

PeerArg: Argumentative Peer Review with LLMs

Purin Sukpanichnant, Anna Rapberger, Francesca Toni

Imperial College London, United Kingdom

{ps1620, a.rapberger, ft}@imperial.ac.uk

Abstract

Peer review is an essential process to determine the quality of papers submitted to scientific conferences or journals. However, it is subjective and prone to biases. Several studies have been conducted to apply techniques from NLP to support peer review, but they are based on black-box techniques and their outputs are difficult to interpret and trust. In this paper, we propose a novel pipeline to support and understand the reviewing and decision-making processes of peer review: the PeerArg system combining LLMs with methods from knowledge representation. PeerArg takes in input a set of reviews for a paper and outputs the paper acceptance prediction. We evaluate the performance of the PeerArg pipeline on three different datasets, in comparison with a novel end-2-end LLM that uses few-shot learning to predict paper acceptance given reviews. The results indicate that the end-2-end LLM is capable of predicting paper acceptance from reviews, but a variant of the PeerArg pipeline outperforms this LLM.

1 Introduction

Peer review is a process where work is examined and evaluated by a group of people with expertise in the relevant field. The process is crucial to ensure the quality of the work. It has been adopted by many conferences and journals, to ensure the published papers are of adequate quality. Peer review is hence a core component of the progress in several academic areas. Nevertheless, fairness is the main weakness of the process. Peer review involves lots of discussion and evaluation from human reviewers, which are subjective and are prone to biases and irrationality. For example, the reviewers may be more likely to accept a paper whose results agree with what they believe, known as confirmation bias (Mahoney, 1977). Another bias is first-impression bias, where the initial impressions of the document (e.g. typographical layout (Moys, 2017)) affect the entire judgement.

There has been an emerging trend to study how to apply techniques from Natural Language Processing (NLP) to improve the peer review process. Zhou et al. (2024) study review generation and found that the generated reviews consider more aspects of a paper than human reviewers. Another example is given by Nuijten and Polanin (2020), with a tool that checks for statistical inconsistencies in papers. Review understanding is another application of NLP in peer review. This could lessen the burden for meta-reviewers or conference chairs who read the reviews before deciding whether to accept a paper or not. For example, Kumar

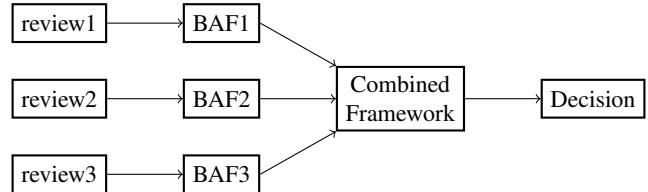


Figure 1: Overview of the PeerArg pipeline. Firstly, a bipolar argumentation framework BAF_i is extracted from review i . Then, the frameworks are combined. The final decision is drawn from the combined framework.

et al. (2023) propose methods to generate pros and cons summary given reviews. Bhatia et al. (2020) present a system to generate a meta-review from reviews. Most of the studies for review understanding used deep learning models as their backbone. Even though they have impressive performances, the black-box nature of the models makes it difficult to understand the rationale behind the results and trust the models.

In this paper, we propose a new technique to address these shortcomings. We use symbolic AI methods, specifically methods from computational argumentation (Dung, 1995; Amgoud et al., 2008; Baroni et al., 2018), to enhance review understanding and to assess the alignment between paper acceptance decision-making process and the reviews. Computational argumentation studies how to model the way humans build and use arguments so that it can be represented in a machine (Atkinson et al., 2017). Whereas existing argumentation studies in peer review focus on the structure of arguments (Hua et al., 2019; Fromm et al., 2020), our work uses a form of bipolar argumentation frameworks (Amgoud et al., 2008), adding a support relation to abstract argumentation (Dung, 1995), to model both the reviews and the review aggregation process. Our main contribution is *PeerArg*, a novel framework for transparent review aggregation. PeerArg predicts the acceptance status of the paper, as illustrated in Figure 1.

As part of this paper, we furthermore propose a few-shot learning end-2-end LLM taking in reviews of a paper and predict the paper acceptance, inspired from Zhou et al. (2024) and Gorur et al. (2024). We evaluate PeerArg and the end-2-end LLM on three review datasets: two conference review datasets, and a journal review dataset. Our empirical studies indicate that enhancing LLMs with methods from computational argumentation has beneficial effects. The results show that the LLM can be used to predict the paper acceptance from reviews, and with a certain hyperparameters combination, the PeerArg pipeline outperforms the LLM in all the datasets.

2 Related Work

Methods for improving the interpretation and understanding of reviews using AI have received significant attention in recent years.

Understanding Reviews with AI Several components of peer review have the potential to be improved using AI, such as pre-review screening, matching papers with reviewers, or review generation. One direction to enhance the reviewing process is review summarisation. Kumar et al. (2023) propose methods to generate a pros and cons summary of given reviews of a paper. Bhatia et al. (2023) propose a tool called *MetaGen* to generate a meta-review from given reviews.

Conferences give specific guidelines concerning with aspects they expect the reviewers to consider. Several studies have been done around accurately predicting and understanding these scores as this enhances the trustworthiness of the reviews. An example is PeerRead (Kang et al., 2018b), the first public dataset of scientific peer reviews with corresponding decisions and aspect scores, i.e., the ratings the reviewers gave for each of the aspects. This dataset was used in Li et al. (2020) to evaluate the proposed multi-task framework for peer-review aspect score predictions. Chakraborty et al. (2020) performed an aspect-based sentiment analysis and determined that there has been a correlation between the distribution of aspect-based sentiments and the acceptance decision of papers. Recent studies are shifting to the use of a Large Language Model (LLM). Zhou et al. (2024) evaluated two LLMs, GPT-3.5 and GPT-4, on the review scores prediction and review generation tasks. The results indicated that LLMs can infer aspect scores given a review; however, their performances were inadequate when given a paper.

Argumentation in Peer Review Methods from computational argumentation have rarely been applied in peer review applications. Notable exceptions are the works by Hua et al. (2019) and Fromm et al. (2020). However, in contrast to our work, neither consider the relations between arguments.

Hua et al. (2019) applied argumentation to understand the content and structure of peer reviews. They detect argumentative text in a review and classify it into one of the following types: evaluation, request, fact, reference, or quote. The authors then analysed and compared argumentation in reviews across several ML and NLP conferences. The results show that there have been some discrepancies in the argumentation trend across different conferences. For example, ACL and NeurIPS tend to contain most arguments, strong reject/accept reviews tend to have fewer arguments.

Another proposal to was made in Fromm et al. (2020). The authors aimed to extract the most relevant arguments from a review and evaluated its effect towards the paper acceptance decision. The experiment empirically indicated that correct decisions can be made by using merely half of the review.

3 Preliminaries

We recall bipolar argumentation (Amgoud et al., 2008) and few-shot learning for LLMs (Brown et al., 2020).

3.1 Bipolar Argumentation

In bipolar argumentation frameworks (Amgoud et al., 2008), arguments are abstract entities; relations between them are either supports or attacks. We define them as follows.

Definition 1. A bipolar argumentation framework (*BAF*) is a tuple $\langle X, Att, Supp \rangle$ where X is a finite set of arguments and $Att, Supp \subseteq X \times X$ are attack and support relations between arguments. An argument $a \in X$ attacks an argument $b \in X$ if and only if $(a, b) \in Att$. Similarly, an argument $a \in X$ supports an argument $b \in X$ if and only if $(a, b) \in Supp$.

We furthermore consider quantitative bipolar argumentation frameworks (*QBAF*) (Baroni et al., 2018).

Definition 2. A quantitative BAF (*QBAF*) is a tuple $\langle X, Att, Supp, \beta \rangle$ over range $D = [0, 1]$ where $\langle X, Att, Supp \rangle$ is a BAF and $\beta : X \rightarrow D$ is a total function that assigns a base score to each argument.

By $A(a) = \{b | (b, a) \in Att\}$ we denote the attackers, by $S(a) = \{b | (b, a) \in Supp\}$ the supporters of an argument a .

A semantics $\sigma_Q : X \rightarrow D$ for a QBAF Q determines the final strength of each argument. In this work, we use the DF-QuAD semantics (Rago et al., 2016) and MLP-based semantics (Potyka, 2021).

Definition 3. Let $Q = \langle X, Att, Supp, \beta \rangle$ be a QBAF over $D = [0, 1]$. Let $\delta : D^* \rightarrow D$ denote the strength aggregation function,¹ such that, for $T = (v_1, \dots, v_n) \in D^*$:

$$\begin{aligned} &\text{if } n = 0: \delta(T) = 0; \\ &\text{if } n = 1: \delta(T) = v_1; \\ &\text{if } n = 2: \delta(T) = f(v_1, v_2); \\ &\text{if } n > 2: \delta(T) = f(\delta(v_1, \dots, v_{n-1}), v_n) \end{aligned}$$

where, $f(x, y) = x + (1 - x) \cdot y = x + y - x \cdot y$, $x, y \in D$.

Let $\varphi : D \times D \times D \rightarrow D$ denote the influence function, where for $v_0, v_a, v_s \in D$:

$$\varphi(v_0, v_a, v_s) = \begin{cases} v_0 - v_0 \cdot |v_s - v_a| & \text{if } v_a \geq v_s \\ v_0 + (1 - v_0) \cdot |v_s - v_a| & \text{if } v_a < v_s \end{cases}$$

For any $a \in X$, the DF-QuAD semantics is defined by

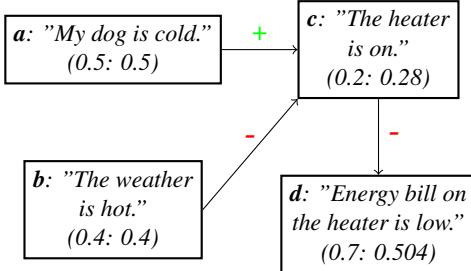
$$\sigma_{DF-QuAD}(a) = \varphi(\beta(a), \delta(\sigma(A(a))), \delta(\sigma(S(a))))$$

s.t. $\sigma(A(a)) = (\sigma(a_1), \dots, \sigma(a_n))$ where (a_1, \dots, a_n) is an arbitrary permutation of the $(n \geq 0)$ attackers in $A(a)$, and $\sigma(S(a)) = (\sigma(s_1), \dots, \sigma(s_m))$ where (s_1, \dots, s_m) is an arbitrary permutation of the $(m \geq 0)$ supporters in $S(a)$.

For each argument, DF-QuAD semantics aggregates the strengths of its attackers and supporters, and determines how both aggregates influence the base score of the argument.

¹Here, D^* is the set of all sequences of elements of D .

Example 1. An example of a QBAF is as illustrated below, with a set of arguments $X = \{a, b, c, d\}$, attack relation $Att = \{(b, c), (c, d)\}$, support relation $Supp = \{(a, c)\}$, base score function $\beta(a) = 0.5$, $\beta(b) = 0.4$, $\beta(c) = 0.2$, and $\beta(d) = 0.7$. The evaluation under DF-QuAD semantics yields the final strength function $\sigma(a) = 0.5$, $\sigma(b) = 0.4$, $\sigma(c) = 0.28$, and $\sigma(d) = 0.504$. The base score and the final strength of the arguments are written in brackets.



Definition 4. Let $Q = \langle X, Att, Supp, \beta \rangle$ be a QBAF over $D = [0, 1]$. The MLP-based semantics is defined by

$$\sigma_{MLP}(a) = \begin{cases} \lim_{k \rightarrow \infty} s_a^{(k)} & \text{if the limit exists} \\ \perp & \text{otherwise} \end{cases}$$

for any argument $a \in X$, where $s_a^{(k)}$ is defined using a two-step process (Mossakowski and Neuhaus, 2018). Starting with $s_a^{(0)} = \beta(a)$ (i.e. a base score of that argument), we then iterate over two steps:

$$\text{Aggregation: } \alpha_a^{(i+1)} = \sum_{(b,a) \in Supp} s_b^{(i)} - \sum_{(b,a) \in Att} s_b^{(i)}$$

$$\text{Influence: } s_a^{(i+1)} = \varphi(\varphi^{-1}(\beta(a)) + \alpha_a^{(i+1)})$$

where $\varphi : \mathbb{R} \rightarrow D$ is strictly monotonically increasing.

MLP-based semantics treats a QBAF as a multi-layer perceptron (MLP), where edges from attacking/supporting arguments have weights -1 and +1 respectively. Strengths are calculated similar to a forward pass in an MLP.

Example 2. Let us head back to the QBAF from Example 1. Assuming φ is the sigmoid function, the evaluation under MLP-based semantics yields the final strength function $\sigma(a) = 0.5$, $\sigma(b) = 0.4$, $\sigma(c) = 0.216$, and $\sigma(d) = 0.653$.

3.2 Few-Shot Learning for LLMs

Large language models (LLMs) are the pretrained transformer models (Vaswani et al., 2023) that take in a text input and generate the output text. LLMs are useful for multiple tasks such as machine translation, text summarisation, and text generation. Due to the extremely large number of parameters in the LLMs e.g. 175 billion parameters for GPT-3 (Brown et al., 2020), fine-tuning them for a particular task requires a large amount of computing resources and time.

Few-shot learning (Brown et al., 2020) is a way to resolve such constraints. The method works by providing the LLMs some examples of the task we expect them to perform, and let the LLMs

Review=> “\nSummary: \n- This paper proposes a hand-designed network architecture on a graph of object proposals to perform soft non-maximum suppression to get object count....” Review=> “Summary\n - This paper mainly focuses on a counting problem in visual question answering (VQA) using attention mechanism....” Review=> “This paper tackles the object counting problem in visual question answering. It is based on the two-stage method that object proposals are generated from the first stage with attention....” Decision=> accept	Review=> “* This paper models images with a latent code representation, and then tries to modify the latent code to minimize changes in image space....” Review=> “The main contribution of the paper is a method that provides visual explanations of classification decisions. The proposed method uses \n...” Review=> “The authors address two important issues: semi-supervised learning from relatively few labelled training examples in the presence of many unlabelled examples,...” Decision=> reject	Primer
	Review=> “first input review” Review=> “second input review” ... Review=> “n-th input review” Decision=>	Prompt

Figure 2: End-2-End LLM Input Template (the sample reviews are truncated due to lack of space)

learn to produce the result in a similar format as the provided examples for our inputs. This learning is done without fine-tuning, utilising the general knowledge the LLMs have acquired during their pre-training period. Examples are called *primer* and the input is called *prompt*, as in (Gorur et al., 2024).

4 Paper acceptance by end-to-end LLMs

In this section, we propose an LLM that applies the few-shot learning methodology to classify a paper acceptance given the reviews of the paper, called *end-2-end LLM*.²

The end-2-end LLM uses a quantised 4-bit pretrained Mistral-7B-0.1 from Mistral AI³ as a pretrained LLM. The template of an input prompt given to the LLM is inspired from (Gorur et al., 2024), consisting of a primer and a prompt. The primer consists of reviews of four papers each with the final decision, two accepts and two rejects, taken from (Ghosal et al., 2022a,c; Kang et al., 2018c,a). The prompt is a set of reviews of the paper we want to predict, with no labels.

²https://gitlab.doc.ic.ac.uk/ps1620/peerarg/-/tree/master/llm_e2e?ref_type=heads

³<https://huggingface.co/mistralai/Mistral-7B-v0.1>

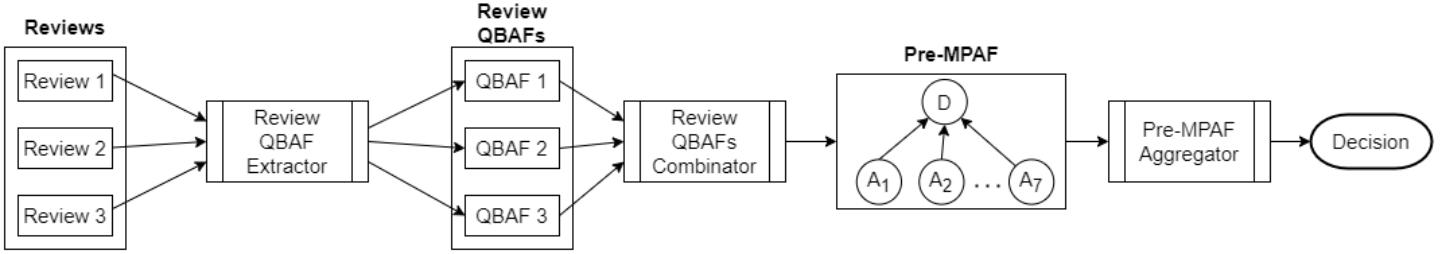


Figure 3: PeerArg pipeline diagram

The partial template for the primer & prompt input to LLM is shown in Figure 2 consisting of two reviews-decision samples. Each review starts with *Review==>* placeholder followed by the raw review in double quotation mark and a newline to separate each review. Each set of reviews is then followed by the decision which starts with *Decision==>* and the accept/reject decision. These samples are the primer of our input to the LLM.

The prompt of the LLM input is in a review-decision format similar to the samples in the primer. The only difference is that there is no decision attached to the *Decision==>* placeholder. This is to let the LLM predict the outcome.

5 Peer-Review Enhanced with KR

One downside of the end-2-end LLM is its black-box nature. The model only returns the final result without intermediary steps which makes it difficult to rationalise with. To resolve this issue, we incorporate knowledge representation into the process. In this section, we present the PeerArg pipeline⁴. Given the reviews of a paper, the pipeline represents each of them as an argumentation framework, combining these frameworks into a single framework, and aggregating to determine the paper acceptance.

The pipeline diagram is as illustrated in Figure 3, assuming that there are three reviewers for a paper. The pipeline consists of three main steps: *Review QBAF extraction*, *Review QBAFs combination*, and *Pre-MPAF aggregation*.

5.1 Review QBAF Extractor

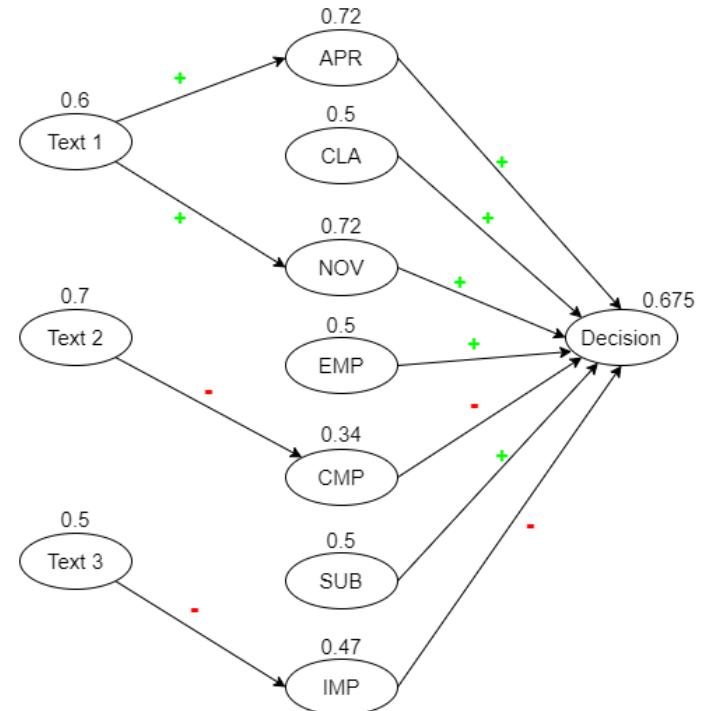
The first step is to extract an argumentation framework from each review. Reviewers are typically encouraged to assess the quality of a paper under various aspects, each of which may affect the decision more/less than others, and provide justification for their impression of the paper w.r.t. these aspects. Following this idea, we can treat aspects as arguments (called *aspect arguments*) that attack or support the decision (the so-called *decision argument*). Each sentence in a review is associated with different aspects, and so can be considered as an argument (called *text argument*). As a result, each review can be considered as a three-level QBAF consisting of text and aspect arguments, and decision argument.

There are several different aspects that conferences expect reviewers to consider. These vary for different conferences. In this

paper, we focus on the aspects provided by the ACL 2016 conference, namely *appropriateness*, *clarity*, *novelty*, *empirical* and *theoretical soundness*, *meaningful comparison*, *substance*, and *impact*. For our framework, we introduce an argument for each of these aspects; abbreviated by APR, CLA, NOV, EMP, CMP, SUB, and IMP, respectively. Below, we introduce our review QBAFs.

Definition 5. A review QBAF is a tuple $\langle X, \text{Att}, \text{Supp}, \beta \rangle$ over $D = [0, 1]$, where $X = T \cup A \cup \{\text{Decision}\}$ with T as the set of text arguments, $A = \{\text{APR}, \text{CLA}, \text{NOV}, \text{EMP}, \text{CMP}, \text{SUB}, \text{IMP}\}$ as the set of aspect arguments, and *Decision* as the decision argument. The relations $\text{Att}, \text{Supp} \subseteq (T \times A) \cup (A \times \text{Decision})$ are such that Att and Supp are disjoint. A semantics $\sigma : X \rightarrow D$ assigns a strength to each argument.

Example 3. An example of a review QBAF is given below.



This review QBAF has three text arguments, called Text 1, Text 2 and Text 3 which attack/support different aspect arguments; e.g., Text 1 supports APR (appropriateness) as well as NOV (novelty of the work). Each argument is annotated with the final strength, depicted above it. Edges with minus are attacks and edges with plus are supports.

⁴Repository: <https://gitlab.doc.ic.ac.uk/ps1620/peerarg>

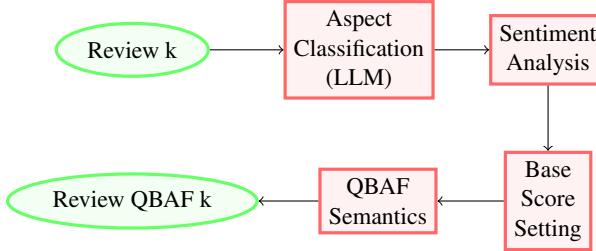


Figure 4: The process to obtain a review QBAF from a review.

To obtain a review QBAF from a review, there are four main steps, as illustrated in Figure 4. Initially, we start with **aspect classification** where each sentence in the review is classified which aspects it belongs to. In this paper, we use a few-shot learning LLM for aspect classification. The next step is **sentiment analysis** to determine if the sentence is positive/negative towards the aspects. A graph structure of the review QBAF is obtained at this step. Then, the third step is the **base score setting** where the base scores are set for all the arguments. Finally, **QBAF semantics** is applied to calculate the final strength of each argument.

Initially, all edges between aspect arguments and the decision argument are undecided, because each aspect may support or be against the paper being accepted, depending on its strength. Figure 5 illustrates the process to determine the relations of the undecided edges of the incomplete QBAF of a review, up to the calculation of the strength of the decision argument.

To determine the relations from the aspect arguments to the decision argument, we first apply a (gradual) semantics to calculate the strengths of the aspect arguments. These strengths indicate how supportive the aspects are towards paper acceptance. A strength of 0 means a paper is very poor on this aspect, in contrast, a strength of 1 means the paper is excellent on this aspect. We set 0.5 as a midpoint. An aspect argument with strength below 0.5 is attacking; otherwise, it is supporting the decision argument.

The further the strength is away from 0.5, the stronger attacker/supporter an aspect argument is. To incorporate this idea, we reflect the strength around 0.5 before scaling up by a factor of 2 so that the strength is still in the range [0, 1]. Formally, for an aspect argument a of strength s , we define

$$\beta(a) = 2 \cdot |s - 0.5|.$$

Once these strengths (towards the decision argument) of aspect arguments are calculated, the semantics is then applied to calculate the final strength of the decision argument.

We depict the original strengths of the aspect arguments inside nodes in the bottom-left box in Figure 5; the updated strengths are depicted next to the arguments (in red and green, respectively).

5.2 Review QBAFs Combinator

The next step is to combine our extracted review QBAFs. Review QBAFs share the same aspect arguments and decision argument, but may have different attacks, supports and text arguments. Once all the review QBAFs are extracted from the reviews, their text arguments are trimmed off. Below, we define a trimmed QBAF,

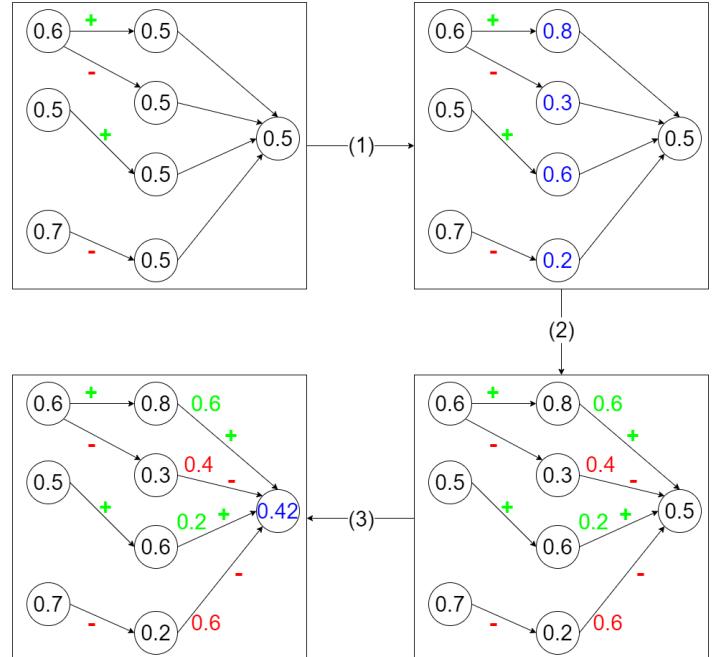


Figure 5: From an incomplete QBAF to a complete QBAF, with edges with minus and plus being attacks and supports respectively. (1) First, we apply QBAF semantics to get strengths for aspect arguments. (2) From the aspect arguments' strengths, we determine their relation to the decision argument; and calculate their scores (red and green numbers). (3) We apply QBAF semantics to get the final strength of the decision argument.

which simply removes the text arguments from the review QBAF.

Definition 6. Given a review QBAF $\langle X, Att, Supp, \beta \rangle$ under semantics σ where $X = T \cup A \cup \{\text{Decision}\}$, a trimmed review QBAF is $\langle X_{\text{trim}}, Att_{\text{trim}}, Supp_{\text{trim}}, \beta_{\text{trim}} \rangle$ where

$$X_{\text{trim}} = A \cup \{\text{Decision}\}$$

$$Att_{\text{trim}} = \{(b, \text{Decision}) | b \in A, (b, \text{Decision}) \in Att\}$$

$$Supp_{\text{trim}} = \{(b, \text{Decision}) | b \in A, (b, \text{Decision}) \in Supp\}$$

$$\beta_{\text{trim}} = \{(a, \beta(a)) | a \in A \cup \{\text{Decision}\}\}$$

A semantics of the trimmed review QBAF is a function $\sigma_{\text{trim}} = \{(a, \sigma(a)) | a \in A \cup \{\text{Decision}\}\}$.

In the remainder, we will refer to trimmed review QBAFs simply as review QBAFs. Next, we combine the frameworks.

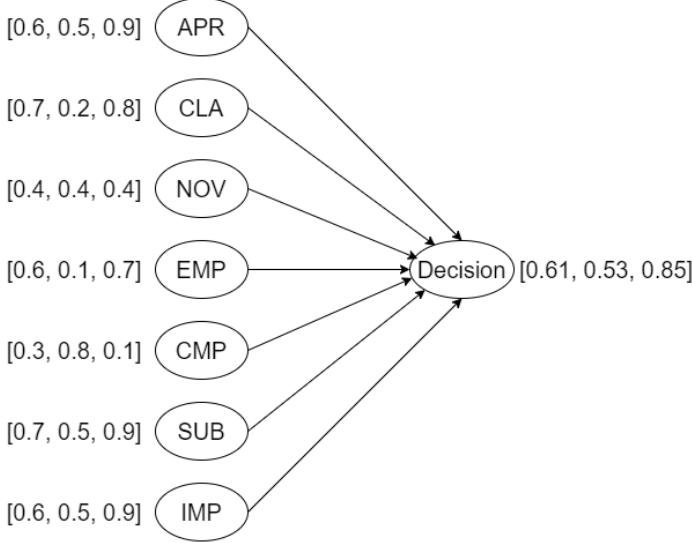
Definition 7. Given n review QBAFs Q_1, \dots, Q_n where $Q_i = \langle X, Att_i, Supp_i, \beta_i \rangle$ for each $i \leq n$, and semantics $\sigma_1, \dots, \sigma_n$. The pre-Multi-Party Argumentation Framework (pre-MPAF) is defined as $\langle X, Und, \beta_{\text{vec}} \rangle$ where

$$Und = \{(a, b) | (a, b) \in \bigcup_{i=1}^n (Att_i \cup Supp_i)\}$$

and $\beta_{\text{vec}} : X \rightarrow [0, 1]^n$ is a total function such that $\beta_{\text{vec}}(a) = [\sigma_1(a), \dots, \sigma_n(a)]$ for each argument $a \in X$.

The outcome of the semantics σ_i from the n review QBAFs is the base score of the pre-MPAF.

Example 4. An example of a pre-MPAF is illustrated below.



We have $\beta_{vec} = \{ (APR, [0.6, 0.5, 0.9]), (CLA, [0.7, 0.2, 0.8]), (NOV, [0.4, 0.4, 0.4]), (EMP, [0.6, 0.1, 0.7]), (CMP, [0.3, 0.8, 0.1]), (SUB, [0.7, 0.5, 0.9]), (IMP, [0.6, 0.5, 0.9]) \}$.

5.3 Pre-MPAF Aggregator

To obtain the final decision, we aggregate the information obtained from the pre-MPAFs. We implement two types of aggregation methods in PeerArg, as illustrated in Figure 6.

The first method (path (1), left, in Figure 6) aggregates the strength vectors of the pre-MPAFs and uses them to identify support and attack relations. The outcome is a QBAF, called *multi-party argumentation framework (MPAF)*, which is then used to determine the strength of the decision argument, based on DF-QuAD and MLP-based semantics. In the second method (path (2), right, in Figure 6), we focus on the strength of the decision argument. We apply a *decision strength interpretation* to convert a list of strengths of the decision argument into a list of decisions and aggregate them to return the final accept/reject decision. The final strength of the decision argument *Decision* is then used to determine the paper acceptance. We use a simple threshold such that the paper is predicted to be accepted only if the strength is more than 0.5; otherwise, rejected.

Aggregation with Argumentation (Path 1) We aggregate our pre-MPAF to obtain an MPAF by applying an aggregation function to identify a strength for each argument and to complete the attack and support relations. The process is illustrated in Figure 7. Given a pre-MPAF $\langle X, Und, \beta_{vec} \rangle$, we calculate the average of the sequence of strengths $\beta_{vec}(a)$ for all arguments $a \in X$ to determine the relations between aspect and decision argument(s).

Definition 8. Given a pre-MPAF $\langle X, Und, \beta_{vec} \rangle$. For each $a \in X$, we calculate the average

$$\gamma(a) = \frac{1}{|\beta_{vec}(a)|} \sum_{x \in \beta_{vec}(a)} x.$$

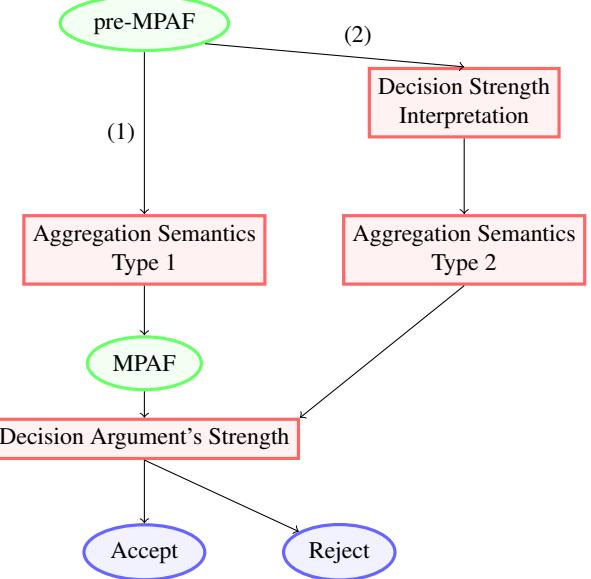


Figure 6: Pre-MPAF Aggregator diagram, with (1) and (2) as the first and the second implementation types.

We define our MPAF $\langle X, Att, Supp, \beta \rangle$ with

$$Att = \{(a, Decision) \in Und \mid \gamma(a) < 0.5\}$$

$$Supp = \{(a, Decision) \in Und \mid \gamma(a) \geq 0.5\}$$

$$\beta(a) = \begin{cases} \gamma(a) & \text{if } a = \text{Decision} \\ 2 \cdot |\gamma(a) - 0.5| & \text{otherwise} \end{cases}$$

We obtain the base score $\beta(a)$ of the arguments by averaging their strength vectors $\beta_{vec}(a)$ and recenter it around 0.5, similar as described in Section 5.1. The average $\gamma(a)$ furthermore determines the relation between the decision and the aspect arguments.

In the next step, the DF-QuAD semantics and the MLP-based semantics, respectively, can be applied to calculate the strength of the decision argument of the MPAF.

Similar to review QBAFs, we depict the initial strength of the aspect arguments of MPAFs in the nodes in Figure 7; the updated base scores are depicted next to the arguments.

Decision Vector Aggregation (Path 2) We convert the strength vector of the decision argument into a decision vector. We employ two different decision interpretations: *binary* and *uniform five-level*. Binary interpretation simply treats a strength of 0.5 or below to be '*reject*', and over 0.5 to be '*accept*'. For the uniform five-level interpretation, the argument's strength range between 0 and 1 is divided equally into five regions. For a decision argument with strength s , the decision $d(s)$ is

$$d(s) = \begin{cases} \text{strong reject} & 0 \leq s < 0.2 \\ \text{weak reject} & 0.2 \leq s < 0.4 \\ \text{borderline} & 0.4 \leq s < 0.6 \\ \text{weak accept} & 0.6 \leq s < 0.8 \\ \text{strong accept} & 0.8 \leq s \leq 1.0 \end{cases}$$

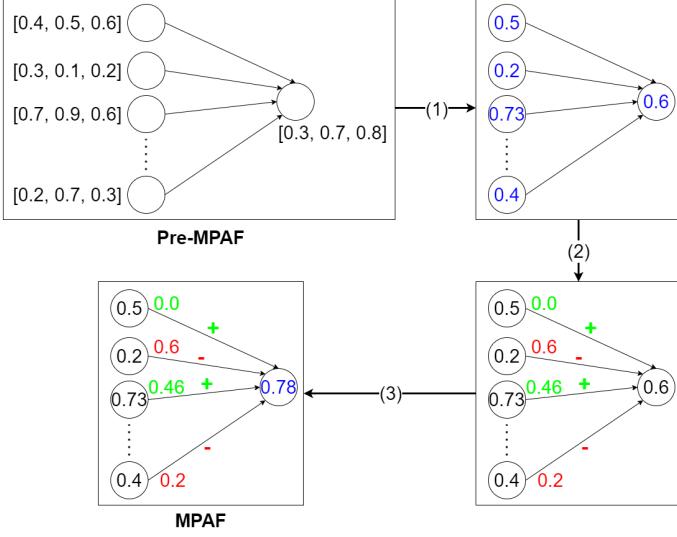


Figure 7: Pre-MPAF to MPAF pipeline. (1) Each strength vector is averaged, giving a strength for the aspect arguments and a base score for the decision argument. (2) Relations and the strength towards the decision arguments are calculated. (3) Semantics (DF-QuAD or MLP-based) is applied to calculate the decision argument’s strength.

We obtain a vector where each entry reflects the decision of one individual reviewer. In the next step, we aggregate the list of decisions. For this, we use majority-voting and all-accept aggregation, and obtain four aggregation functions: *binary majority-voting*, *uniform five-level majority-voting*, *binary all-accept*, and *uniform five-level all-accept*.

Aggregations based on binary decisions are given below.

Definition 9. Let $\vec{d} \in \{\text{reject}, \text{accept}\}^k$ denote a binary decision vector of length k , let $\text{Acc} = \{d \in \vec{d} \mid d = \text{accept}\}$ and $\text{Rej} = \{d \in \vec{d} \mid d = \text{reject}\}$. We define two aggregation methods.

$$\begin{aligned}\sigma_{\text{binary-majority}}(\vec{d}) &= \begin{cases} 1.0 & |\text{Acc}| > |\text{Rej}| \\ 0.0 & |\text{Acc}| \leq |\text{Rej}|\end{cases} \\ \sigma_{\text{binary-all-accept}}(\vec{d}) &= \begin{cases} 1.0 & |\text{Rej}| = 0 \\ 0.0 & \text{otherwise}\end{cases}\end{aligned}$$

We define similar aggregation methods for the uniform five-level decision interpretations. For this, we consider numerical values for the decisions, as usual.

$$map(d) = \begin{cases} -2 & \text{if } d = \text{strong reject (sr)} \\ -1 & \text{if } d = \text{weak reject (wr)} \\ 0 & \text{if } d = \text{borderline (bo)} \\ 1 & \text{if } d = \text{weak accept (wa)} \\ 2 & \text{if } d = \text{strong accept (sa)}\end{cases}$$

Definition 10. Let $\vec{d} \in \{\text{sr}, \text{wr}, \text{bo}, \text{wa}, \text{sa}\}^k$ denote a 5-level decision vector of length k . We define the following aggregation

methods.

$$\begin{aligned}\sigma_{\text{5-level-majority}}(\vec{d}) &= \begin{cases} 1.0 & \sum_{d \in \vec{d}} map(d) > 0 \\ 0.0 & \sum_{d \in \vec{d}} map(d) \leq 0\end{cases} \\ \sigma_{\text{5-level-all-accept}}(\vec{d}) &= \begin{cases} 1.0 & map(d) > 0 \text{ for all } d \in \vec{d} \\ 0.0 & \text{otherwise}\end{cases}\end{aligned}$$

To reach a decision, we apply these functions to the vector associated with the *Decision* argument. For instance, the binary majority-voting aggregation function converts strengths of the *Decision* argument in the pre-MPAF to ‘accept’ or ‘reject’ decisions. The majority decision is then taken, favouring ‘reject’ if tied. For example, a strength vector $[0.1, 0.2, 0.8]$ would be converted to [‘reject’, ‘reject’, ‘accept’] and since there are more ‘reject’ than ‘accept’, the overall decision is ‘reject’ and the strength of the decision argument is set to 0.0.

The idea behind the uniform five-level majority-voting aggregation function is to convert each decision strength of the decision argument of the given pre-MPAF into one of the five decisions (strong-reject, weak-reject, borderline, weak-accept, and strong-accept), map them into a weight value from -2 to 2 in order, then sum the total weights. The paper is predicted to be accepted (by setting strength to 1.0) only if the sum is positive. Using the same example as before, the strength vector $[0.1, 0.2, 0.8]$ would be converted to [‘strong-reject’, ‘weak-reject’, ‘strong-accept’] which is then mapped to [-2, -1, +2]. The sum is -1 which is negative so the strength of the decision argument is set to 0.0.

Instead of counting numbers of ‘accept’ and ‘reject’, we predict ‘reject’ if any decision in a vector is ‘reject’ for the binary all-accept function. Similarly, for the uniform five-level all-accept function, the paper is predicted ‘accept’ only if all the decisions in a vector are strong/weak-accept.

6 Experiments

We evaluate the performance of PeerArg in paper acceptance prediction in comparison with the end-2-end LLM approach. We use three datasets for our classification evaluation: Peer-Review-Analyze (PRA), PeerRead, and Multi-disciplinary Open Peer Review Dataset (MOPRD). The PRA dataset (Ghosal et al., 2022b) contains reviews of accepted and rejected papers from ICLR 2018⁵ conference. Each sentence in a review is annotated for which aspects it belongs to, and the sentiments the sentence has towards such aspects. The PeerRead dataset (Kang et al., 2018b) contains reviews from various computer science conferences. In this paper, only the reviews from the ACL 2017 conference were used since their corresponding papers were classified as accepted or rejected. Additionally, each review has scores for each aspect. We therefore consider two cases when these scores are set and not set as base scores of the aspect arguments. Finally, the MOPRD dataset (Lin et al., 2023) contains reviews from several journals in various fields such as computer science, biology, and medicine. In this paper, we used the reviews from the medical field in our evaluation.

⁵The 6th International Conference on Learning Representations; <https://iclr.cc/archive/www/2018.html>

Base Score	QBAF Semantics	Decision Strength	Aggregation Semantics
default, sentiment	DF-QuAD, MLP	binary, 5-level	majority, all-accept, DF-QuAD, MLP

Table 1: PeerArg Hyperparameters

	Base Score	QBAF Semantics	Decision Strength	Aggregation Semantics
(1)	sentiment	MLP	binary	majority
(2)	sentiment	DF-QuAD	-	DF-QuAD
(3)	default	DF-QuAD	5-level	all-accept
(4)	sentiment	DF-QuAD	5-level	majority

Table 2: Best PeerArg Hyperparameter Combinations. (1) is the best combination for PRA; (2) is the best combination for PeerRead with default (0.5) base scores for the aspect arguments; (3) is the best combination for PeerRead with reviewer ratings as base scores for the aspect arguments; (4) is the best overall combination.

Hyperparameters From Section 5, there are 6 possible variables in the PeerArg pipeline that can be adjusted including *aspect classification*, *sentiment analysis*, *base score setting*, *QBAF semantics*, *decision strength interpretation*, and *aggregation semantics*. We set the aspect classification to be a few-shot learning LLM and the sentiment analysis to be a pretrained sentiment analysis model. The technical details are in Appendices A.1 and A.2. Accordingly, the remaining four variables are the hyperparameters in our experiments.

All hyperparameter combinations are given in Table 1. The base score setting determines how the base scores of the text arguments should be set, either to default (0.5) or to the sentiment strengths obtained during the sentiment analysis process. For QBAF semantics, we consider either DF-QuAD or MLP-based semantics. The decision strength interpretation is either binary or uniform five-level (this only applies for path 2 in Figure 6). Finally, we aggregate using either path 1 in Figure 6, i.e., constructing an MPAF and applying DF-QuAD or MLP-based semantics; or path 2 in Figure 6 and aggregate using majority-voting or all-accept.

Experimental Results We tested the end-2-end LLM as well as all possible combinations of hyperparameters on PRA and PeerRead datasets, identifying the four best combinations, and testing them along with the end-2-end LLM on the MOPRD. Our results show that the end-2-end LLM performs well but is outperformed by PeerRead on all datasets w.r.t. the best overall combination of hyperparameters.

Table 2 shows the best combinations of the hyperparameters for PRA and PeerRead datasets. For PRA (1), the base scores of the aspect arguments were set to 0.5 (default scores). We tested PeerRead with default scores (2) and using the given aspect scores in

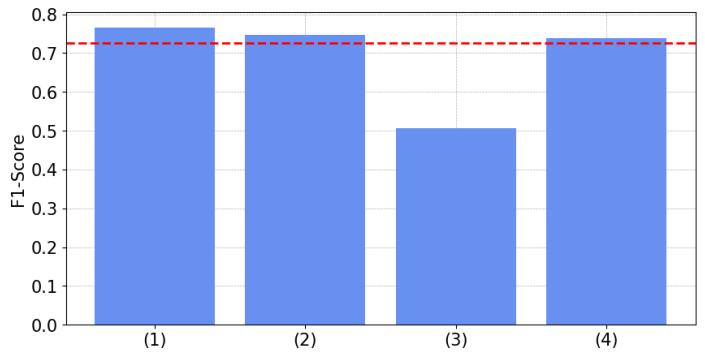


Figure 8: Performances (macro F1) on PRA. The proportion of the correctly predicted paper acceptance is (1) 76.64%, (2) 74.69%, (3) 50.69%, and (4) 73.82%. The red dotted line is the performance of the end-2-end LLM (72.5%).

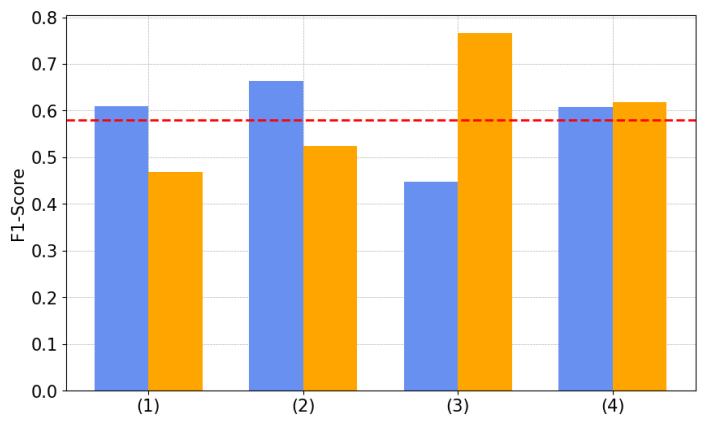


Figure 9: Performances (macro F1) on PeerRead when aspect arguments have default base scores (blue), and when they take on aspect scores from reviews (orange). The proportion of the correctly predicted paper acceptance is (1) 60.91% (blue); 46.94% (orange), (2) 66.29% (blue); 52.49% (orange), (3) 44.83% (blue); 76.56% (orange), and (4) 60.74% (blue); 61.87% (orange). The red dotted line is the performance of the end-2-end-LLM (58%).

each review as base scores for the aspect arguments (3). We find that the best combination of hyperparameters, indicated in (4), is to set sentiment strengths as base scores for text arguments, DF-QuAD semantics, uniform five-level as a decision strength interpretation method, and majority-voting for aggregation.

Crucially, the best overall combination (4) outperformed the end-2-end LLM on all considered datasets. The performances of the four combinations in comparison with the end-2-end LLM on PRA, PeerRead, and MOPRD are visualised in Figures 8, 9, and 10 respectively. For evaluation, we use the macro F1-score which takes both precision (fraction of the true positives over the reported positives) and recall (fraction of the true positives over the number of true positives and false negatives) into account. One remark is that combination (3) only performs well when the aspect scores from reviews are used, otherwise it is outperformed by the end-2-end LLM in all the other settings.

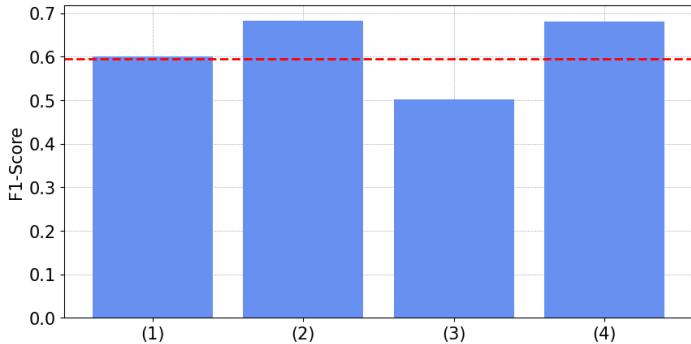


Figure 10: Performances (macro F1) on MOPRD. The proportion of the correctly predicted paper acceptance is (1) 60.12%, (2) 68.32%, (3) 50.27%, and (4) 68.09%. The red dotted line is the performance of the end-2-end LLM (59.5%).

7 Conclusion

We introduced two approaches, PeerArg and an end-2-end LLM, to enhance the peer reviewing process by predicting paper acceptance from reviews. In contrast to the end-2-end LLM that uses few-shot learning techniques to predict paper acceptance in a black-box nature, PeerArg adopts both methods from LLM and computational argumentation to support a decision in the peer reviewing process. Our experimental results show that PeerArg can outperform the end-2-end LLM, while being more transparent due to the interpretable nature of argumentation.

For future work, we plan to leverage the interpretability of the proposed argumentation model to improve the explainability of the (automated) review aggregation process, similar to how argumentation models are recently used to interpret neural networks, especially the multi-layer perceptrons (Potyka, 2021). Moreover, we aim to combine text arguments into pre-MPAFs in the reviews aggregation step. It would also be interesting to explore uncertainty in aggregation and how it would affect the acceptance prediction.

A Appendix (Technical Details)

This section outlines all the relevant models involved in the argument mining process (aspect classification & sentiment analysis) of the PeerArg pipeline.

A.1 Aspect Classification

The LLM for aspect classification is a quantised 4-bit pretrained Mistral-7B-v0.1 model from Mistral AI⁶. We used few-shot learning method (Brown et al., 2020). Inspired from (Gorur et al., 2024), the template contains a primer and a prompt. The primer has a description of seven aspect criteria for a paper review, and ten different samples of review sentences with their corresponding aspects. The prompt contains the review sentence we want to classify the aspect(s) but with no labels. Note that we split the review into sentences using NLTK sentence tokeniser (Bird et al.,

7 aspect criteria for paper review:

- Appropriateness (APR): Does the paper fit with this conference?
- Clarity (CLA): Is the paper well-written and well structured? Is it clear what was done and why?
- Novelty (NOV): How original is the approach? Does the paper break new ground in topic, methodology, or content?
- Empirical and Theoretical Soundness (EMP): Is the approach sound and well-chosen? Are the empirical claims supported by proper experiments? Are the results of the experiments correctly interpreted? Are the arguments in the paper cogent and well-supported?
- Meaningful Comparison (CMP): Is the work adequately compared against existing literature? Are the references adequate?
- Substance (SUB): Does this paper have enough substance, or would it benefit from more ideas or results?
- Impact (IMP): Is the work significant? Does it inspire new ideas or insights which can be impactful to the community?

N.B. Other comments not belonging to any aspect should be classified as OTHER.

Sentence: -Deep Generative Replay section and description of DGDMN are written poorly and is very incomprehensible.

Aspects: CLA

Sentence: * One observation not discussed by the authors is that the performance of the student network under each low precision regime doesn't improve with deeper teacher networks (see Table 1, 2 & 3).

Aspects: SUB, EMP

Sentence: **input sentence from a review**

Primer
Prompt

Aspects:

Figure 11: LLM Aspect Classification Input Template

2009). We also removed the newlines, backward slashes, and leading dashes from each sentence.

The partial template for the primer & prompt input to the LLM is shown in Figure 11. The aspect criteria are described first, followed by the samples of sentence-to-aspects classification. Each sample starts with *Sentence*:-placeholder followed by a sentence, then a newline with *Aspects*:-placeholder followed by one or more aspects the sentence is associated with. The prompt section has a similar sentence-aspects template, but leaving space after the *Aspects*:-placeholder to let the LLM predict the aspects.

A.2 Sentiment Analysis

Sentiment analysis is done per sentence in a review using a pre-trained RoBERTa model trained on Twitter tweets fine-tuned for sentiment analysis⁷. The model takes an input text and returns an output in the form $\langle \text{label}, \text{strength} \rangle$ where *label* is the predicted sentiment (positive/neutral/negative) and *strength* is how likely this sentiment is.

⁶<https://huggingface.co/mistralai/Mistral-7B-v0.1>

⁷<https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest>

Acknowledgments

This research was partially funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 101020934, ADIX), by J.P. Morgan and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme, and by the INDICATE project (Grant No. EP/Y017749/1).

References

- L. Amgoud, C. Cayrol, M.-C. Lagasquie-Schiex, and P. Livet. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems*, 23:1062 – 1093, 10 2008. doi: 10.1002/int.20307.
- K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. Simari, M. Thimm, and S. Villata. Towards artificial argumentation. *AI Magazine*, 38(3):25–36, Oct. 2017. doi: 10.1609/aimag.v38i3.2704. URL <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2704>.
- P. Baroni, A. Rago, and F. Toni. How many properties do we need for gradual argumentation? *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11544. URL <https://ojs.aaai.org/index.php/AAAI/article/view/11544>.
- C. Bhatia, T. Pradhan, and S. Pal. Metagen: An academic meta-review generation system. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’20, page 1653–1656, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401190. URL <https://doi.org/10.1145/3397271.3401190>.
- S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O’Reilly Media, Inc., 2009.
- T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.
- S. Chakraborty, P. Goyal, and A. Mukherjee. Aspect-based sentiment analysis of scientific reviews. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL ’20, page 207–216, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375856. doi: 10.1145/3383583.3398541. URL <https://doi.org/10.1145/3383583.3398541>.
- P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
- ISSN 0004-3702. doi: [https://doi.org/10.1016/0004-3702\(94\)00041-X](https://doi.org/10.1016/0004-3702(94)00041-X). URL <https://www.sciencedirect.com/science/article/pii/000437029400041X>.
- M. Fromm, E. Faerman, M. Berrendorf, S. Bhargava, R. Qi, Y. Zhang, L. Dennert, S. Selle, Y. Mao, and T. Seidl. Argument mining driven analysis of peer-reviews. In *AAAI Conference on Artificial Intelligence*, 2020. URL <https://api.semanticscholar.org/CorpusID:229153112>.
- T. Ghosal, S. Kumar, P. K. Bharti, and A. Ekbal. Peer review analyze: A novel benchmark resource for computational analysis of peer reviews. <https://github.com/Tirthankar-Ghosal/Peer-Review-Analyze-1.0>, 2022a. Includes data from the B12Js_yRb raw reviews of ICLR-2018, accessed on 2024-07-23.
- T. Ghosal, S. Kumar, P. K. Bharti, and A. Ekbal. Peer review analyze: A novel benchmark resource for computational analysis of peer reviews. *PloS one*, 17(1):e0259238, 2022b.
- T. Ghosal, S. Kumar, P. K. Bharti, and A. Ekbal. Peer review analyze: A novel benchmark resource for computational analysis of peer reviews. <https://github.com/Tirthankar-Ghosal/Peer-Review-Analyze-1.0>, 2022c. Includes data from the B13EC5u6W raw reviews of ICLR-2018, accessed on 2024-07-23.
- D. Gorur, A. Rago, and F. Toni. Can large language models perform relation-based argument mining? *CoRR*, abs/2402.11243, 2024. doi: 10.48550/ARXIV.2402.11243. URL <https://doi.org/10.48550/arXiv.2402.11243>.
- X. Hua, M. Nikolov, N. Badugu, and L. Wang. Argument mining for understanding peer reviews. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2131–2137, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1219. URL <https://aclanthology.org/N19-1219>.
- D. Kang, W. Ammar, B. Dalvi, M. van Zuylen, S. Kohlmeier, E. Hovy, and R. Schwartz. A dataset of peer reviews (peer-read): Collection, insights and nlp applications. <https://github.com/allenai/PeerRead>, 2018a. Includes data from 134.json, accessed on 2024-07-23.
- D. Kang, W. Ammar, B. Dalvi, M. van Zuylen, S. Kohlmeier, E. Hovy, and R. Schwartz. A dataset of peer reviews (Peer-Read): Collection, insights and NLP applications. In M. Walker, H. Ji, and A. Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics. doi: 10.18653/v1/N18-1149. URL <https://aclanthology.org/N18-1149>.

- D. Kang, W. Ammar, B. Dalvi, M. van Zuylen, S. Kohlmeier, E. Hovy, and R. Schwartz. A dataset of peer reviews (peer-read): Collection, insights and nlp applications. <https://github.com/allenai/PeerRead>, 2018c. Includes data from 367.json, accessed on 2024-07-23.
- S. Kumar, T. Ghosal, and A. Ekbal. APCS: Towards argument based pros and cons summarization of peer reviews. In T. Ghosal, F. Grezes, T. Allen, K. Lockhart, A. Accomazzi, and S. Blanco-Cuaresma, editors, *Proceedings of the Second Workshop on Information Extraction from Scientific Publications*, pages 117–129, Bali, Indonesia, Nov. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.wiesp-1.14. URL <https://aclanthology.org/2023.wiesp-1.14>.
- J. Li, A. Sato, K. Shimura, and F. Fukumoto. Multi-task peer-review score prediction. In M. K. Chandrasekaran, A. de Waard, G. Feigenblat, D. Freitag, T. Ghosal, E. Hovy, P. Knoth, D. Konopnicki, P. Mayr, R. M. Patton, and M. Shmueli-Scheuer, editors, *Proceedings of the First Workshop on Scholarly Document Processing*, pages 121–126, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.sdp-1.14. URL <https://aclanthology.org/2020.sdp-1.14>.
- J. Lin, J. Song, Z. Zhou, Y. Chen, and X. Shi. Moprd: A multidisciplinary open peer review dataset. *Neural Computing and Applications*, 35(34):24191–24206, Sept. 2023. ISSN 1433-3058. doi: 10.1007/s00521-023-08891-5. URL <http://dx.doi.org/10.1007/s00521-023-08891-5>.
- M. J. Mahoney. Publication prejudices: An experimental study of confirmatory bias in the peer review system. *Cognitive Therapy and Research*, 1977. doi: <https://doi.org/10.1007/BF01173636>.
- T. Mossakowski and F. Neuhaus. Modular semantics and characteristics for bipolar weighted argumentation graphs. *CoRR*, abs/1807.06685, 2018. URL <http://arxiv.org/abs/1807.06685>.
- D. J.-L. Moys. Typographic layout and first impressions: testing how changes in text layout influence reader’s judgments of documents. In *Visible Language*, 2017. URL <https://api.semanticscholar.org/CorpusID:60988587>.
- M. B. Nuijten and J. R. Polanin. “statcheck”: Automatically detect statistical reporting inconsistencies to increase reproducibility of meta-analyses. *Research synthesis methods*, 11(5):574–579, 2020. doi: 10.1002/jrsm.1408.
- N. Potyka. Interpreting neural networks as quantitative argumentation frameworks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6463–6470, May 2021. doi: 10.1609/aaai.v35i7.16801. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16801>.
- A. Rago, F. Toni, M. Aurisicchio, and P. Baroni. Discontinuity-free decision support with quantitative argumentation debates. In *International Conference on Principles of Knowledge Representation and Reasoning*, 2016. URL <https://api.semanticscholar.org/CorpusID:32233959>.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.
- R. Zhou, L. Chen, and K. Yu. Is LLM a reliable reviewer? a comprehensive evaluation of LLM on automatic paper reviewing tasks. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9340–9351, Torino, Italia, May 2024. ELRA and ICCL. URL <https://aclanthology.org/2024.lrec-main.816>.