DISAGREEMENT AND CONTRADICTION DETECTION IN PEER REVIEWS

Nisarg Ganatra, Harsh Raval, Shubham Shah
Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT)
Gandhinagar, India
{202311018, 202311028, 202311049}@daiict.ac.in

1 Introduction

In academic publishing, peer review is a vital step in ensuring the quality and credibility of research. However, variations in reviewers' feedback can lead to conflicts, ranging from general differences of opinion to direct contradictions in assessments. Detecting and addressing these conflicts not only streamlines the review process but also improves transparency and fairness for authors and editors. This study aims to develop an automated model that identifies the following types of review conflicts:

- **Disagreement with Contradiction:** This occurs when one reviewer explicitly refutes or denies another's statements, creating a direct opposition that cannot logically coexist. For instance, if one reviewer states that a method is reliable while another claims it is flawed, the perspectives are fundamentally incompatible.

- **Disagreement without Contradiction:** This involves differing viewpoints without direct opposition. Reviewers may prioritize different aspects, such as methodological rigor versus practical application, without entirely rejecting each other's statements. This type of disagreement allows for complementary perspectives.

2 Related Work

In the context of peer review and rebuttals as in DISAPERE , contradiction is formally captured in rebuttal actions such as "reject criticism" or "contradict assertion." This represents a clear, explicit stance against the reviewer's argument . Examples of rebuttal actions that fall under contradiction in the paper include:

- **Reject Criticism:** The author denies the validity of a negative review statement.

- **Contradict Assertion:** The author refutes a factual assertion made by the reviewer.

Disagreement, in contrast, can involve softer rebuttal actions, such as mitigating criticism, where the author accepts some aspects of the reviewer's critique but argues that those aspects are not central to the overall argument.

In the MReD dataset, sentences are annotated with categories like strength, weakness, and AC disagreement. Understanding the difference between contradiction and disagreement is key for annotators, especially when labeling sentences like AC disagreement (when the area chair disagrees with the reviewers) .

- **Contradiction:** Contradiction would be reflected in AC disagreement if the area chair completely disputes the truth of what the reviewer said. For example, if a reviewer says, "The methodology is flawed," and the area chair says, "The methodology is flawless," these statements would directly oppose each other, making it a contradiction.

- **Disagreement:** Disagreement occurs when the area chair has a different opinion or interpretation but doesn't completely negate the reviewer's point. For example, if the reviewer says, "The methodology is unclear," and the area chair says, "While the methodology could be clearer, it is still valid," this would be a disagreement, not a contradiction.

In ArgSciChat, contradictions arise when one participant challenges the other's interpretation of facts (exploratory content) or rationale provided from the same paper in a way that makes both claims incompatible.

- **Contradiction:** If one participant is citing specific parts of the paper (rationales) to support a claim, and the other participant directly

challenges that with an opposing factual claim grounded in the same paper, a contradiction is present.

- **Disagreement:** Disagreement, in the context of ArgSciChat, happens when participants express different opinions or interpretations of the paper's content without necessarily opposing the facts. This typically occurs in the argumentative (ARG) phase of the dialogue, where scientists discuss their opinions about the methods, findings, or implications of the research. Disagreement here is not about whether the facts are true or false, but rather about how those facts should be understood, interpreted, or applied.

3 Problem Formulation

Given a set of manuscripts $M = \{m_1, m_2, \ldots, m_n\}$ and a set of reviewers $R = \{r_1, r_2, \ldots, r_m\}$, where each manuscript m is assigned to r reviewers, the goal is to design and train a novel model capable of detecting conflicts in the reviews for each manuscript. Specifically, the model should be able to identify two types of conflicts among the reviews:

- **Disagreement with Contradiction (Dc):**
- **Disagreement without Contradiction (Dn):**

Given this, the model must provide clear classifications of reviewer feedback into the defined categories of conflict, facilitating better editorial decisions and improving the overall peer review process.

4 Detecting Contradictions

Detecting contradictions in peer reviews is crucial for understanding the conflicting perspectives of reviewers. The model utilizes natural language processing techniques to analyze sentences within the reviews and identify instances where contradictions occur, particularly through negation, antonyms, and syntactic similarities. This helps in clarifying the areas of contention that need further editorial consideration.

And now this task of detecting contradictions in peer reviews can be framed as a conditional language-modeling problem, where the model predicts the likelihood of a contradiction given a pair of sentences from peer reviews. The objective is to train a model to generate outputs that reflect the likelihood of contradiction based on contextual cues within the sentences.

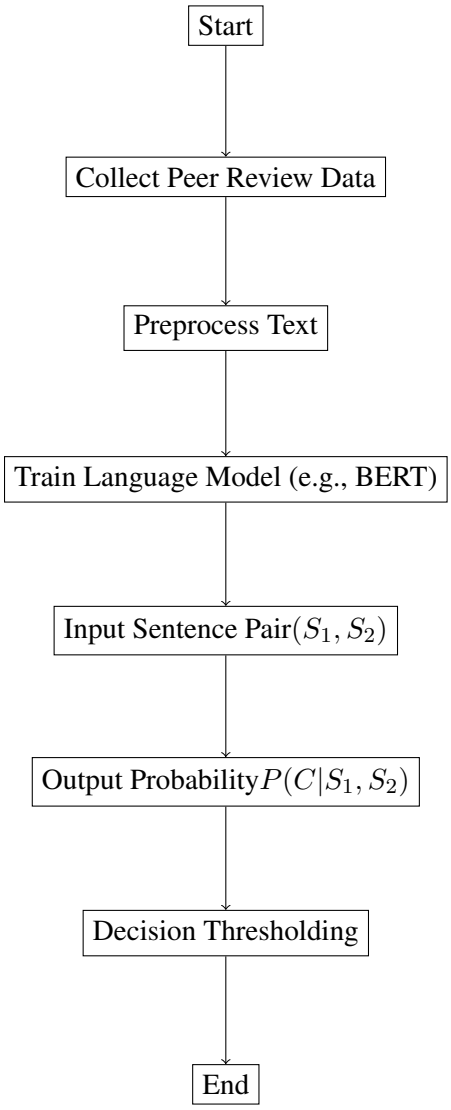

Figure 1: Pipeline for Detecting Contradictions in Peer Reviews

Algorithm 1 Detect Contradictions

```
Function detectContradictions(paragraph)
    sentences ← tokenize(paragraph)
    sentence_vector_list ← []
for each sentence in sentences do
        cleaned_words ← removeStopwords(lemmatize(tokenize(sentence)))
        sentence_vector ← computeTFIDF(cleaned_words)
        append sentence_vector to sentence_vector_list
end for
    contradictions ← []
for i ← 0 to length(sentence_vector_list) - 1 do
    for j ← i+1 to length(sentence_vector_list) do
        if computeCosineSimilarity(sentence_vector_list[i], sentence_vector_list[j]) > threshold then
            if detectsNegation(sentences[i], sentences[j]) OR detectsAntonym(sentences[i], sentences[j]) OR syntacticStructuresMatch(sentences[i], sentences[j]) then
                contradictions.append((sentences[i], sentences[j]))
            end if
        end if
    end for
end for
return contradictions
End Function
Function tokenize(text)
return list of tokens
Function lemmatize(words)
return lemmatized list of words
Function removeStopwords(words)
return cleaned list of words
Function computeTFIDF(words)
return TF-IDF vector
Function computeCosineSimilarity(vector1, vector2)
return similarity score
Function detectsNegation(sentence1, sentence2)
return true if negation found
Function detectsAntonym(sentence1, sentence2)
return true if antonym found
Function syntacticStructuresMatch(sentence1, sentence2)
return true if structures match =0
```

5 Dataset

The MReD (Meta-Review Dataset) is a specialized dataset designed to support structure-controllable text generation in the domain of peer reviews. Collected from the openreview system of the International Conference on Learning Representations (ICLR), MReD comprises **7,089** meta-reviews spanning from **2018 to 2021**. Each meta-review encapsulates the summarized opinions of multiple reviewers along with the area chair's final decision, making this dataset a rich resource for studying and generating structured summaries within the scientific review process.

Each meta-review in MReD is segmented and annotated at the sentence level across **nine predefined categories**:

- Abstract
- Strength
- Weakness
- Rating Summary
- Area Chair Disagreement
- Rebuttal Process
- Suggestion
- Decision
- Miscellaneous

For instance, a typical meta-review begins with an "abstract" summarizing the main contributions of the reviewed paper, followed by strengths and weaknesses identified by reviewers, and concludes with the final decision. This granular annotation enables models to generate text based not only on content but also on specific organizational structures by applying control signals that mimic human writing patterns in meta-reviews.

In terms of data composition, MReD contains a total of **45,929** annotated sentences. The dataset captures key aspects of the peer review process, allowing models to learn from structured input to generate varied outputs according to different structures. For example, outputs can emphasize strengths and abstract summaries in an "accept" decision or focus on weaknesses and suggestions in a "reject" scenario. MReD is thus positioned as a valuable resource for exploring advanced text generation that is both content-driven and structurally controlled, with applications extending

beyond meta-review generation to broader scientific summarization tasks.

5.1 Data Augmentation

Data augmentation techniques will be employed to enhance the dataset and improve the model's performance:

- **Synthetic Contradiction Pairs:** Create additional examples by pairing statements labeled as weakness with those labeled as strength or rating summary, and modify statements to create contradictions.
- **Paraphrasing:** Use paraphrasing techniques on weakness and area chair disagreement statements to increase dataset variety and allow the model to handle linguistic variations.
- **Antonym and Negation Injection:** Introduce antonyms or negation phrases to MReD's existing weakness or disagreement statements to create more nuanced disagreement and contradiction scenarios, strengthening the model's generalization abilities.

5.2 Evaluation Measures

The following evaluation metrics will be used:

- **Accuracy:** Measures the overall correctness of predictions across all disagreement types.
- **Precision, Recall, and F1 Score:** These metrics are particularly useful for binary classifications (contradiction vs. non-contradiction):
  - **Precision (Contradiction):** Measures how accurately the model identifies contradictions among all contradiction predictions.
  - **Recall (Contradiction):** Reflects how well the model detects actual contradictions.
  - **F1 Score:** Balances precision and recall for a clearer view of model performance, particularly important when data is imbalanced.
- **Confusion Matrix:** Provides an in-depth look at classification outcomes, especially useful to understand cases of false positives (disagreement misclassified as contradiction) and false negatives (missed contradictions).
- **AUC-ROC Curve:** Evaluates the model's ability to distinguish between disagreements with and without contradiction by analyzing true and false positive rates across various thresholds.

5.3 Experiments to be Conducted

Key experiments will include:

- **Baseline Comparisons:** Implement and compare classical NLP methods for contradiction detection using rules and feature extraction (e.g., antonym detection, cosine similarity on TF-IDF vectors).
- **Transformer-Based Models:** Fine-tune BERT-based models on the MReD dataset to analyze model performance when using contextualized embeddings for contradiction detection.
- **Conditional Language Modeling:** Frame the task as a conditional language-modeling problem and train a model to conditionally predict contradictions based on review sentences paired with control signals (e.g., "strength," "weakness").
- **Impact of Augmentation:** Evaluate the effectiveness of the data augmentation techniques (paraphrasing, antonym/negation injection) to see if they improve model robustness and generalization in contradiction detection.
- **Threshold Optimization:** Test different similarity thresholds for detecting contradictions to optimize the decision boundary in cases where contradictions are subtle.

5.4 Ablation Studies

To identify essential components, the following ablation studies are suggested:

- **Feature Removal Ablation:** Remove specific features, such as antonym detection, negation identification, and syntactic structure matching, to determine each component's impact on performance.

Experiment/Method	Accuracy	Precision	Recall	F1 Score	AUC-ROC
Rule-Based (TF-IDF + Similarity)	...	...	...	...	...
BERT	...	...	...	...	...
RoBERTa	...	...	...	...	...
BERT + Data Augmentation	...	...	...	...	...
Conditional Language Model	...	...	...	...	...

Table 1: Performance of Contradiction Detection Methods on MReD Dataset

- **Control Sequence Variation:** Test how varying control sequences (e.g., weakness paired with rating summary vs. weakness paired with strength) affects the model's ability to distinguish contradictions from disagreements without contradiction.
- **Augmentation Impact Analysis:** Measure the effect of each augmentation technique (e.g., synthetic contradiction pairs, paraphrasing) on model performance to understand the benefit of each data augmentation method.
- **Cosine Similarity Thresholding:** Perform sensitivity analysis on the threshold used in cosine similarity to evaluate the robustness of the model's contradiction detection capabilities under different levels of similarity filtering.

6 Methodology

6.1 Model Architecture

The proposed model architecture for disagreement and contradiction detection consists of several layers, as shown in Figure 2:

- **Preprocessing and Embedding Layer:** This layer performs tokenization and applies pre-trained embeddings (e.g., BERT, RoBERTa) to transform raw text into contextual embeddings.
- **Document Encoding Layer:** Produces contextual embeddings for each review using the pre-trained model, which are fine-tuned during training.
- **Graph Construction Layer:** Constructs an adjacency matrix A based on similarity or heuristic rules, and assigns edge weights. The graph is used to model relationships between reviews.
- **Graph Neural Network (GNN) Layer:** Uses message-passing mechanisms (e.g., GCN, GAT) to update node embeddings, learning inter-review relationships. Learnable weights $W_{\text{gnn}}^{(k)}$ are applied in each layer.
- **Aggregation Layer:** Aggregates graph-level information using attention mechanisms or pooling. The result is a single graph-level embedding h_{graph}.
- **Decoder Layer:** A Transformer-based decoder generates the meta-review using the graph embedding as input.
- **Output Layer:** Produces token probabilities for the generated meta-review sequence.

The model is optimized to identify disagreement and contradiction among reviewer comments.

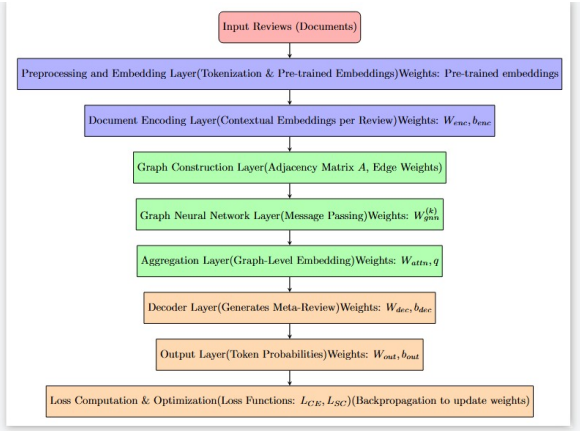

Figure 2: Model Architecture for Disagreement and Contradiction Detection

6.2 Training Setup

The training process involves several components. First, in **data preparation**, reviews $\{R_1, R_2, \ldots, R_N\}$ for each target meta-review are tokenized using a BERT tokenizer,

converted into input IDs and attention masks, and then dynamically batched with padding. Adjacency matrices are also constructed dynamically for each review set.

Next, during **model initialization**, pre-trained embeddings from models like BERT or RoBERTa are used, with optional fine-tuning. Graph Neural Network (GNN) layers such as GCN or GAT are initialized with learnable weights $W_{\text{gnn}}(k), b_{\text{gnn}}(k)$. A Transformer decoder, with weights $W_{\text{dec}}, b_{\text{dec}}$ and positional encodings, is used for sequence generation. The output projection layer $W_{\text{out}}, b_{\text{out}}$ finalizes predictions. Optimization employs Adam or AdamW, along with a learning rate scheduler.

The **training loop** starts with a forward pass, where tokenized reviews are passed through embedding layers to obtain contextual embeddings H_i. A graph is constructed using similarity-based adjacency matrices, and GNN layers update node embeddings. A graph-level embedding h_{graph} is computed using attention or pooling, feeding into the decoder to generate token probabilities. Loss $L = L_{\text{CE}} + \lambda L_{\text{SC}}$ is computed, where L_{CE} is the cross-entropy loss, and L_{SC} (optional) ensures structural consistency or addresses key disagreements.

In the **backward pass**, gradients are computed and weights are updated using the optimizer. During **evaluation**, validation loss is calculated, and metrics like accuracy, precision, recall, and F1-score, as well as human evaluations, are used to assess model performance.

6.3 Loss Function

The total loss function combines two components:

1. **Cross-Entropy Loss** (L_{CE}): Ensures alignment between generated meta-reviews and the target summaries.

$$L_{\text{CE}} = -\frac{1}{T}\sum_{t=1}^{T} \log P(y_t^* \mid y_{<t}^*, h_{\text{graph}})$$

2. **Structure Consistency Loss** (L_{SC}): Encourages adherence to structural formats and accurate detection of disagreements/contradictions.

$$L_{\text{SC}} = \sum_{i=1}^{N_s} \omega_i \cdot \delta(s_i, \hat{s}_i)$$

The final loss is:

$$L = L_{\text{CE}} + \lambda L_{\text{SC}}$$

where λ controls the influence of the structural consistency loss.

7 Experimental Results

The following table presents the summarization quality of various models evaluated on the MReD dataset, using ROUGE metrics (R1, R2, RL), which measure the overlap and quality of generated text summaries compared to reference summaries.

Model	R1	R2	RL
Source Generic	27.58	3.97	14.14
Target Generic	27.98	5.52	15.01
MMR, Unctrl	31.43	5.45	16.31
LexRank, Unctrl	31.74	6.67	16.71
TextRank, Unctrl	32.72	7.37	17.25
BART Large, Unctrl	33.31	8.63	19.67
BART Large, Sen-Ctrl	38.73	10.82	23.05

Table 2: Summarization Quality of Different Models on MReD Dataset (ROUGE Metrics)

References

[1] DISAPERE: A Dataset for Discourse Structure in Peer Review Discussions.

[2] MReD: A Meta-Review Dataset for Structure-Controllable Text Generation.

[3] A Dataset of Argumentative Dialogues on Scientific Papers.