LREC-COLING 2024

ELRA ICCL

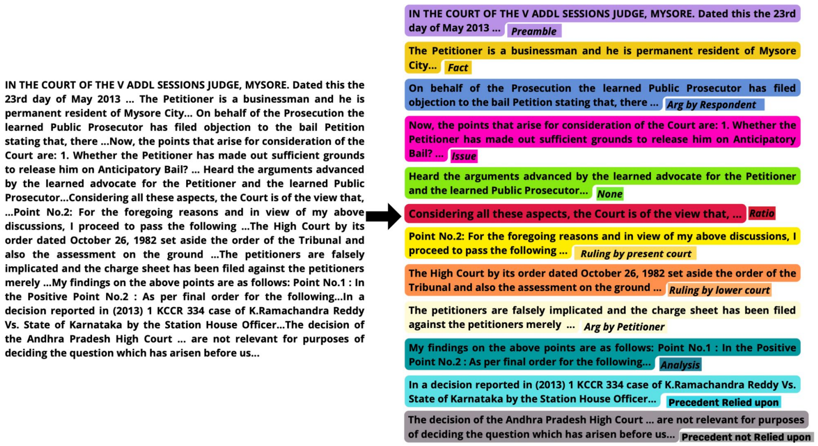
# Mind Your Neighbours: Leveraging Analogous Instances for Rhetorical Role Labeling for Legal Documents

**Santosh T.Y.S.S, Hassan Sarwat, Ahmed Abdou, Matthias Grabmair**
Technical University of Munich, Germany

# Rhetorical Role Labeling

- Assigning functional roles to the sentences in the legal judgement
  - Such as preamble, factual content, evidence, reasoning, etc.
  - Essential for various tasks, such as case summarization, semantic search and argument mining

- Challenges to tackle RRL
  - Contextual dependencies - surrounding sentences and case's context
  - Intertwining nature of rhetorical roles
    - Rationale behind a judgment (Ratio of the decision) often overlaps with Precedents and Statutes
  - Limited annotation data
  - Label imbalance among different rhetorical roles



# Prior Works

- Initially as sentence classification, treating each sentence in isolation using CRF and hand-crafted features (Saravanan et al., 2008, Savelka and Ashley 2018, Walker et al. 2019)

- Later sequential sentence classification, addressing contextual dependencies between sentences (Yamada et al., 2019, Bhattacharya et al., 2021; Ghosh and Wyner, 2019; Malik et al., 2022; Kalamkar et al., 2022).
  - Effectively addresses contextual dependency challenge of RRL, other challenges remain unaddressed.

- Address data scarcity through data augmentation (Santosh et al. 2023)
  - But word deletion, sentence swapping and back-translation introduce noise and disrupt coherence

# Current Work - Leveraging “Neighbours”

- Harnesses knowledge from semantically and contextually similar instances - “Neighbours”
  - Grasp underlying rare patterns.
  - Enhance understanding of complex label-semantics relationships
  - Improve nuanced label assignments to handle less common labels

- Explore approaches to incorporate these neighbours
  - Directly at inference time
    - Using label Interpolation with
      - K-nearest neighbors, Single, and Multiple prototypes
  - During training
    - Contrastive, Novel Discourse-aware Contrastive learning
    - Single and Multi Prototypical learning

- Assess cross-domain generalizability (train on one dataset and test on the other dataset) of our methods

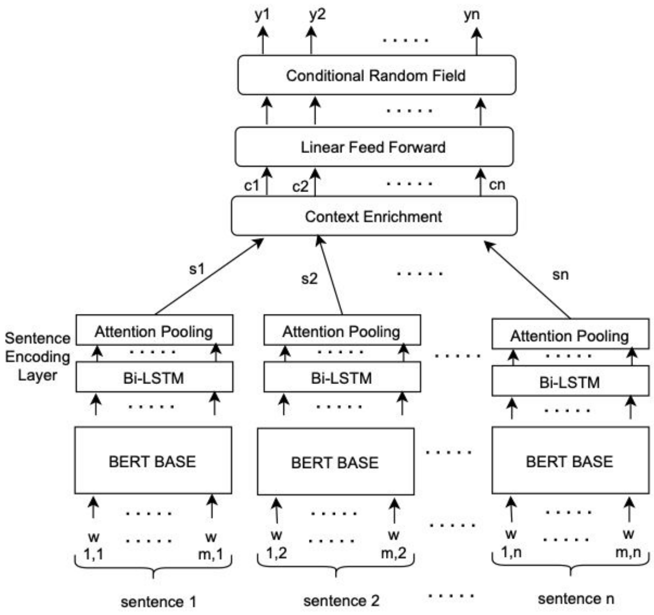
# Dataset & Metrics

- **Build** (Kalamkar et al., 2022)
  - 214 Judgments from Indian supreme court, high court, and district courts.
  - Tax and Criminal law cases
  - 13 rhetorical role labels, including ‘None’.
- **Paheli** (Bhattacharya et al., 2021)
  - 50 judgments from the Supreme Court of India
  - 5 domains: Criminal, Land and Property, Constitutional, Labour and Industrial, and Intellectual Property Rights
  - 7 rhetorical roles.
- **M-CL / M-IT** (Malik et al., 2022)
  - Judgments from the Indian Supreme Court, High Courts, and Tribunal courts.
  - M-CL - 50 documents - Competition Law
  - M-IT - 50 documents - Income Tax cases
  - 7 rhetorical role labels

Metrics: Macro-F1 and Micro-F1

# RRL Baseline

Hierarchical Sequential Labeling Network (Kalamkar et al., 2022)



# RQ1: Neighbours at inference

- Interpolation with KNN
  - After training, construct the datastore as set of all contextualized sentence representation-rhetorical label pairs from all the training examples

$$\{K, V\} = \{(c_i, l_i) | \forall x_i \in x, \forall l_i \in l, (x, l) \in D\}$$

  - During inference time, find the k-nearest neighbours N
  - Derive the distribution of labels using labels of the retrieved neighbours based on softmax of their negative distances

$$p_{kNN}(l_i | x, x_i) \propto \sum_{(k,v) \in N} \mathbb{1}_{l_i = v} \exp\left(\frac{-d(c_i, k)}{\tau}\right)$$

  - Finally interpolate with baseline

$$p_{final}(l_i | x, x_i) = \lambda p_{baseline}(l_i | x, x_i) + (1 - \lambda) p_{kNN}(l_i | x, x_i)$$

# RQ1: Neighbours at inference

- Interpolation with Single Prototype
  - Instead of storing all training instances, store one prototype for each label
    - Captures essential semantics of various sentences under rhetorical role,
  - Average of the sentences representations with same rhetorical role as prototype
    - Geometrically, center of clusters for different labels
  - Interpolation same as KNN but with all prototypes

- Interpolation with Multiple Prototypes
  - Use multiple prototypes for each label.
    - Instances with same rhetorical role can exhibit distinct variations, resulting in diverse representations scattered across the embedding space.
    - Averaging into a single prototype might diminish specificity.
    - We cluster the instances belonging to each rhetorical role using K-means, and select multiple prototypes for each label from k centroids.

# RQ1: Neighbours at inference

| | Build | | Paheli | | M-CL | | M-IT | |
|---|---|---|---|---|---|---|---|---|
| | mac.F1 | mic.F1 | mac.F1 | mic.F1 | mac.F1 | mic.F1 | mac.F1 | mic.F1 |
| **Baseline** | 60.20 | 79.13 | 62.43 | 66.02 | 59.51 | 67.04 | 70.76 | 70.50 |
| **+ KNN** | 62.92 | 81.04 | 66.53 | 70.82 | 63.14 | 73.02 | 72.16 | 71.62 |
| **+ Single Proto** | 61.23 | 80.12 | 62.43 | 66.02 | 61.42 | 71.64 | 71.97 | 71.08 |
| **+ Mutli Proto** | 63.23 | 81.96 | 65.36 | 70.02 | 62.73 | 72.78 | 72.82 | 72.46 |

- Interpolation using training examples during inference boost the performance, in Macro-F1.
- Single prototype struggle to capture the diverse aspects within each rhetorical role
- Multiple prototype can act as smoothing effect that reduces noise or human label variations in the kNN-based approach,

# RQ2: Neighbours during training

- Contrastive learning:
  - Bring an anchor point closer to related samples while pushing it away from unrelated samples in embedding space.
  - Samples with the same/different labels are considered related/unrelated with respect to an anchor

$$L^{cont=-\frac{1}{N^2}\sum_{i,j}} \frac{\exp(\delta(c_i, c_j)d(c_i, c_j))}{\sum_{j'} \exp(1-\delta(c_i, c_{j'}))d(c_i, c_{j'})}$$

$$d(c_i, c_j) = \frac{1}{(1+\exp(\frac{c_i}{|c_i|}\frac{c_j}{|c_j|}))}$$

  - Lengthy legal documents limits batch size, so lack enough positive samples for the minority class instances
  - We use memory bank (Wu et al., 2018) - progressively reuse encoded representations from previous batches to into  fixed-size queue for each rhetorical role
    - We use instances from memory as well to compute the contrastive loss

# RQ2: Neighbours during training

- Discourse-aware Contrastive learning:

- Sentences in close proximity within a document, sharing the same label, should exhibit a stronger proximity compared to sentences with the same label but positioned farther apart in the document.

- Introduce a penalty inversely proportional to the absolute difference in their positions.
  - Higher penalty on positive sentence pairs that are closer in the document, encouraging them to be closer in the embedding space

$$L^{cont} = -\frac{1}{N^2} \sum_{i,j} \frac{\exp(\beta(i,j)\delta(c_i, c_j)d(c_i, c_j))}{\sum_{j'} \exp(1 - \beta(i,j)\delta(c_i, c_{j'}))d(c_i, c_{j'})}$$

$$\beta(i,j) \propto \frac{1}{|j - i|}$$

# RQ2: Neighbours during training

- Single Prototypical Learning

- Randomly initialize one prototype for each label and get learnt during fine-tuning

- Prototype centric view (pcv)
  - bring samples belonging to label  closer to the corresponding prototype , pushing away samples of other labels from this prototype.

$$L_j^{pcv} = -\frac{1}{N}(\sum_{c_p \in S_j} \log(d(z_j, c_p)) + \sum_{c_i \in S'_j} \log(1 - d(z_j, c_i)))$$

- Sample centric view (scv)
  - Sample brought closer to its prototype, while pushing away from other prototypes

$$L_j^{scv} = -\frac{1}{K}(\log(d(z_j, c_j)) + \sum_{z_p \in Z'_j} \log(1 - d(z_p, c_j)))$$

# RQ2: Neighbours during training

- Multiple Prototypical Learning

- Set of M prototypes per label is randomly initialized and a diversity loss is used to penalize prototypes of the same label if they are too similar to each other.

$$L_k^{div} = \sum_{\substack{q \neq r \\ z_q, z_r \in Z_k}} \max(0, z_q \cdot z_r - \theta)$$

- Sample Centric View is modified to ensure that each sample is in close proximity to at least one prototype among all the prototypes of the same class.

$$L_j^{scv} = -\min_{z_q \in Z_k} \log(d(z_q, c_j) + \frac{1}{(k-1)M} \sum_{z_p \in Z'_k} \log(1 - d(z_p, c_j))$$

# RQ2: Neighbours during training

| | Build | | Paheli | | M-CL | | M-IT | |
|---|---|---|---|---|---|---|---|---|
| | mac.F1 | mic.F1 | mac.F1 | mic.F1 | mac.F1 | mic.F1 | mac.F1 | mic.F1 |
| **Baseline** | 60.20 | 79.13 | 62.43 | 66.02 | 59.51 | 67.04 | 70.76 | 70.50 |
| **+ Contrastive** | 64.55 | 83.54 | 68.06 | 71.91 | 62.24 | 72.42 | 73.41 | 73.53 |
| **+ Contrastive + MB** | 66.51 | 83.29 | 71.76 | 72.69 | 63.14 | 72.72 | 72.22 | 72.46 |
| **+ Disc. Contr.** | 66.37 | 83.81 | 71.99 | 73.85 | 66.94 | 73.02 | 72.23 | 74.01 |
| **+ Disc. Contr. + MB** | 66.48 | 83.67 | 71.19 | 73.28 | 64.72 | 72.36 | 72.85 | 73.05 |

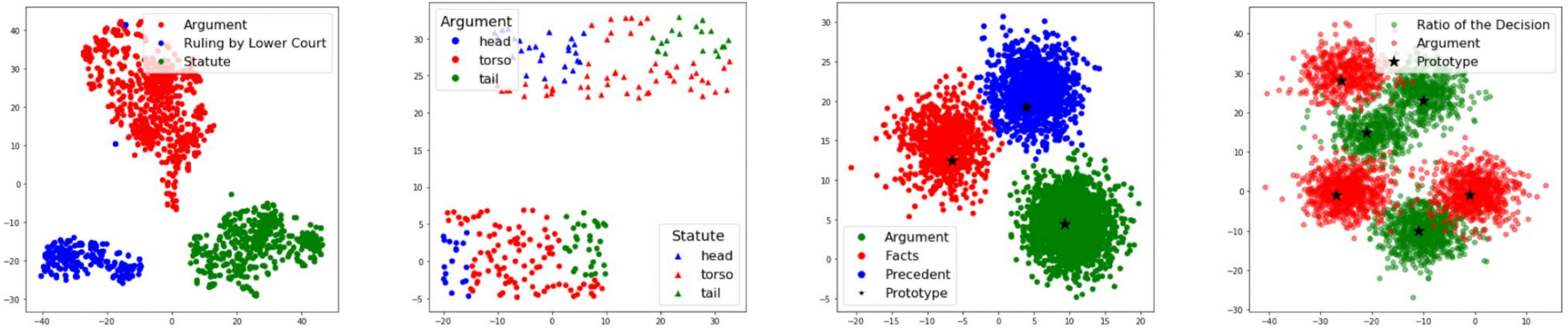
- Contrastive loss improves performance, further improved with discourse-aware loss
- Augmenting with a memory bank further enhances performance, in macro-F1, benefiting sparse classes

# RQ2: Neighbours during training

| | Build | | Paheli | | M-CL | | M-IT | |
|---|---|---|---|---|---|---|---|---|
| | mac.F1 | mic.F1 | mac.F1 | mic.F1 | mac.F1 | mic.F1 | mac.F1 | mic.F1 |
| **Baseline** | 60.20 | 79.13 | 62.43 | 66.02 | 59.51 | 67.04 | 70.76 | 70.50 |
| **+ Contrastive** | 64.55 | 83.54 | 68.06 | 71.91 | 62.24 | 72.42 | 73.41 | 73.53 |
| **+ Contrastive + MB** | 66.51 | 83.29 | 71.76 | 72.69 | 63.14 | 72.72 | 72.22 | 72.46 |
| **+ Disc. Contr.** | 66.37 | 83.81 | 71.99 | 73.85 | 66.94 | 73.02 | 72.23 | 74.01 |
| **+ Disc. Contr. + MB** | 66.48 | 83.67 | 71.19 | 73.28 | 64.72 | 72.36 | 72.85 | 73.05 |
| **+ Single Proto.** | 66.01 | 81.45 | 69.94 | 71.09 | 64.42 | 71.52 | 72.59 | 71.98 |
| **+ Multi Proto.** | 66.35 | 83.05 | 71.38 | 72.92 | 65.91 | 73.57 | 73.02 | 74.13 |
| **+ Disc. Contr. + Single Proto.** | 67.02 | 83.91 | 74.28 | 73.86 | 65.87 | 72.12 | 72.50 | 72.1 |
| **+ Disc. Contr. + Multi Proto.** | 67.21 | 83.65 | 75.52 | 76.34 | 68.66 | 74.59 | 73.14 | 72.22 |

- Prototypical learning improves over contrastive learning.
- Combining both prototypical and contrastive boosts performance.

# RQ2: Neighbours during training




(a) Contrastive (b) Disc.-aware Contr. (c) Single-Prototypical (d) Multi-prototypical

Figure 2: t-SNE visualizations of different models on M-CL dataset. Disc.: Discourse, Contr.: Contrastive. head, torso and tail in Disc.-aware Contr. plot indicate the relative position of the sentence in a document.

# RQ3: Cross-domain generalizability

| Train ↓ | Test → | Paheli | M-CL | M-IT |
|---|---|---|---|---|
| | **Random** | 19.10 | 7.87 | 9.12 |
| **Paheli** | Baseline | 62.43 | 56.98 | 57.31 |
| | Disc. Contr. | 71.99 | 56.54 | 57.40 |
| | Single Proto. | 69.94 | 58.30 | 59.92 |
| | Multi Proto. | 71.38 | 57.47 | 59.48 |
| | DC + Single Pr | 74.28 | 62.27 | 60.33 |
| | DC + Multi Pr | 75.52 | 60.89 | 60.61 |

- Baseline model shows an ability to transfer knowledge from one domain to another, outperforming random[1] guessing
- Discourse-aware contrastive model improves in-domain performance, it marginally reduces cross-domain performance
- Prototypical learning acts as a more robust guiding point, preventing overfitting to noisy neighbors as in contrastive models improving cross-domain transfer

# Conclusions

- Enhanced the performance of RRL by leveraging knowledge from neighbours, semantically similar instances
- Interpolation with kNN and multiple prototypes at the inference time shown promising improvements
    - especially in addressing the challenging issue of label imbalance, without requiring re-training.
- Incorporating neighbourhood constraints during training with our proposed discourse-aware contrastive learning and prototypical learning has demonstrated improvements.
- Prototypical methods proven to be robust, showcasing performance gains even in cross-domain scenarios, generalizing beyond the training domains