CSE508 Winter 2024 A2 Report

Rahul Ajith
2021083

PART 1

Approach and Methodologies:

1. **ResNet50 for Feature Extraction**: Leveraging a pre-trained ResNet50 model from TensorFlow's Keras applications, which is widely recognized for its efficiency and accuracy in extracting features from images. ResNet50 is pre-trained on the ImageNet dataset, allowing it to recognize a wide range of visual patterns.
2. **Image Preprocessing**:
    - **Image Loading**: Images are loaded from URLs using the **skimage.io** module. This step is critical as it handles images directly from the web.
    - **Preprocessing Steps**: Images undergo several preprocessing steps to ensure they are in the correct format for the ResNet50 model:
        - **Resizing**: Each image is resized to 224x224 pixels, the input size expected by ResNet50.
        - **Contrast Enhancement**: Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to enhance image contrast, using the **exposure.equalize_adapthist** function from **skimage**.
        - **Format Adjustment**: Images are adjusted to have three channels (RGB) and are scaled back to the 0-255 range if necessary.
    - **Preparation for ResNet50**: Images are converted to arrays and preprocessed using **preprocess_input** from Keras, which scales the pixel values in a manner expected by the model.
3. **Feature Extraction**:
    - The preprocessed image is fed into the ResNet50 model to obtain deep features from the 'avg_pool' layer, providing a 2048-dimensional feature vector.
    - Feature vectors are normalized to unit length to standardize their scale, making them more comparable.
4. **Data Handling**:
    - **Dataset Loading**: The dataset containing image URLs is loaded using pandas, a versatile tool for data manipulation in Python.
    - **Feature Extraction Loop**: Each image URL in the dataset is processed individually to extract features, handling exceptions gracefully to ensure

the process continues even if some images cannot be processed.

Assumptions:

1. **Image URLs Accessibility**: It's assumed that all image URLs in the dataset are accessible and valid. Inaccessible or broken links are handled as exceptions, returning a zero vector for such cases.
2. **Single Image per Entry**: The script extracts features for the first image URL found in each dataset entry. If an entry contains multiple URLs, only the first is considered.
3. **Homogeneity of Data**: The approach assumes that the dataset is relatively homogeneous, meaning that the images are of a nature and quality that benefits from the chosen preprocessing and feature extraction methods.

Results:

```
1/1 [==============================] - 0s 171ms/step
1/1 [==============================] - 0s 171ms/step
1/1 [==============================] - 0s 172ms/step
1/1 [==============================] - 0s 175ms/step
1/1 [==============================] - 0s 181ms/step
1/1 [==============================] - 0s 173ms/step
1/1 [==============================] - 0s 168ms/step
1/1 [==============================] - 0s 164ms/step
1/1 [==============================] - 0s 166ms/step
1/1 [==============================] - 0s 174ms/step
1/1 [==============================] - 0s 168ms/step
1/1 [==============================] - 0s 179ms/step
1/1 [==============================] - 0s 175ms/step
1/1 [==============================] - 0s 173ms/step
1/1 [==============================] - 0s 173ms/step
1/1 [==============================] - 0s 167ms/step
1/1 [==============================] - 0s 175ms/step
1/1 [==============================] - 0s 170ms/step
1/1 [==============================] - 0s 182ms/step
1/1 [==============================] - 0s 173ms/step
1/1 [==============================] - 0s 171ms/step
1/1 [==============================] - 0s 176ms/step
1/1 [==============================] - 0s 176ms/step
1/1 [==============================] - 0s 174ms/step
1/1 [==============================] - 0s 166ms/step
1/1 [==============================] - 0s 172ms/step
1/1 [==============================] - 0s 169ms/step
1/1 [==============================] - 0s 165ms/step
Feature extraction completed and saved to /content/drive/My Drive/IR ASSIGNMENT 2/extracted_features.pkl.
```

The result of this process is a list of 2048-dimensional feature vectors, one for each image in the dataset. These features represent a deep, meaningful encoding of the visual content of the images, suitable for various machine learning and image retrieval tasks.

- **Output**: The extracted features are serialized using pickle and saved to a specified file path (**/content/drive/My Drive/IR ASSIGNMENT 2/extracted_features.pkl**). This file can be loaded in future sessions for analysis, modeling, or comparison tasks.

PART 2

Approach and Methodologies:

1. **Text Preprocessing**:
   - **Normalization**: Converts text to lowercase to ensure uniformity.
   - **Punctuation Removal**: Eliminates punctuation to focus on words.
   - **Tokenization**: Splits text into individual words (tokens) using NLTK's **word_tokenize**.
   - **Stopword Removal**: Filters out common words (stopwords) that are typically irrelevant to the analysis.
   - **Stemming and Lemmatization**: Reduces words to their base or root form, enhancing the uniformity of textual data.
2. **TF-IDF Computation**:
   - **Term Frequency (TF)**: Measures how frequently a term occurs in a document. For each document, a dictionary mapping terms to their term frequencies is created.
   - **Inverse Document Frequency (IDF)**: Computes the importance of a term across all documents. Terms that appear in many documents have a lower IDF score, indicating they are less distinctive.
   - **TF-IDF**: Combines TF and IDF, yielding a set of scores that reflect the importance of each term in each document relative to the entire corpus.
3. **Data Handling and Saving**:
   - **Dataset Loading**: Utilizes pandas for loading and manipulating the dataset.
   - **Feature Extraction Loop**: Applies text preprocessing, TF computation, and TF-IDF score calculation for each review.
   - **Result Storage**: Saves the TF-IDF scores using Python's pickle module for serialization, facilitating easy loading for future use.

Assumptions:

- **Data Quality**: Assumes that the review texts are sufficiently clean and relevant for NLP tasks. Irrelevant or nonsensical text data could skew TF-IDF scores.
- **Homogeneity of Reviews**: Assumes reviews are generally comparable in nature (e.g., all product reviews), making the computed TF-IDF scores meaningful across the dataset.
- **Stopwords and Stemming/Lemmatization Efficacy**: Assumes that the predefined list of stopwords and the stemming/lemmatization process are appropriate for the dataset's language and context.

Results:

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
TF-IDF scores saved to /content/drive/My Drive/IR ASSIGNMENT 2/tfidf_scores.pkl
```

The process yields a collection of TF-IDF scores for each document in the dataset, encapsulated in a list where each element corresponds to a dictionary mapping terms to their TF-IDF scores. These scores represent a numerical fingerprint of each review, highlighting the most distinctive terms in relation to the entire dataset.

- **Output**: The TF-IDF scores are saved to a specified file path (**/content/drive/My Drive/IR ASSIGNMENT 2/tfidf_scores.pkl**). This serialized file contains the computed scores, ready for loading and use in subsequent analytical or machine learning workflows.

PART 3

Approach and Methodologies:

1. **Text Processing with TF-IDF**:
    - **Objective**: Compute textual similarities between user-provided review text and the reviews in the dataset.
    - **Methodology**: Utilize the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique to transform review texts into numerical vectors. Then, compute cosine similarities between the user's review vector and each review vector in the dataset.
    - **Tools and Libraries**: **TfidfVectorizer** from **sklearn.feature_extraction.text** for vectorization and **cosine_similarity** from **sklearn.metrics.pairwise** to calculate similarities.
2. **Image Processing with Pre-trained Models**:
    - **Objective**: Determine visual similarities between a user-provided image and images in the dataset.
    - **Methodology**: (Pseudocode due to environment limitations) Suggest using a pre-trained deep learning model to extract feature vectors from images, including the user's image and dataset images. Subsequently, compute cosine similarities based on these feature vectors.
    - **Tools and Libraries**: TensorFlow (or PyTorch) along with models from

TensorFlow Hub (or similar repositories). A commonly used model for feature extraction is InceptionV3 or ResNet.

3. **Composite Score Calculation**:
    - **Objective**: Integrate image and text similarity scores for each item in the dataset into a single composite score.
    - **Methodology**: Average the computed image and text similarity scores for each dataset item to derive a composite similarity score, reflecting both visual and textual relevance to the user input.
4. **Ranking and Selection**:
    - **Objective**: Identify and display the top-ranking dataset items based on their composite similarity scores.
    - **Methodology**: Sort the dataset items by their composite scores in descending order. Select the top items for display, ensuring no multiple items from the same review are selected.

Assumptions:

- The dataset is structured with two columns: "Image" and "Review Text", without explicit IDs or URLs for images. The "Image" column content is assumed to be descriptive or identifier-like, aiding in fetching actual image data when needed.
- Image similarities can't be directly computed within this environment due to the lack of image processing capabilities and access to external image data. A pseudocode approach is provided instead, assuming the availability of necessary computational resources and libraries elsewhere.
- Cosine similarity scores, both for images and text, are pre-computed and stored in respective **.pkl** files. This setup bypasses the need for real-time computation of these scores within this limited environment.

Results:

- The approach successfully outlines a comprehensive methodology for integrating textual and visual similarities into a unified retrieval system.
- Textual similarity computation via TF-IDF and cosine similarity is straightforward and can be readily implemented with the provided tools and libraries.
- Image processing for similarity computations, though not directly executable here, is conceptualized using standard practices in machine learning and computer vision, offering a clear path for implementation in a suitable environment.
- The composite score calculation and ranking logic offer a holistic measure of relevance, considering both textual and visual aspects, thereby addressing complex retrieval scenarios that require multimodal similarity assessments.

This methodology underscores the importance of multimodal data processing in retrieval tasks, highlighting the synergy between NLP and computer vision techniques to enhance information retrieval systems.

PART 4

Approach and Methodologies:

1. **Loading Pre-Saved Similarity Data**:
   - The process begins by loading pre-saved similarity scores for both images and reviews from **.pkl** files. These scores are assumed to have been previously computed, reflecting the similarity of each item in the dataset to a specific query or set of queries.
2. **Composite Score Calculation**:
   - For each pair of image and review within the top-k elements, a composite score is calculated by averaging their respective similarity scores. This step merges the visual and textual similarity dimensions into a single metric, facilitating a unified ranking.
3. **Ranking and Displaying Results**:
   - The pairs are then ranked based on their composite scores, with higher scores indicating greater overall similarity to the input query(ies). The top-ranking pairs are displayed, showing the balance between image and text relevance.

Assumptions:

- **Pre-Computed Similarities**: It's assumed that similarity scores for both images and reviews are already available and meaningfully computed. The method of computation (e.g., feature extraction model for images, TF-IDF for text) and the baseline (query) for these scores are not specified.
- **Alignment of Image and Review Data**: The approach implies that each image-review pair (indexed similarly across the two lists) is relevant to be compared together, suggesting an assumption about their alignment or relatedness in the context of similarity scoring.
- **Top-K Limitation**: The focus on the top-k elements assumes that the most relevant comparisons for a given query are contained within the first few entries of the pre-saved data, which may not always capture the full spectrum of relevant items in the dataset.

Results:

```
Ranked Combined Retrieval Results:
Rank: 1, Image ID: ['https://images-na.ssl-images-amazon.com/images/I/81q5+IxFVUL._SY88.jpg'], Review ID: Loving these vintage springs on my vintage strat. They have a good tens
Rank: 2, Image ID: ['https://images-na.ssl-images-amazon.com/images/I/71edIGOwydL._SY88.jpg'], Review ID: Nice solid springs and defeinitely more silent. Easy installation and tl

Pictured with some old uninstalled springs next to them., Composite Score: 0.5370
Rank: 3, Image ID: ['https://images-na.ssl-images-amazon.com/images/I/71-PDjqCBcL._SY88.jpg', 'https://images-na.ssl-images-amazon.com/images/I/71bxnwWGCYL._SY88.jpg'], Review II
```

- **Unified Multimodal Ranking**: The methodology effectively integrates visual and textual similarities into a composite score, offering a multimodal ranking that considers both aspects equally. This provides a more nuanced and comprehensive assessment of similarity than considering either modality alone.

PART 5

Detailed Approach

1. **Loading Data**: The dataset and pre-saved similarity scores for images and reviews are loaded. These scores represent pre-computed similarities of dataset items to some reference queries or features, though the exact reference points (e.g., specific images or text queries) are not detailed in the code.
2. **User Input Handling**: The code accepts user inputs in the form of an image URL and a review text. These inputs are placeholders for the actual query features that would be compared against the dataset in a real application scenario.
3. **Finding Top Similar Items**: A function **find_top_similar_items** simulates the retrieval process by selecting the top 3 items based on their similarity scores. This step is abstracted for both images and reviews, with the sorting based on the pre-computed scores indicating presumed relevance.
4. **Result Display**:
   - For **image retrieval**, items are displayed with their URLs, corresponding reviews from the dataset, and similarity scores. The displayed composite score averages the image and the associated text similarity scores, aiming to provide a holistic view of item relevance.
   - For **text retrieval**, a similar format is followed, with the focus shifted to text-based relevance. Items are again ranked based on composite scores, highlighting the interplay between text and image similarities.

Assumptions

- **Pre-computed Similarities**: It's assumed that similarity scores are reflective of meaningful comparisons between dataset items and some reference queries or features, although the specifics of these computations are not described.
- **Data Structure and Integrity**: The methodology assumes a one-to-one correspondence between "Image" and "Review Text" in the dataset, allowing for direct lookups and associations between these elements based on similarity rankings.
- **Abstracted Similarity Computation**: The actual computation or criteria for determining similarity scores are abstracted away, focusing instead on how these pre-computed scores are utilized for retrieval and display.

Results Reflection

```
USING IMAGE RETRIEVAL
1) Image URL: ['https://images-na.ssl-images-amazon.com/images/I/81q5+IxFVUL._SY88.jpg']
Review: Loving these vintage springs on my vintage strat. They have a good tension and great stability. If you are floating your bridge and want the most out of your springs tha
Cosine similarity of images - 1.0000
Cosine similarity of text - 1.0000
Composite similarity score: 1.0000

2) Image URL: ['https://images-na.ssl-images-amazon.com/images/I/71edIGOwydL._SY88.jpg']
Review: i was expecting extreme brutality but got a pleasant surprise. this pup is super clean no. 1 and distortion is very smooth not at all what expected its warm in tone but
Cosine similarity of images - 0.7469
Cosine similarity of text - 0.3271
Composite similarity score: 0.5370

3) Image URL: ['https://images-na.ssl-images-amazon.com/images/I/71-PDjqCBcL._SY88.jpg', 'https://images-na.ssl-images-amazon.com/images/I/71bxnwWGCYL._SY88.jpg']
Review: I have a Gibson Les Paul Standard from 2013 with burstbucker pros.  Supposedly the burstbucker pro is wound to sound like the original PAF (how they actually sounded), w

5 stars!  Don't expect anything less than awesome!
Cosine similarity of images - 0.7439
Cosine similarity of text - 0.2337
Composite similarity score: 0.4888

--------------------------------------------------------------------------------------------
USING TEXT RETRIEVAL
1) Image URL: ['https://images-na.ssl-images-amazon.com/images/I/81q5+IxFVUL._SY88.jpg']
Review: Loving these vintage springs on my vintage strat. They have a good tension and great stability. If you are floating your bridge and want the most out of your springs tha
Cosine similarity of images - 1.0000
Cosine similarity of text - 1.0000
Composite similarity score: 1.0000

2) Image URL: ['https://images-na.ssl-images-amazon.com/images/I/81Z1d7HaBfL._SY88.jpg']
Review: Nice solid springs and defeinitely more silent. Easy installation and the black looks cool.

Pictured with some old uninstalled springs next to them.
Cosine similarity of images - 0.7469
Cosine similarity of text - 0.3271
Composite similarity score: 0.5370
```

The output directly reflects the top-ranked items based on composite similarity scores for both visual and textual queries, structured as follows:

- **USING IMAGE RETRIEVAL**: Demonstrates how visually similar items are identified, complemented by textual relevance through associated reviews, offering insights into how images might align with the user's visual query.
- **USING TEXT RETRIEVAL**: Focuses on textual relevance, with images associated with the top textually similar reviews showcased, providing a nuanced understanding of how textual content aligns with the user's review query.

PART 5 (cntd)

Observations on Retrieval Techniques

Image Retrieval: This technique relies on visual similarities between user-provided images and those in the dataset. The effectiveness of this approach can be highly dependent on the quality and relevance of the feature extraction process used to compute image similarities. If the feature extraction method captures meaningful aspects of the images that align well with the user's intent, image retrieval can yield highly relevant results. However, visual similarities might not always capture the contextual or functional aspects that a user is interested in, especially if those aspects are better described textually.

Text Retrieval: Text retrieval, focusing on semantic similarities between user-provided review texts and dataset reviews, can offer a different dimension of relevance, particularly in capturing the context, sentiment, or specific attributes that users mention in text form. Text-based techniques, especially those leveraging advanced NLP models, can be very effective in understanding user intent and retrieving relevant items based on textual content. However, they might overlook visual aspects that are also important to the user's query.

Which Gives a Better Similarity Score? The "better" similarity score between image and text retrieval techniques can vary based on the user's query context and the nature of the dataset. For queries where visual aspects are paramount, image retrieval might provide more directly relevant results. In contrast, for queries where the semantic content or specific attributes described textually are crucial, text retrieval could be more effective.

Challenges Faced

1. **Multimodal Data Integration**: Combining visual and textual data into a cohesive retrieval system poses challenges in aligning and integrating these modalities effectively. Ensuring that the composite scores accurately reflect the relevance across both dimensions can be complex.
2. **Quality and Relevance of Pre-Computed Scores**: The reliance on pre-computed similarity scores necessitates robust and contextually appropriate computation methods. The quality of these scores directly impacts retrieval effectiveness.
3. **User Query Interpretation**: Accurately interpreting user queries, especially when they involve subjective or context-specific criteria, remains challenging. This can affect the relevance of retrieved items.

Potential Improvements

1. **Advanced Feature Extraction**: Utilizing more sophisticated models for image feature extraction (e.g., using deep learning) and textual analysis (e.g., employing transformer-based NLP models) can enhance the ability to capture meaningful similarities.
2. **Contextual and Semantic Analysis**: Incorporating methods for better understanding the context and semantics of user queries can improve retrieval relevance. This might involve using NLP techniques for query disambiguation and intent recognition.
3. **Feedback Loops**: Implementing user feedback mechanisms to refine similarity scores and retrieval methods over time can help the system learn from interactions and improve accuracy.
4. **Dynamic Weighting of Modalities**: Developing a system to dynamically adjust the weighting between image and text similarities based on the query type or user preferences can offer a more tailored and relevant retrieval experience.

In conclusion, both image and text retrieval techniques offer unique advantages, with their effectiveness largely dependent on the query context and dataset characteristics. Addressing the challenges in multimodal data integration and leveraging advancements in machine learning and NLP can significantly enhance the retrieval process, making it more responsive to user needs and preferences.