Project Anuvaad

**Aim** – Ideology on Project Sunbird Anuvaad

**Abstract:**

**Introduction:**

Sunbird Anuvaad is open- sourced project that based on Al and ML technologies to end to end document translation on Indian languages.

**Requirement of projects / need / Problem Solving:**

1. 1)India has historically been a country with many different countries and languages with English serving as subsidiary official language. In current Century, growing demand for translation services in local Languages best project for business exponentially growth.
2. Documents differ from one state to the next, and when someone moves from one linguistic state to another, he must update his documents in the regional language. All of this confirms the use of **Indian language translation** services.
3. 0Indian language translation services can help hospitality brands like a hotel to make their international guests stay longer with the help of multilingual online as well as offline promotional collateral.
4. In today’s globalized business world, language translation has become an important part of communication. Business needs to speak customer language that makes easier for branding and promoting the product.

**Basic Structures:**

**Basic workflow:**

**Observation On Technology Stack:**

**1)Open sourced and Rest API:**

* It is one of the major benefits of open-source software to be able to change, edit, and use the code however one wants. Developing customized open-source software for users may seem more useful to developers than to users, but users greatly benefit from the options that developers create.
* Due to multiple parties making modifications and improvements, OSS inevitably has containment, performance, and security flaws.
* Due to its difficulty setting up and lack of friendly user interfaces and compatibility problems, it is harder to use and adapt.
* When free software, more customization, and more transparency are important, open-source is better. If the user wants something that is stable, has more software support, and is more user-friendly, closed-source is better.

**2)PRIMA – Layout detection mode**

**Layout parse and Label Studio**: (customised data feed to the model for training & better accuracy on new data)

* Since raw data documents have a lot of complicated layouts and existing OCR, we use a layout parser and label studio to tackle this problem
* Layout parse supports extraction based on layout data and API, so it can very quickly generalize to different types of data.
* Label studio refines some data for new model uses so that we can predict with high accuracy on new types of data using new models.
* Then Label layout annotation using cocosplit tool to split a multi-class annotation dataset with preserving class distribution among train and test set.
* Its pip install library uses in deep learning so, it helps to solve the following problem
* Extract structured data from complex documents, such as scientific papers, newspapers, and business analytics papers.
* Starting from raw text data, the output is never great and it is difficult to train our models or learn from it.
* It is difficult to label layout annotations in large text documents.

**3) CRAFT – Character Region Awareness for Text Detection**

- it is scene text detection (STR) method based on neural network that showing promising result. PyTorch implementation that effectively detect text area by exploring each character region and affinity between character. Region and affinity with character level bounding boxes.

-The region score is used to localize individual characters in the image, and the affinity score is used to group each character into a single instance.

- CRAFT is detect individual character even when character level annotation is not given.

4)  **LASER**

- Sentence Alignment is a process of aligning sentences that 'mean the same' from two unorganized sets of sentences. Suppose we two document doc\_1 has English sentence and doc\_2 has Hindi sentence where we match a sentence S1 from D1 to a sentence S2 from D2 if S1 and S2 have the same meaning.

In order to achieve this, we make use of pre-trained encoder-decoder models that transform these sentences into vector representations, these vector representations (embeddings) are then used to calculate the similarity between the sentences.

LASER stands for Language-Agnostic SEntence Representations which developed by Facebook AI to support around 93 languages.

1. LASER is trained on parallel datasets from various resources like Wikipedia, mapping a sentence from one language directly to another language in order to encourage consistency between the sentence embeddings.
2. LASER's embeddings are 1024 dimensional which is much comprehensive as compared to LaBSE's embeddings that only account for approx. 700 dimensions per sentence.
3. LASER being built on PyTorch runs into issues while handling parallel calls, upon making API calls in parallel, we kept getting warnings on the console and eventually, the system broke upon 10 parallel calls simultaneously.

**Results:**

1. **Recognition**
2. **Availability**
3. **Accuracy**
4. **Speed**

**Conclusion:**

**As a result of studying AI ML technology, I believe that the following technology can be used to improve recognition, accuracy, and speed during the application phase of a project.**

1. **COCO-text:** In terms of recognition accuracy, its reading document is accurate from low graphics images, and its capabilities are based on three tasks: text localization, cropped word recognition, and end-to-end recognition. [ refrence]
2. **FiftyOne tool:** open-source machine learning developer tool and it is designed to let researchers and engineers easily work with and visualize image and video datasets with annotations and model predictions stored in various formats. [R]
3. **End-to-end text detectors**: An end-to-end approach trains the detection and recognition modules simultaneously to enhance detection accuracy by leveraging the recognition result. Because relatively small receptive field is sufficient to cover a single character in a large image, which makes CRAFT robust in detecting scale variant texts.

**Reference:**

1. [**https://towardsdatascience.com/how-to-work-with-object-detection-datasets-in-coco-format-9bf4fb5848a4**](https://towardsdatascience.com/how-to-work-with-object-detection-datasets-in-coco-format-9bf4fb5848a4) **parsal layout detection replace Fifty-one**
2. [**https://rrc.cvc.uab.es/?ch=5&com=tasks**](https://rrc.cvc.uab.es/?ch=5&com=tasks) **for cocosplit tool replace coco text tool**
3. T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun. An end-to-end textspotter with explicit alignment and attention. In CVPR, pages 5020–5029, 2018 [end to end text detector]
4. <https://www.linkedin.com/pulse/labse-laser-sentence-alignment-vishal-mahuli/>
5. https://medium.com/doma/using-nlp-bert-to-improve-ocr-accuracy-385c98ae174c