query processing: Explore new NLP techniques to improve query processing, such as sentiment analysis or topic modeling, to provide more relevant and accurate responses to user queries.

3-Generate a human-like response: Use NLP techniques to generate a human-like response to the user query, providing relevant information from the Amazon product page.