Odyssey - InkWave: Transforming Ink to Bytes

PART 2

Project Proposal

Authors: Marc Cruz, Prerna Joshi

Status: In Review

Date: Jan 22, 2024

Learning Objectives

An ambitious project to hone ML related skills on a practical project! Some objectives are:

- Gain familiarity with current Computer Vision Machine Learning Models, Large Language Models and Natural Language Processing Models
- Learn various parts of application development
- Learn to be cohesive in a team environment
- Gain experience with the development lifecycle via agile/scrum

Project Goals

The objective is developing an application for a machine learning model capable of converting handwritten notes into digitized ones. The digitized text would be used to be formatted into various files, including pdf and markdown. We will be building a UI for the ML model so that we can see it in the works and use it when required easily.

Management

In terms of leadership, the software lead would be delegating portions of the work once in 2 weeks. Having developers syncs weekly at the same time when sprints are coming to an end and gather everyone's updates. Designed in a way to gain experience in a scrum/sprint structured team over 10-12 weeks.

Dataset

We have used these kaggle datasets to train our models: https://www.kaggle.com/datasets/landlord/handwriting-recognition/data (Learning CV Model Dataset)

https://www.kaggle.com/datasets/vipin20/nlp-word-correction (Learning NLP/LLM Model Dataset)
https://writeoff.cs.byu.edu/censustree/dataset.tar.gz (CV dataset)
https://www.kaggle.com/datasets/jpmiller/layoutlm/data(Tentative NLP/LLM Model dataset)

Github:

https://github.com/cppsea/InkWave/tree/main

Timeline

Spring Semester 24:
Sprint 1: Figuring out the datasets to be used, cleaning and splitting the datasets, researching Computer Vision and Natural Language Processing, and understanding their algorithms.
Sprint 2: Starting to build the CV and NLP/LLM Models
Sprint 3: Training and validating the Models
Sprint 4: Fine-tuning the models and starting to figure out how to integrate them.
Sprint 5: Integrate the models and train/test them looking for bugs or issues.
Sprint 6: Evaluating the integrated model, finding weaknesses, and tweaking the models accordingly.
Sprint 7: Making sure the model is properly completed and ready to be transitioned into an app/ UI

Fall Semester 24 (App/UI) :
Sprint 1: Kickoff & Requirements (All Teams)

- **Front-end**: Finalize UI/UX design, wireframes for scanning, viewing, and reformatting notes.
- **Back-end**: Set up database schema, define API structure for handling data and ML interactions.
- **ML Management**: Review the existing ML model to ensure smooth integration with the app, prepare APIs for model execution.

Sprint 2: Core Architecture & API Setup

- **Front-end**: Build basic UI for scanning, reformatting, and viewing converted notes.
- **Back-end**: Implement database structure, set up API endpoints for the chained model and NLP.
- **ML Management**: Set up the API for the existing ML model, ensure proper data flow between front-end and back-end.

Sprint 3: Front-end & Model Integration

- **Front-end**: Complete UI for uploading, viewing previous conversions, and displaying results.
- **Back-end**: Ensure smooth API communication between front-end and ML model, handle data storage.
- **ML Management**: Integrate the ML model with the API, test conversion workflows (e.g., handwritten to text).

Sprint 4: Testing & Refinement

- **Front-end**: Conduct user testing for scanning and formatting features, adjust UI based on feedback.
- **Back-end**: Perform integration testing for APIs, ensure correct data flow between components.
- **ML Management**: Test ML model accuracy with various handwriting styles and adjust if needed.

Sprint 5: Bug Fixes, Optimization & Deployment

- **Front-end**: Fix UI bugs, finalize design, and optimize performance.
- **Back-end**: Final testing of the API and database, optimize performance.
- **ML Management**: Final ML testing, ensure seamless integration with the back-end, and prepare for deployment.

Computer Vision Machine Learning Models

Computer vision models are designed to translate visual data based on features and context identified during training. These models commonly use Neural networks,k-means clustering and SVM algorithms. It is often used for feature extraction,object recognition and detection and image classification. These types of models are often used in facial recognition, autonomous vehicles,augmented reality,text extraction, etc.

Supplemental Material:
https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-computer-vision#object-classification
https://www.ibm.com/topics/computer-vision
https://www.sciencedirect.com/science/article/pii/S1877050920308218#:~:text=The%20commonly%20used%20algorithms%20are,%2C%20graphic%20documents%2C%20and%20videos.

Natural Language Processing Models

Natural Language Processing is a field of AI that focuses on the interaction between computers and human language. It enables computers to understand,interpret and generate human language in a way that is meaningful and relevant. It uses the concepts of tokenization, POS Tagging, Named Entity Recognition, Sentiment Analysis, Text Classification, Machine Translation. These models commonly use the algorithms of Statistical models, Hidden Markov Models(HMMs), SVM, Conditional Random Fields(CRFs), Rule based systems, Word Embeddings, RNN, LSTM, CNN and Transformer Models.

Supplemental Material:
https://www.ibm.com/topics/natural-language-processing
https://www.deeplearning.ai/resources/natural-language-processing/
https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/what-is-natural-language-processing-nlp

Large Language Models

Large Language Models (LLMs) are advanced artificial intelligence models that are trained on vast amounts of text data to understand and generate human-like language. They use deep learning techniques, particularly transformer architectures, to process and generate text. LLMs like OpenAI's GPT (Generative Pre-trained Transformer) series, such as GPT-3, are capable of performing various natural language processing tasks, including text generation, summarization, translation, question answering, and more. These models have gained significant attention for their ability to generate coherent and contextually relevant text across a wide range of applications, but they also raise concerns regarding ethical use, biases, and potential misuse.

Supplemental Material:
https://www.ibm.com/topics/large-language-models

https://www.techtarget.com/whatis/definition/large-language-model-LLM
https://cloud.google.com/ai/llms

APP Dev
https://www.youtube.com/watch?v=fis26HvvDII
https://www.velvetech.com/blog/mobile-app-development-process/