models to be downloaded in DSP:

bert-large-cased: <https://huggingface.co/google-bert/bert-large-cased>

en\_core\_web\_sm : en\_core\_web\_sm-3.7.1-py3-none-any.whl

bert-base-cased: <https://huggingface.co/bert-base-cased/tree/main>

roberta base: <https://huggingface.co/FacebookAI/roberta-base/tree/main>

paddlepaddle models

<https://github.com/explosion/spacy-models/releases/download/zh_core_web_sm-3.7.0/zh_core_web_sm-3.7.0-py3-none-any.whl>

<https://paddleocr.bj.bcebos.com/PP-OCRv4/chinese/ch_PP-OCRv4_det_infer.tar>

<https://paddleocr.bj.bcebos.com/PP-OCRv4/chinese/ch_PP-OCRv4_rec_infer.tar>

<https://paddleocr.bj.bcebos.com/dygraph_v2.0/ch/ch_ppocr_mobile_v2.0_cls_infer.tar>

gliner model

<https://huggingface.co/urchade/gliner_large-v2.1>

NLLB model

<https://huggingface.co/facebook/nllb-200-1.3B>

word2vec

<https://huggingface.co/fse/word2vec-google-news-300>

ollama model and software

Ollama software has to be downloaded (download link- <https://ollama.com/download>)

to pull , run following commands in cmd-

> ollama pull llama3

> ollama pull llama2

Thanks nd Regards,

Anjali

|  |  |
| --- | --- |
| As a user, I want to use regex to select biographies for creation of dataset for labelling exercise  (2839) | "Label list preparation 1) prepare list of labels to be extracted from biographies  Biography selection - 15 1) Apply appropriate filters on biography column to -(1) a) select biographies of individual watchlist with  b) v\_info\_source in PEPEX, PEPIN, ANEWS, SANEX c) select only unique biographies 2) For labels in label list, formulate regex to extract information on 18 different labels from biographies. - (14) a) 18 labels are- Name, ID, DOB, POB, Nationality, Gender, CPO, CPI, Occupation, title, crime, deceased status, marital status, location, address, descent, other names, only relation. b) Regex is written for 16 (excluding CPO and other names) labels.  c) More than one regex is written as each label has variations. In total - there are 46 sub labels (excluding CPO and other names).   Consolidated dataset creation - 1 3) Select and create dataframe of biographies for each variation.  4) Iteratively perform these steps- a) Merge biographies to form a consolidated set b) re apply regex formulated to find count of variations in the consolidated set c) For variations where count is low, merge new biographies  d) repeat b) and c) steps till we've reached maximum count of entity variation, or the count is atleast 300. |
| As a user, I want to generate labels on the biographies in the created dataset using a pretrained model  (GLINER)  (2840) | Label Dataset Creation and model based labelling- 2 5) Assign unique identifier to each biography in the final consolidated dataset  6) Generate tokens from biographies. and  7) Create column for 'Label'   Generating pretrained model prediction - 12 1) Define tags for each label (and variation). The tags are selected on trial and error basis, depending on which can generate the best prediction. (6) 1) Use pretrained model to predict entities on selected biographies - (1) 2) Arrange predictions so that they align with generated tokens - (2)  Final Dataset for labelling - 2 1) Create dataset with unique biography identifier, watchlist biography, tokens, labels, PWM\_ID/WL\_ID as columns 2) Save the dataset  3) create a editable dataset from saved dataset for labelling" |
| As a user, I want to generate labels for the created dataset from the biographies  (2840) | 1) Go through each token and understand the context to assign one of the entities  2) Tag B-entity or I-entity or O-  B-entity1 I-entity1 B-entity2 I-entity2 .. .. 3) Check model prediction and make edits if any |
| As a system, I want to preprocess input text, prepare tags and examples to create an LLM based model for extraction of entities and relations from biography data  (Langchain Chat model) | Document Parsing and Preparation - 4 1) Load biography using LangChain CSV loader 2) Split documents into manageable chunks using different chunking strategies like RecursiveTextSplitter, Semantic Chunking, etc.   Input preparation - 28 1) Entity tags - (8) i) For each of 18 tags, design label(s) which can extract the entities  ii) Improve the entity labels, description and label definitions 2) Relations tags - (8) i) For each of 18 tags, design label(s) which can extract the relations ii) Improve the relation labels, description and label definitions 3) Prepare examples to be given for few shot learning by the model. Specify text, head\_type, head, tail, tail\_type, relation - (8) This step is experimental and has to be done while observing model prediction.  4) Define prompts (system prompt and human prompt) - (2) 5) Define output format - (2) |
| As a system, I want to create an LLM based model for extraction of entities and relations from biography data  (Knowledge graph llama3) | Model prediction - 6 1) Load LLMs available for the task (Llama3) 2) Load the biography, split it into sentences 3) In each sentence, apply the model 2) Apply the model and generate the predictions  Post processing and standardisation of model prediction - 12 1) Convert pronouns into entity using coreference resolution- (6) 2) Analysis of prediction generated by the model to identify standardisation steps 3) Apply standardisation steps to get (tags, entities) 4) Convert (tags, entities) to column of tags format  Performance Evaluation - 4 1) Apply post processing steps on the model output 2) Compare output generated by the model to ground truth labels 3) generate recall and precision scores for the model |
| As a system, I want to create an Graph based model using LLM chain for extraction of entities and relations from biography data  (GraphLLMChain) | Input preparation - 8 1) Modify the entity tags, relation tags, few shot examples according to expected output from the GraphLLMChain model Eg: for graph based model, we'll have to transform examples so that the output is in graph format  Model Prediction - 2 2) Load LLM and split into chunks 3) Apply the model for each chunk and aggregate the output  Post processing and standardisation of model prediction - 8 1) Convert pronouns into entity using coreference resolution - 2  2) Analysis of prediction generated by the model to identify standardisation steps - 6 3) Apply standardisation steps to get (tags, entities) 4) Convert (tags, entities) to column of tags format  Performance Evaluation - 4 1) Apply post processing steps on the model output 2) Compare output generated by the model to ground truth labels 3) generate recall and precision scores for the model |