ATLAS Submission

CCDS Project template (https://cookiecutter-data-science.drivendata.org/)

The ATLAS submission to the Data Competition for NVDRS youth novel variables.

Summary

We applied a mixed-methods approach to NVDRS Narrative data to examine the types of events and experiences related to decedent engagement in online spaces. Aligned with our theoretical framing, we propose the following five variables be added to NVDRS: “Private Sharing”, “Conflict”, “Victimization”, “Withdrawal”, and “Harmful Exposure”. Specifically, we 1) use open-coding of NVDRS narratives to identify key themes related to decedents’ non-normative behaviors online, and 2) use large language models (LLMs) to classify each narrative into these themes.

For 1: To obtain a sub-sample for annotation with relatively high density of relevant cases, a set of 51 keyphrases and heuristic inclusion/exclusion rules were developed to identify a subset of 1,279 narratives that are likely to reference the decedent’s online behavior. To identify these key-phrases, we started with a set of 15 hand-collected key-phrases (e.g., social media, phone, video) and augmented these using log-odds ratios of word frequencies and word embeddings to identify words occurring in similar narratives. These narratives contain words or phrases often used to describe online activity (e.g., “online” or “post”). Then eight authors conducted a thematic analysis to develop a codebook followed by coding of these cases. We manually annotated 472 unique narratives; 388 of these were singly annotated and 84 were multiply-annotated (by 2-4 annotators each) for inter annotator agreement, so there were 642 annotations in total.

For 2: We conducted zero-shot learning using Meta’s open-source LLM Llama3.1-8B-Instruct. Our inferece had three steps: First, we prompted the LLM to break the narratives into individual sentences to account for the model’s difficulty in processing the full narrative. Second, we prompted the LLM to recognize whether each sentence references online spaces to help distinguish between online behaviors of interests and their offline counterparts. Third, we craft a multiple-choice prompt for each code to create clear inclusion and exclusion criteria for each code.

Setup

1. Install Python 3.11.9
2. Set up Jupyter Notebooks in a CUDA 12.4 environment
3. Install the required python packages (see `requirements.txt`). You may get an error about failing to install pyproject, but this shouldn’t affect performance.

Hardware

The solution was run on one NVIDIA RTX A6000 GPU, using vLLM0.5.4, Hugging Face Transformers 4.43.3, and PyTorch 2.4.0 on Python 3.11.9 in a CUDA 12.4 environment.

Inference time: 85 minutes

A detailed breakdown of how long each step took is given in `notebooks/run_inference.ipynb`

Project Organization

The project follows Data Competition’s Cookie-Cutter style.

```
├── LICENSE            <- Open-source license if one is chosen
├── Makefile           <- Makefile with convenience commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources. (in this case, the annotations)
│   ├── interim        <- Intermediate data that has been transformed. (in this case, the files used to identify keywords)
│   ├── processed      <- The final, canonical data sets for modeling. (in this case, the final keyword and IAA tables)
│   └── raw            <- The original, immutable data dump. (in this case, the raw competition data)
│
├── docs               <- (Unused) A default mkdocs project; see www.mkdocs.org for details
│
├── models             <- (Unused) Trained and serialized models, model predictions, or model summaries (empty)
│
├── notebooks          <- Jupyter notebooks for running inference. Naming convention is a number (for ordering) and a short `-` delimited description, e.g.
│                         `1.0-initial-data-exploration`.
│
├── pyproject.toml     <- Project configuration file with package metadata for
│                         src and configuration for tools like black
│
├── references         <- (Unused) Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── setup.cfg          <- Configuration file for flake8
│
└── src            <- (Unused) Source code for use in this project.
    │
    └── __init__.py    <- Makes src a Python module
```

Run inference

The notebook `notebooks/3-run-inference.ipynb` shows all prompts used to generate predictions. It uses the annotations (in `data/external`) and raw competition data (in `data/raw`) as inputs, and saves two files containing a) the narratives split into sentences and b) the predictions for each case.