Proposed workflow

Plab Group 2 project - NLP

Deepika Pradeep

Overview

Create corpus of PubMed abstracts

Tag abstracts with keywords using pre-trained NER model

Store abstracts in a database with tagged metadata for later queries

Create a GUI that can be used to query the corpus

Retrieving PubMed abstracts with E-utilities

ESearch

Searches Pubmed, returns list of unique identifiers (UID)

EFetch

Returns full data records for list of UIDs. See example

Result format

Store retrieved results as cache files

Guidelines

Follow NCBI guidelines when using API

db: pubmed; **term**: ?, **retmax** & **restart** (find out how many results using **EGQuery**), **edat**	**db**: pubmed; **id**: list of UIDs; **retmax** & **restart**; **rettype**: abstract; **retmode**: text	- **JSON** format - Metadata - MeSH terms - Use BioC API? Simpler format - BioC JSON / BioC XML	- <= **3 requests/second** - large jobs - **off-peak hrs** - use the **tool** and **email** parameters for id - **up to 100,000 UIDs** can be retrieved with one URL

Parameters/Specifications

Constructing the NER pipeline

- Download Huggingface transformers library and install required packages
- Preprocess data before training
- Initialize pretrained model, hyperparameter optimization/Bayesian optimization
    - Set max length
- Convert data to tensors, load into dataloaders
- Train and validate over a couple of epochs - store metrics
- Evaluate model performance - learning curve

NER

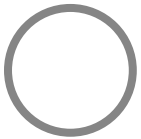
Huggingface

- Download transformers library and install required packages

BioBERT

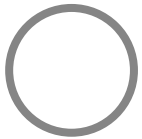
- APIs for BioBERT, BERT tokenizer

- BioBERT - trained on Pubmed abstracts, PMC articles - more suitable for task



Training

- Preprocess data before training
- Initialize pretrained model; set max length
- Hyperparameter optimization
- Convert data to tensors, load into dataloaders
- Train and validate over a couple of epochs - store metrics



Evaluation

- Evaluation metrics: confusion matrix, precision, recall
- Look at learning curve to evaluate model performance; underfitting/overfitting

Workflow

- Collect abstracts from PubMed using API script and store as cache files
- Similar to assignments 4-6, build a database, add abstracts iteratively to database (function should recursively add entries from files in a specified directory)
- Use a subset of the abstracts to fine-tune the NER model (if needed take data from 4 years ago instead of 3); add tags for each abstract in the test data set using the NER model
- Build a GUI that can retrieve entries in the database when user queries it
- When a search term is entered, return all abstracts that the NER model tagged with the same search term

Backend in summary

Building API

- E-utilities
  - Parameters **rettype** (returned view) as abstract; **date of publication** (from the last 3+ years)
- Follow API guidelines! Avoid peak hrs
- Metadata elements to select entries
  - MeSH terms as tags
- Use BioC API? BioC format - easier to deal with text and annotations
- **FASTAPI**, **requests** libraries

⭐CLI command - to run API
⭐Unit testing - check for cache file, check for correct date, check cache file for relevant fields

Storing abstracts

- Make database as in Plab 6
- Fields in database
  - Abstract (free-text)
  - Date of publication
  - MeSH terms
  - NER tags
- **SQLAlchemy** to communicate with database

⭐CLI commands - to make database, add entries from files in a directory
⭐Unit testing - check database created, check number o

NER

- Finetune NER model
  - **BioBERT** trained on Pubmed abstracts, PMC articles - so I think it is most suitable
- Optimize hyperparameters - Bayesian hyperparameter optimization?
- Create **NER pipeline**
- Diagnose model by looking at learning curve

⭐CLI command to run NER pipeline
⭐Unit testing?