SATHYABAMA INSTITUTE OF SCIENCE & TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCSA 2604 NATURAL LANGUAGE PROCESSING LAB

LAB 4: CASE STUDY

AIM: Enhancing Customer Service with Semantic Analysis

Problem Statement:

A multinational e-commerce company, "E-Shop Inc.," is looking to improve its customer service operations by leveraging advanced natural language processing techniques. They have a vast repository of customer interactions, including emails, chat transcripts, and social media messages. E-Shop Inc. aims to implement semantic analysis to better understand customer queries and sentiment, ultimately enhancing the overall customer experience..

Objectives :

- Semantic Analysis: The primary objective is to perform semantic analysis on customer queries to understand the underlying meaning and extract relevant information. By identifying synonyms and related terms, the program aims to capture the semantic nuances of the input text.
- Improving Customer Service: The program aims to enhance customer service operations by providing insights into customer queries. By analyzing the semantics of the queries, the program can help identify common issues, extract key information, and facilitate more effective responses.

Dataset:

In the provided program, there isn't a specific dataset used for semantic analysis. Instead, the program demonstrates a basic approach to perform semantic analysis on a set of example customer queries. However, in a real-world scenario, the dataset used for semantic analysis could consist of a collection of text data relevant to the domain of interest, such as customer support tickets, product reviews, social media interactions, or any other type of textual data where semantic analysis is applicable.

Approach:

- **Tokenization**: The program starts by tokenizing the input text into individual words or tokens. Tokenization is a fundamental step in natural language processing (NLP) for breaking down text into its constituent parts.
- **Stopword Removal**: Stopwords, such as "is", "the", "and", etc., are removed from the tokens to filter out irrelevant words that do not carry much semantic meaning.

- **Lemmatization**: The program lemmatizes the remaining tokens to reduce them to their base or dictionary form. Lemmatization helps in normalizing words and reducing inflectional forms to a common base, improving the accuracy of semantic analysis.
- **Synonym Generation**: Using the WordNet database, the program retrieves synonyms for each lemmatized token. WordNet is a lexical database of the English language that provides semantic relationships between words, including synonyms, hypernyms, hyponyms, etc.
- **Output Generation**: Finally, the program outputs the synonyms generated for each customer query, providing insights into the semantic content of the queries.

Program :

```
import nltk
from nltk.corpus import wordnet
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

# Initialize NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Function to perform semantic analysis
def semantic_analysis(text):
    # Tokenize text
    tokens = word_tokenize(text)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [word for word in tokens if word.lower() not in
stop_words]

    # Lemmatization
    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in
filtered_tokens]

    # Synonyms generation
    synonyms = set()
    for token in lemmatized_tokens:
        for syn in wordnet.synsets(token):
            for lemma in syn.lemmas():
                synonyms.add(lemma.name())

    return list(synonyms)

# Example customer queries
customer_queries = [
```

```
    "I received a damaged product. Can I get a refund?",
    "I'm having trouble accessing my account.",
    "How can I track my order status?",
    "The item I received doesn't match the description.",
    "Is there a discount available for bulk orders?"
]

# Semantic analysis for each query
for query in customer_queries:
    print("Customer Query:", query)
    synonyms = semantic_analysis(query)
    print("Semantic Analysis (Synonyms):", synonyms)
    print("\n")
```

Output:

[nltk_data] Downloading package punkt to /root/nltk_data...

[nltk_data] Unzipping tokenizers/punkt.zip.

[nltk_data] Downloading package stopwords to /root/nltk_data...

[nltk_data] Unzipping corpora/stopwords.zip.

[nltk_data] Downloading package wordnet to /root/nltk_data...

Customer Query: I received a damaged product. Can I get a refund?

Semantic Analysis (Synonyms): ['refund', 'grow', 'baffle', 'pay_off', 'Cartesian_product', 'arrive', 'engender', 'standard', 'have', 'damaged', 'experience', 'develop', 'sustain', 'product', 'acquire', 'encounter', 'take_in', 'find', 'stupefy', 'bugger_off', 'draw', 'pose', 'aim', 'nonplus', 'induce', 'mother', 'stimulate', 'make', 'repayment', 'convey', 'cause', 'mathematical_product', 'get', 'damage', 'produce', 'set_out', 'merchandise', 'buzz_off', 'beat', 'meet', 'start', 'commence', 'return', 'pick_up', 'production', 'fix', 'stick', "get_under_one's_skin", 'go', 'mystify', 'take', 'perplex', 'welcome', 'vex', 'begin', 'come', 'fuck_off', 'bring', 'contract', 'capture', 'generate', 'give_back', 'incur', 'repay', 'let', 'become', 'start_out', 'gravel', 'scram', 'obtain', 'pay_back', 'amaze', 'catch', 'beget', 'get_down', 'set_about', 'invite', 'bring_forth', 'drive', 'sire', 'intersection', 'discredited', 'suffer', 'received', 'ware', 'dumbfound', 'fetch', 'father', 'arrest', 'flummox', 'puzzle', 'bewilder', 'receive']

Customer Query: I'm having trouble accessing my account.

Semantic Analysis (Synonyms): ['write_up', 'report', 'invoice', 'trouble', 'describe', 'answer_for', 'access', 'pain', 'get_at', 'distract', 'disorder', 'unhinge', 'account_statement', 'calculate', 'disoblige', 'fuss', 'bill', 'disquiet', 'inconvenience', 'incommode', 'news_report', 'disturb', 'explanation', 'problem', 'perturb', 'cark', 'account', 'accounting', 'business_relationship', 'score', 'bother', 'history', 'story', 'difficulty', 'worry', 'inconvenience_oneself', 'hassle', 'chronicle', 'discommode', 'ail', 'put_out', 'upset', 'trouble_oneself']

Customer Query: How can I track my order status?

Semantic Analysis (Synonyms): ['cover', 'rank', 'cart_track', 'purchase_order', 'pass_over', 'parliamentary_law', 'cartroad', 'status', 'get_across', 'traverse', 'prescribe', 'order', 'orderliness', 'rails', 'fiat', 'gild', 'regularise', 'runway', 'govern', 'ordination', 'give_chase', 'put', 'cut_across', 'chase', 'consecrate', 'cut', 'grade', 'social_club', 'order_of_magnitude', 'path', 'rules_of_order', 'caterpillar_tread', 'tail', 'Holy_Order', 'cross', 'data_track', 'monastic_order', 'rate', 'go_after', 'say', 'edict', 'regularize', 'Order', 'caterpillar_track', 'parliamentary_procedure', 'cut_through', 'rail', 'enjoin', 'course', 'racecourse', 'arrange', 'club', 'society', 'ordinate', 'set_up', 'rescript', 'chase_after', 'place', 'dictate', 'tell', 'range', 'decree', 'regulate', 'lodge', 'condition', 'track', 'raceway', 'ordain', 'racetrack', 'get_over', 'lead', 'guild', 'running', 'tag', 'ordering', 'position', 'trail', 'dog']

Customer Query: The item I received doesn't match the description.

Semantic Analysis (Synonyms): ['point', 'mate', 'catch', 'equalize', 'detail', 'touch', 'invite', 'welcome', 'get', 'standard', 'jibe', 'description', 'verbal_description', 'gibe', 'have', 'rival', 'equal', 'pit', 'experience', 'fit', 'check', 'item', 'mates', 'correspond', 'oppose', 'pair', 'lucifer', 'received', 'peer', 'cope_with', 'encounter', 'play_off', 'meet', 'take_in', 'friction_match', 'find', 'particular', 'equate', 'match', 'couple', 'equalise', 'pick_up', 'agree', 'incur', 'compeer', 'twin', 'tally', 'obtain', 'token', 'receive']

Customer Query: Is there a discount available for bulk orders?

Semantic Analysis (Synonyms): ['tell', 'put', 'rank', 'parliamentary_procedure', 'purchase_order', 'consecrate', 'majority', 'range', 'useable', 'decree', 'parliamentary_law', 'regulate', 'price_reduction', 'mass', 'dictate', 'social_club', 'lodge', 'usable', 'enjoin', 'discount', 'grade', 'order_of_magnitude', 'brush_off', 'rules_of_order', 'rebate', 'push_aside', 'prescribe', 'arrange', 'bulge', 'bank_discount', 'dismiss', 'club', 'ordain', 'order', 'society', 'bulk', 'ordinate', 'Holy_Order', 'brush_aside', 'orderliness', 'guild', 'set_up', 'fiat', 'gild', 'regularise', 'uncommitted', 'ordering', 'available', 'monastic_order', 'deduction', 'ordination', 'govern', 'rate', 'discount_rate', 'say', 'edict', 'regularize', 'rescript', 'disregard', 'ignore', 'volume', 'Order', 'place']

Result:

By following this approach, the program aims to achieve the objectives of performing semantic analysis on customer queries and improving customer service operations by providing valuable insights into the semantics of the queries.