**Final Project Proposal**

Group 3: Jehan Bugli

**Overview**

The Congressional Research Service (CRS) is a vital organization that assists legislators with [policy analysis](https://www.congress.gov/crs-products), [official bill summaries](https://www.congress.gov/help/bill-summaries), and more; however, their job is rapidly becoming more difficult due to [rapidly increasing bill introductions](https://rollcall.com/2025/03/05/publishing-pileup-congressional-bills-slow-to-reach-public/) causing significant backlogs for official text releases.

While the CRS is reportedly [exploring AI applications](https://fedscoop.com/congressional-research-service-eyes-ai-bill-summaries/), the current workflow is largely manual; when paired with growing work volume and [potential federal spending cuts](https://www.nbcnews.com/politics/congress/senate-republicans-vote-advance-massive-budget-blueprint-trumps-agenda-rcna199509) on the horizon, the utility of automated assistance is clear.

I aim to explore topic tagging automation; namely, using NLP methods to accurately tag bills with one of [32 policy area designations](https://www.congress.gov/help/field-values/policy-area). If successful, I think this could have an appreciable impact on CRS workflows by reducing monotonous work and expediting processing. If the policy area designations are changed, the training pipeline could also be reconstructed to propose tags retrospectively.

**Datasets**

To provide utility for a CRS analyst’s workflow, training data must be limited to the text one would have available. Official bill text (in XML format, including federal code citations) can be retrieved in XML format from the [GovInfo bulk data repository](https://www.govinfo.gov/bulkdata).

The repository also stores prior CRS-generated information; the dataset will consist of bills that already have policy area tags.

I plan to explore whether or not incorporating cited code text could prove useful; federal code sections can be retrieved from this repository or [a House website](https://uscode.house.gov/download/download.shtml).

**Methods**

An interpretable modeling approach would be ideal, and I aim to start with rule-based models as a result. The CRS would likely want the ability to explain tagging results to legislators and stakeholders as needed. I then plan to explore added gains from discarding interpretability (potentially moving to autoencoder-based feature extraction) and judge the practical utility in doing so.

**Packages (tentative)**

* NLTK: Preprocessing and tokenization exploration
* spaCy: Preprocessing, tokenization exploration, feature extraction
* Pandas / Polars: Dataframe processing and manipulation
* sklearn: TF-IDF feature extraction, modeling
* Pytorch: Modeling depending on classical results
* uv: Project / package management (not necessary but I think this project makes for a nice test case)

**Tasks**

As noted above, the main task is classification; I aim to match legislative text up with a known set of 32 policy areas.

**Performance**

I think that the best metric for real-world use is recall (the rate of true positives over all actual positive instances). In most cases, private individuals and entities care about a relatively small policy area subset; while filtering through irrelevant items is monotonous, missing a relevant piece of industry-relevant legislation would be a serious issue for a firm’s government affairs officer. There are business cases to be made for other scores, but I intend to make recall the primary metric.

**Schedule (tentative)**

* Week 1: Complete data collection and a feature extraction pipeline
* Week 2: Complete modeling exploration (classical + others as needed)
* Week 3: Work on app functionality and refine the modeling pipeline