PROJECT 1

Title: Automatic Image Captioning (Project Supervisor: Prof. Vineet)

Objective: Build an image captioning model to generate captions of an image using CNN

Dataset Link: Flickr8k_dataset

Dataset description: A collection of sentence-based image description

- Dataset consists of 8k images in JPEG format with different shapes and sizes.
- Images are paired with five different captions which provide clear descriptions of the salient entities and events.
- The images were chosen from six different Flickr groups and included a variety of scenes and situations.

Project Overview: Captioning the images with proper description is a popular research area of Artificial Intelligence. A good description of an image is often said as "Visualizing a picture in the mind". The generation of descriptions from the image is a challenging task that can help and have a great impact in various applications such as usage in virtual assistants, image indexing, a recommendation in editing applications, helping visually impaired persons, and several other natural language processing applications. In this project, we need to create a multimodal neural network that involves the concept of Computer Vision and Natural Language Process in recognizing the context of images and describing them in natural languages (English, etc). Deploy the model and evaluate the model on 10 different real-time images.

Tools: Natural Language Toolkit, TensorFlow, PyTorch, Keras

Deployments: FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

Final Submissions:

- GitHub Repository of the project
- Project Technical Report
- Project Presentation with desired outcomes
- Summary of 3 research papers

PROJECT 2

Project Title: AI-based Generative QA System (Project Supervisor: Prof. Manish Shrivastava)

Objective: Finetune any GPT variant model for two tasks:

1. Given the body of an email, generate a succinct subject for the same.
2. Given a ques-on pertaining to the AIML system, model a system to generate its corresponding answer.

Project Overview: This project intends to familiarize the par-cipants with genera-ve text systems. The project will consist of two dis-nct tasks pertaining to the field. In the first task, the par-cipants will get to work with a clean, prepared dataset and try hands-on fine tuning with any GPT model of their choice. While learning how to implement the finetuning of a GPT model on the subject line genera-on task, they will be crea-ng a fresh, new dataset for the next task. The trained QA model can be then deployed for its tes-ng on answering new AIML queries.

The project would offer a complete learning experience, with the project cycle consis-ng of dataset cura-on, idea-on, implementa-on and deployment. We provide an overview of each of the two tasks below.

1. **Email Subject Line Generation**

   As opposed to commonly solved tasks in the domain of news summarization or headline generation – which are closely related works to this problem – the problem[1] offers uniqueness in having to generate extremely short, concise summary in the form of the email subject. This involves iden-fying the most salient sentences from the email body, and abstrac-ng the message contained in those sentences into only a few words. From the implementa-on point of view, this project offers an opportunity to play with genera-ve models in NLP, using any GPT[2] variant of their choice. One would also get to study the evalua-on of text genera-on through different metrics

1. **Question Answering on AIML Queries**

   Having learnt the process of model finetuning and evalua-on on the first task, the project primarily revolves around fulfilling the objec-ve of the second task: modeling a domain-specific GPT-variant model that can answer to ques-ons specific to the AIML course. It has been observed that while pretrained models can produce relevant textual output for general, open-domain textual prompts, the models lack the capability of producing finer outputs when it comes to domain-specific tasks. For this purpose, we commonly finetune the model on a dataset specific to that task, to tailor its exper-se on it. Here, the par-cipants will work together to build a novel, relevant dataset for the task. Post finetuning, they will observe its performance on unseen, related questions.

[1]Introduced in "This Email Could Save Your Life: Introducing the Task of Email Subject Line Genera@on", ACL 2019

[2]GPT 2 (an example of a GPT-variant model) implementa@on on Hugging Face: hPps://huggingface.co/docs/transformers/model_doc/gpt

Dataset :

1. The Annotated Enron Subject Line Corpus: hVps://github.com/ryanzhumich/AESLC This will be used for the first task.
2. AIML QA Corpus: To be curated as a collec-ve effort of all the NLP project team

Dataset description:

1. **The Annotated Enron Subject Line Corpus**
    - The dataset consists of a subset of cleaned, filtered and deduplicated emails from the Enron Email Corpus which consists of employee email inboxes from the Enron Corporation.
    - Evaluation (dev, test) split of the data contains 3 annotated subject lines by human annotators. Mul-ple possible references facilitate a beVer evalua-on of the generated subject, since it is difficult to have only one unique, appropriate subject per email.
    - Some dataset statistics:
        - Sizes of train / dev / test splits: 14,436 / 1,960 / 1,906
        - An email contains an average of 75 words
        - A subject contains an average of 4 words
2. **AIML QA Corpus**
    - This dataset will be created as a collec-ve effort of all the teams par-cipa-ng in NLP projects as a part of the AIML course, and later used for fine tuning the GPT model.
    - Each team will be provided with a ques-on bank consis-ng of 250 ques-ons each. The ques-ons are to be provided with a short, 1-2 line answer to be entered in a CSV file.
    - The given ques-ons will be extracted from the course material already covered through the AIML lectures.
    - Participants will have to adhere to a strict deadline to complete the dataset crea-on task (within 1 month of the project start) to facilitate sufficient -me for QA modeling.
    - Post comple-on of the dataset, a common train / dev / test split will be provided to the teams for experimen-ng on the main task.

Tools: Hugging Face, PyTorch, Tensorflow, Keras, WandB, NLTK

Deployments: FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

Final Submissions:

- Project technical report & presenta-on with desired outcomes
- An overview of the modeling techniques used for the problem
- GitHub Repository of the project
- Summary of 3 research papers

PROJECT 3

Title: Image tagging and road object detection (Project Supervisor: Prof. Anoop)

Objective: Detect object tagging in the video and examine how parallel object detection on multiple patches can allow the detection of smaller objects in the overall image without decreasing the resolution.

Dataset Link: BDD 100K Dataset.

Dataset description: The Berkeley Deep Drive (BDD) dataset is one of the largest and most diverse video datasets for autonomous vehicles.

- The dataset contains 100,000 video clips collected from more than 50,000 rides covering New York, San Francisco Bay Area, and other regions.
- The dataset contains diverse scene types such as city streets, residential areas, and highways.
- Furthermore, the videos were recorded in diverse weather conditions at different times of the day.

Project Overview: Object detection and segmentation methods are one of the most challenging problems in computer vision which aim to identify all target objects and determine the categories and position information. Numerous approaches have been proposed to solve this problem, mainly inspired by methods of computer vision and deep learning. In this project, we aim to build a model which detects multiple objects and segmentation in a moving video. For eg. Image tagging, lane detection, drivable area segmentation, road object detection, semantic segmentation, instance segmentation, multi-object detection tracking, multi-object segmentation tracking, domain adaptation, and imitation learning.

Tools: TensorFlow, PyTorch, Keras

Deployments: FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

Final Submissions:

- GitHub Repository of the project
- Project Technical Report
- Project Presentation with desired outcomes
- Summary of 3 research papers

PROJECT 4

Title : Automatic Speech Recognition(ASR) (Project Supervisor: Prof. Anil)

Objective: Build an ASR model for converting speech to text.

Dataset Link : LibriSpeech

Dataset description: LibriSpeech is a corpus of reading English speech, suitable for training and evaluating speech recognition systems, published in 2015 by Johns Hopkins University. It is derived from audiobooks that are part of the LibriVox project and contains 1000 hours of speech sampled at 16 kHz of 2000 speakers. The LibriVox project1, a volunteer effort, is responsible for the creation of approximately 8000 public domain audiobooks, the majority of which are in English. Most of the recordings are based on texts from Project Gutenberg2, also in the public domain. The data is already divided into train/dev/test sets. The total size of the data is 60 GB and subsets are available of different sizes.

Initially, we recommend working only with 'dev-clean' and 'test-clean' datasets for building the model. We can use any one or a combination of both data sets as a training set. A subset of either 'dev-clean' or 'test-clean' can be used for testing purposes. Once modeling is done with these smaller data sets, start modeling using 'train-clean'/'train-other' data sets of larger sizes as a training set. Now, 'dev-clean', 'test-clean', and 'test-other' datasets are used for validation/testing purposes only.

Project Overview: Automatic speech recognition is the application of Machine learning or AI where human speech is processed and converted into readable text. We can find numerous applications such as Instagram for real-time captions, Spotify for podcast transcriptions, Youtube video transcription, Zoom meeting transcriptions, etc. The field has grown exponentially over the last few years. An explosion of applications taking advantage of ASR technology in their products to make audio and video data more accessible.

There are different approaches to Automatic Speech Recognition, viz. traditional HMM (Hidden Markov Models) and GMM (Gaussian Mixture Models) and end-to-end deep learning models. In this project, we aim to build and deploy a model that can generate the written text from the speech with a decent accuracy.

Tools: Kaldi, PyTorch, Audio Processing tool/library.

Operating system: Linux - Ubuntu

Deployments: FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

Reference: Papers using libriSpeech

Final Submissions:

- GitHub Repository of the project
- Project Technical Report
- Project Presentation with desired outcomes
- Summary of 3 research papers