CSE508_Winter2024_A2_-2021444-

REPORT

QNS1

Overview

This document outlines the development of a feature extraction component for a Multimodal Retrieval System, focusing on both image and text data processing.

Methodologies

Image Feature Extraction: Utilized ResNet50, a pre-trained Convolutional Neural Network, for extracting image features. Preprocessing steps included resizing, tensor conversion, and normalization.
Text Feature Extraction: Implemented text preprocessing techniques such as lower-casing, punctuation removal, stop word filtering, and lemmatization. The processed text data aimed to enhance the system's understanding and handling of textual content.
Assumptions

Images are accessible via URLs and predominantly contain content relevant to the retrieval system’s domain.
Textual reviews are in English, contain useful descriptive information, and vary in length and detail.

Results

Successfully extracted and normalized features from images, making them ready for similarity comparison and retrieval tasks.
Processed text data to remove noise and standardize format, facilitating accurate TF-IDF scoring in subsequent steps.

Code Functionalities & Libraries Used

The code employs Python libraries and functions to execute image and text feature extraction:

Libraries:

Torch & torchvision: For loading and preprocessing images, and utilizing pre-trained models like ResNet50 for image feature extraction.

Pandas: For handling datasets and facilitating data manipulation.

PIL (Python Imaging Library): To open and manipulate images fetched from URLs.

NLTK: For natural language processing tasks, including stop word removal and lemmatization in text data.

NumPy: For numerical operations, especially during normalization of image features.

Key Functions:

extract_features_from_image: Processes images from URLs using ResNet50 to extract and normalize features.

preprocess_text: Cleans and prepares text data through lower-casing, punctuation removal, stop word filtering, and lemmatization.

save_data: Uses pickle to serialize and save processed data, ensuring the persistence of extracted features and preprocessed text for future use.

QNS2

Overview

This section focuses on the methodologies for text feature extraction in a Multimodal Retrieval System. It emphasizes the preprocessing and transformation of text data to facilitate efficient retrieval based on textual content similarity.

Methodologies

Text Preprocessing: Applied comprehensive text cleaning techniques, including URL and social media handle removal, lower-casing, tokenization, and alphabetic filtering.

Normalization: Employed stemming and lemmatization to normalize words, reducing them to their base or root form.

TF-IDF Calculation: Implemented a method to calculate Term Frequency-Inverse Document Frequency (TF-IDF) scores, quantifying the importance of words within the dataset's corpus.

Assumptions

Textual content is rich in descriptive information, requiring sophisticated preprocessing to extract meaningful features.

The effectiveness of TF-IDF scoring assumes that less frequent words carry more informative weight.

Results

Achieved a standardized and cleaned textual dataset ready for analysis.
Computed TF-IDF scores for each document, enabling the identification of significant words for retrieval tasks.

Code Functionalities

Utilizes nltk for natural language processing tasks such as tokenization, stop word removal, stemming, and lemmatization.

Employs regular expressions (re) to remove URLs and social media identifiers from text.

Leverages pandas and numpy for data manipulation and mathematical operations necessary for TF-IDF computation.

Saves preprocessed text and TF-IDF results using pickle for persistence and future retrieval tasks.

Q3

Overview

This document details the implementation of the retrieval component of a Multimodal Retrieval System, leveraging both image and text features to find and rank similar items.

Methodologies

Cosine Similarity Computation: Utilizes cosine similarity to measure the closeness between a query and dataset items, applicable for both dense (image features) and sparse (TF-IDF scores) vectors.

Retrieval Mechanism: Identifies the top N items most similar to the input query, supporting flexible retrieval across images and text.

Assumptions

The system assumes preprocessed and readily available features for images and text, allowing for immediate comparison and retrieval.

Assumes the effectiveness of cosine similarity as a metric for identifying relevant similarities across multimodal data.

Results

Demonstrates the system's capability to retrieve similar images and reviews based on input features, highlighting practical applicability in multimodal contexts.

Code Functionalities

Numpy & Scipy: Leverages these libraries for mathematical operations and similarity computations, essential for handling both dense and sparse vector comparisons.

Pickle: Utilized for loading pre-processed features and saving retrieval results, ensuring persistence and reusability of data.

Functions like load_data, compute_cosine_similarity, and find_most_similar encapsulate the retrieval logic, making the system modular and extendable.