3-Data Cleaning Keyword Extraction

November 15, 2024

1 Data Cleaning and Keyword Extraction

This section covers data cleaning procedures, including removing noise from reviews, handling missing values, and extracting relevant keywords for sentiment analysis.

1.1 Goals

- Prepare the data for further analysis by cleaning and structuring it appropriately.
- Extract key themes from reviews using keyword extraction.

1.2 Steps

1. **Data Cleaning**: Remove irrelevant information (e.g., special characters, stop words) to make the text data suitable for analysis.
2. **Missing Value Handling**: Deal with missing data appropriately by either filling or removing incomplete entries.
3. **Keyword Extraction**: Extract relevant keywords for both food quality and service-related aspects to better understand customer focus.

[ ]:
```
import re
import nltk
from nltk.corpus import stopwords
from nltk import pos_tag, word_tokenize
import pandas as pd

# Download stopwords and POS tagger resources
#nltk.download('stopwords')
#nltk.download('averaged_perceptron_tagger')

filtered_reviews = pd.read_csv('filtered_exp_business_reviews.csv')
stop_words = set(stopwords.words('english'))

# Add custom stopwords
custom_stopwords = stop_words.union({'00', 'bit', 'came', 'extra', 'get',␣
 ↪'also', 'still', 'like', 'one', 'got',
                                     'aback', 'abandoned', 'aaa', 'abdominal',␣
 ↪'aaa beef'})
```

```
# Define a comprehensive set of keywords for targeted cleaning
important_keywords = {
    # Service-related terms
    'wait', 'waiter', 'waitress', 'server', 'staff', 'service', 'host',␣
 ↪'hostess', 'rude', 'friendly', 'slow', 'quick', 'attentive',
    'polite', 'unfriendly', 'helpful', 'professional', 'inattentive',␣
 ↪'delayed', 'efficient', 'courteous', 'hospitality', 'reservation',
    'seated', 'seat', 'line', 'queue', 'manager', 'bartender', 'customer',␣
 ↪'care', 'approachable', 'dining', 'experience', 'unprofessional',
    'prompt', 'long', 'courtesy', 'kind', 'accommodate', 'told', 'asked',␣
 ↪'approach', 'approached',

    # Food-related terms
    'food', 'dish', 'meal', 'cuisine', 'portion', 'taste', 'delicious',␣
 ↪'tasty', 'flavor', 'flavour', 'fresh', 'stale', 'spicy', 'mild',
    'bland', 'savory', 'savoury', 'hot', 'cold', 'crunchy', 'tender', 'crispy',␣
 ↪'greasy', 'dry', 'overcooked', 'undercooked', 'burnt',
    'cooked', 'prepared', 'presentation', 'plating', 'sauce', 'side', 'main',␣
 ↪'appetizer', 'entree', 'dessert', 'beverage', 'drink',
    'soup', 'salad', 'pasta', 'steak', 'burger', 'pizza', 'sushi', 'ramen',␣
 ↪'dimsum', 'noodle', 'pho', 'fried', 'grilled', 'baked', 'boiled',
    'marinated', 'succulent', 'juicy', 'organic', 'vegan', 'vegetarian',␣
 ↪'gluten', 'allergy', 'seasoned', 'spices', 'aroma', 'texture',
    'sweet', 'sour', 'salty', 'bitter', 'umami', 'rich', 'light', 'heavy',␣
 ↪'quality', 'quantity', 'authentic', 'fusion', 'unique', 'regular',
    'favorite', 'special', 'variety', 'option', 'choice', 'chef', 'cheese',␣
 ↪'bread', 'rice', 'pork', 'beef', 'chicken', 'seafood', 'shrimp',

    # Ambiance-related terms
    'ambiance', 'ambience', 'atmosphere', 'decor', 'lighting', 'music',␣
 ↪'noise', 'quiet', 'loud', 'spacious', 'crowded', 'cozy', 'comfortable',
    'elegant', 'modern', 'vintage', 'warm', 'welcoming', 'lively', 'vibe',␣
 ↪'energy', 'smell', 'clean', 'dirty', 'temperature', 'air',
    'conditioning', 'heating', 'intimate', 'family', 'friendly', 'romantic',␣
 ↪'rustic', 'luxurious', 'design', 'interior', 'outdoor',
    'patio', 'balcony', 'setting', 'environment', 'theme', 'aesthetic',

    # Cleanliness-related terms
    'clean', 'dirty', 'hygiene', 'sanitary', 'unsanitary', 'messy', 'tidy',␣
 ↪'restroom', 'bathroom', 'toilet', 'floor', 'table', 'sticky',
    'dust', 'trash', 'garbage', 'bins', 'organized', 'neat', 'spotless',␣
 ↪'filthy', 'stains', 'odor', 'smell', 'bugs', 'pests', 'rats',
    'cockroaches', 'soap', 'sanitizer', 'hand', 'wiped', 'cramped',␣
 ↪'disinfect', 'bathroom cleanliness', 'well-maintained', 'vacuum', 'waste',

    # Value-related terms
```

```
    'price', 'expensive', 'cheap', 'value', 'worth', 'cost', 'affordable',␣
 ↪'deal', 'reasonable', 'expensive', 'overpriced', 'underpriced',
    'service charge', 'hidden fees', 'special', 'discount', 'offer', 'wallet',␣
 ↪'friendly', 'budget', 'premium', 'luxury', 'fair', 'portion size',
    'quantity', 'expensive for what you get', 'worth it', 'not worth it',␣
 ↪'bargain', 'menu price', 'deal', 'reasonable price', 'competitive pricing',

    # General and Miscellaneous Terms
    'time', 'experience', 'good', 'bad', 'okay', 'excellent', 'amazing',␣
 ↪'poor', 'average', 'expectation', 'surprise', 'enjoyed', 'love', 'hate',
    'recommend', 'try', 'come', 'back', 'again', 'never', 'always',␣
 ↪'sometimes', 'often', 'first', 'last', 'disappointed', 'happy', 'satisfied',
    'unsatisfied', 'improvement', 'issue', 'problem', 'perfect', 'better',␣
 ↪'best', 'worst', 'suggest', 'avoid', 'visit', 'highly', 'wait', 'worth',
    'spend', 'money', 'return', 'favorite', 'miss', 'memorable', 'standard',␣
 ↪'normal', 'below', 'above', 'average', 'classic', 'signature',
    'popular', 'well', 'known', 'hidden', 'gem', 'new', 'old', 'location',␣
 ↪'area', 'city', 'local', 'nearby', 'close', 'far', 'walk', 'drive',
    'traffic', 'accessible', 'parking', 'lot', 'reservation', 'book',␣
 ↪'available', 'crowded', 'wait', 'queue', 'short', 'long', 'minutes', 'hours'
}

# Function to clean text with POS tagging and manually filter for important␣
 ↪words
def clean_text_with_targeted_keywords(text):
    # Convert to lowercase and remove non-alphabetic characters
    text = re.sub(r'[^a-zA-Z\s]', '', text.lower())
    # Tokenize and POS tagging
    tokens = word_tokenize(text)
    pos_tags = pos_tag(tokens)
    # Keep only relevant words based on POS tagging and important keywords
    content_words = [word for word, pos in pos_tags if word in␣
 ↪important_keywords and pos.startswith(('N', 'J', 'V'))]
    return ' '.join(content_words)

# Apply the cleaning function using the new broad set of keywords
filtered_reviews['all_text'] = filtered_reviews['text'].
 ↪apply(clean_text_with_targeted_keywords)

# Show a sample of cleaned text to verify
print("Sample of cleaned text after applying expanded keywords:")
print(filtered_reviews['all_text'].head())
```

[ ]:
```
filtered_reviews.to_csv('clean_filtered_reviews.csv', index = False)
```