Legal Brief Argument - Counter Argument Linking

Objective

This project aims to analyze legal briefs by linking argument sections between "moving briefs" and "response briefs" filed in court cases. The focus is on identifying and matching arguments from the initial brief with corresponding counter-arguments in the response. Automating this analysis will help aiding in legal research and argument drafting.

Brief documents are filed by plaintiff / defendant attorneys to argue their stance in support / opposition of a motion. These briefs are filed in a back-and-forth fashion until the judge passes an order on the motion. Briefs contain various section types: introduction, standard of review, factual summary, argument sections, conclusion, etc. Arguments form the crux of a brief.

You are presented with a list of argument sections extracted from a list of brief pairs. The first brief in every brief pair is called the “moving brief” and the second brief in the pair is called the “response brief”, since it is filed in response to the moving brief. Your task is to link argument sections from the moving brief to those in the response brief, thereby getting the set of counter arguments (from the response brief) for every argument in the moving brief. Every argument in the moving brief can have multiple counter arguments from the response brief.

Note: The argument extractor is imperfect, so it is possible that some sections are not actually arguments, but you do not have to implement a separate classifier to identify if a section is actually an argument.

Dataset

You are provided a dataset with 10 brief pairs. For every brief, you are also provided with the list of all the extracted argument sections, each with a heading and content (paragraphs). Additionally, you are provided the actual links (`true_links`) for 8 of the 10 brief pairs for use in your tuning and experimentation. The true links are represented by the pairs of argument headings that should be linked, where every link consists of one heading from a moving brief argument and the corresponding counter-argument’s heading from the response brief. These are present for the `split: train` brief pairs.

Data Format

```
Python
{
```

```
        "moving_brief": <brief_object>,
        "response_brief": <brief_object>,
        "true_links": [
                [moving_brief_heading_i, response_brief_heading_j], ...
        ],
        "split": <train_test_split:str>  # "train" or "test"
}
```

Where a brief_object has the following format:

```python
{
        "brief_id": str,
        "brief_arguments": [
                {
                        "heading": <argument_heading:str>,
                        "content": <argument_paragraphs:str>,
                }, ...
        ]
}
```

Evaluation

You must present your predicted links on the `split: test` pairs during your pitch for manual evaluation.