electronics

MDPI

Article

A Comparison-Based Framework for Argument Quality Assessment

Jianzhu Bao [1], Bojun Jin [1], Yang Sun [1], Yice Zhang [1], Yuhang He [1] and Ruifeng Xu [1,2,3,*]

[1] Harbin Institute of Technology, Shenzhen 518055, China; 20b951008@stu.hit.edu.cn (J.B.); 24s051022@stu.hit.edu.cn (B.J.); yang.sun@stu.hit.edu.cn (Y.S.); 20b951019@stu.hit.edu.cn (Y.Z.); 22s051051@stu.hit.edu.cn (Y.H.)

[2] Peng Cheng Laboratory, Shenzhen 518055, China

[3] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen 518055, China

* Correspondence: xuruifeng@hit.edu.cn

Abstract: Assessing the quality of arguments is both valuable and challenging. Humans often find that making pairwise comparisons between a target argument and several reference arguments facilitates a more precise judgment of the target argument's quality. Inspired by this, we propose a comparison-based framework for argument quality assessments (CompAQA), which scores the quality of an argument through multiple pairwise comparisons. Additionally, we introduce an argument order-based data augmentation strategy to enhance CompAQA's relative quality comparison ability. By introducing multiple reference arguments for pairwise comparisons, CompAQA improves the objectivity and precision of argument quality assessments. Another advantage of CompAQA is its ability to integrate both pairwise argument quality classification and argument quality ranking tasks into a unified framework, distinguishing it from existing methods. We conduct extensive experiments using various pre-trained encoder-only models. Our experiments involve two argument quality ranking datasets (IBM-ArgQ-5.3kArgs and IBM-Rank-30k) and one pairwise argument quality classification dataset (IBM-ArgQ-9.1kPairs). Overall, CompAQA significantly outperforms several strong baselines. Specifically, when using the RoBERTa model as a backbone, CompAQA outperforms the previous best method on the IBM-Rank-30k dataset, improving Pearson correlation by 0.0203 and Spearman correlation by 0.0148. On the IBM-ArgQ-5.3kArgs dataset, it shows improvements of 0.0069 in Pearson correlation and 0.0208 in Spearman correlation. Furthermore, CompAQA demonstrates a 4.71% increase in accuracy over the baseline method on the IBM-ArgQ-9.1kPairs dataset. We also show that CompAQA can be effectively applied to fine-tune larger decoder-only pre-trained models, such as Llama.

Keywords: argument analysis; argument quality assessment; comparison

check for updates

Citation: Bao, J.; Jin, B.; Sun, Y.; Zhang, Y.; He, Y.; Xu, R. A Comparison-Based Framework for Argument Quality Assessment. *Electronics* **2024**, *13*, 4088. https://doi.org/10.3390/electronics13204088

Academic Editor: Christos J. Bouras

Received: 30 August 2024
Revised: 10 October 2024
Accepted: 13 October 2024
Published: 17 October 2024

1. Introduction

Recent years have witnessed significant and rapid advancements in computational argumentation [1–3], where various tasks have been investigated, such as argument mining [4–6], argumentative relation classification [7–9], and argument generation [10–12], among others. As a critical facet in the field of computational argumentation, argument quality assessment has received increasing attention [13–15]. Argument quality plays a crucial role in many downstream applications, such as argumentative writing support [16,17], argument search [18], automatic essay scoring [19,20], and debate systems [21,22].

Some existing studies focus on grading arguments on various quality dimensions [23,24], while others aim to assess the overall quality of arguments [13,21]. Assessing the overall argument quality is very useful in real-world applications because it directly tells a debater which argument is better to use. This line of research primarily focuses on two types of tasks: pairwise argument quality classification and argument quality ranking. The former compares the quality of a pair of arguments [21], while the latter assigns quality scores to an individual argument [13]. Table 1 shows examples of both tasks. For the pairwise argument quality classification task, the input consists of two arguments on the same topic, while the argument quality ranking task takes a single argument as input. In the pairwise argument quality classification example in Table 1, "Argument 2" is deemed higher quality than "Argument 1" because it provides concrete evidence rather than simply stating a claim. The argument quality ranking example in Table 1 demonstrates a high-quality argument that articulates specific evidence and reasoning processes. As a result, it received a high score of 0.80 in the IBM-ArgQ-5.3kArgs dataset [21].

Table 1. Examples of the pairwise argument quality classification task and the argument quality ranking task from the IBM-ArgQ-9.1kPairs and IBM-ArgQ-5.3kArgs datasets [21]. In these examples, the topic of all input arguments is "Gambling should be banned". The pairwise argument quality classification task is a binary classification problem, aiming to predict whether the first argument (binary label: 1) or the second (binary label: 0) is of higher quality. In contrast, the argument quality ranking task is a regression problem, aimed at assigning a quality score in the range of $[0, 1]$ to a single argument, where higher scores indicate better quality.

Task	Input		Output
Pairwise Argument Quality Classification	Argument 1 : Argument 2:	We should ban gambling because there is no benefit to allowing it. We should ban gambling because it preys on people with addictions to make a few wealthy casino owners richer.	Binary Label: 0
Argument Quality Ranking	Gambling doesn't benefit society in that it doesn't produce anything in the way farms provide food or engineers create new technologies or artists create beauty.		Quality Score: 0.80

Previous methods typically address either the pairwise argument quality classification or the argument quality ranking task independently [13,21,25], overlooking their inherent connection and potential synergy. However, directly scoring the quality of an argument is a highly subjective task, and it is difficult to provide an objective score without reference standards. Intuitively, for a given argument, conducting pairwise quality comparisons with multiple other arguments can lead to a more accurate quality score determination, as these comparisons provide relative benchmarks and reduce subjectivity. In fact, in some studies [25,26], the annotation of the argument quality ranking task, namely the quality score of each argument, is achieved through pairwise annotation. Therefore, we contend that pairwise comparisons between arguments should be considered a crucial factor when developing methods for the argument quality ranking task.

For this purpose, in this paper, we introduce a comparison-based argument quality assessment framework, CompAQA. This framework ranks argument quality by performing multiple pairwise comparisons and is naturally applicable to the pairwise argument quality classification task. To predict the quality score of an input argument, CompAQA first selects a set of reference arguments from the training set. Then, multiple pairwise argument quality comparisons are conducted to predict the argument quality score. Additionally, we propose an argument order-based data augmentation strategy to enhance the pairwise comparison results. This strategy aims to mitigate the biases introduced by the input order of two arguments. CompAQA unifies the pairwise argument quality classification and argument quality ranking tasks within a single framework, which is a significant advantage over previous approaches.

Based on pre-trained encoder-only models (BERT [27], RoBERTa [28], and DeBERTa [29]), we evaluate CompAQA on three datasets: one for the pairwise argument quality classification task (IBM-ArgQ-9.1kPairs [21]) and two for the argument quality ranking task

(IBM-ArgQ-5.3kArgs [21] and IBM-Rank-30k [13]). The results demonstrate that our method significantly outperforms strong baselines across all datasets. Using RoBERTa as the base model, CompAQA surpasses the previous state-of-the-art approach on the IBM-Rank-30k dataset, with improvements of 0.0203 and 0.0148 in Pearson and Spearman correlations, respectively. For the IBM-ArgQ-5.3kArgs dataset, CompAQA exhibits enhanced performance, increasing the Pearson correlation by 0.0069 and Spearman correlation by 0.0208. Additionally, when evaluated on the IBM-ArgQ-9.1kPairs dataset, our model achieves a significant 4.71% increase in accuracy compared to the baseline approach. We further demonstrate the versatility of CompAQA by applying it to larger decoder-only pre-trained models, such as Llama, achieving promising results. Additionally, we extend our analysis to evaluate the performance of ChatGPT, a large language model, on the argument quality ranking task. It turns out that the performance of ChatGPT with in-context learning is inferior to that of the smaller fine-tuned models.

We summarize our contributions as follows:

- We introduce CompAQA, a novel comparison-based framework for argument quality assessments, which is applicable to both the pairwise argument quality classification task and the argument quality ranking task.
- CompAQA enhances objectivity and accuracy in argument quality ranking through a systematic approach of leveraging multiple pairwise comparisons with carefully selected reference arguments.
- Extensive evaluations across multiple datasets and model architectures validate the superiority and versatility of CompAQA.

The remainder of this paper is structured as follows: Section 2 discusses related work. Section 3 presents a comprehensive description of our proposed method, CompAQA. Section 4 details our experimental setup, including the datasets, evaluation metrics, implementation details, and baseline methods. Section 5 presents a detailed discussion of the experimental results. Finally, Section 6 concludes our study.

2. Related Work

Assessing the quality of arguments is a highly challenging task, and has long been the subject of much research [30]. Some early work focuses on specific aspects of argument quality, such as relevance [31], semantic aspects [32], structure [33], and sufficiency [17], etc. Drawing from argumentation theories [34,35], Wachsmuth et al. [23] developed a comprehensive framework for evaluating argument quality, encompassing three aspects: logic, rhetoric, and dialectic. Wachsmuth and Werner [24] further explored the assessment of the intrinsic argument quality across 15 fine-grained dimensions. Following Wachsmuth et al. [23], Lauscher et al. [36] proposed a theory-based argument quality assessment corpus and explored approaches based on pre-trained models.

While evaluating arguments on various quality aspects is valuable, assessing the overall argument quality proves more practical for real-world applications, as it provides debaters with clear guidance on which arguments are most effective. Therefore, there has been an increasing focus on developing corpora and resources for assessing the overall quality of arguments. Persing and Ng [37] designed a corpus for assessing argument strength, representing an early exploration of overall argument quality. Habernal and Gurevych [25] studied the convincingness of arguments sourced from the Web. Toledo et al. [21] collected high-quality, human-written arguments and annotated them regarding both individual (IBM-ArgQ-5.3kArgs) and pairwise (IBM-ArgQ-9.1kPairs) argument quality. Gretz et al. [13] constructed a large-scale argument quality ranking dataset, IBM-Rank-30k, comprising over 30k arguments, each annotated with a quality score. Gienapp et al. [26] presented an argument quality annotation framework that can efficiently convert pairwise judgments into trustworthy quality scores. Skitalinskaya et al. [38] proposed a claim quality assessment dataset based on the revision history of online debate websites. Joshi et al. [15] not only annotated argument quality but also provided an analysis of each argument, explaining the rationale behind its perceived veracity.

With recent advances in pre-training techniques, an increasing number of methods based on pre-trained models have been proposed. Toledo et al. [21] presented an early attempt at fine-tuning the pre-trained BERT model to assess argument quality. Marro et al. [14] designed a model based on graph embeddings, incorporating both textual features and argument structure features. Favreau et al. [39] explored a series of BERT-based ranking methods for evaluating argument quality, employing various loss functions including point-wise and list-wise losses. Wang et al. [40] proposed a model based on supervised contrastive learning to capture the complex interplay between arguments. They also incorporated discourse relation knowledge to enhance argument quality assessments. In addition to the aforementioned work directly assessing argument quality, some studies explored this topic from other perspectives. Falk and Lapesa [41] utilized adapter-based methods to examine the relationship among different aspects of argument quality. Fromm et al. [42] investigated the connections between argument quality assessment and other argument mining tasks, such as argument identification and evidence detection.

Among the aforementioned methods, the works most closely related to ours are those of Toledo et al. [21], Favreau et al. [39], and Wang et al. [40], as they share the same task objective and utilize identical datasets. Toledo et al. [21] employed a basic fine-tuning approach with BERT, without considering comparisons between arguments or any shared references. Although Favreau et al. [39] and Wang et al. [40] attempted to address a comparison of arguments through ranking and contrastive learning loss functions, respectively, their approach to achieving this goal remained implicit, relying solely on loss functions. Consequently, the models may not acquire a sufficient ability to compare argument quality effectively. In contrast to these works, our proposed method achieves explicit argument quality comparisons by directly encoding pairs of arguments together using a pre-trained model. This approach enables our method to better construct argument quality comparisons, thereby more accurately predicting quality scores. Moreover, our method can be applied to solve the pairwise argument quality classification task, a capability that previous methods lack.

3. Method

We introduce CompAQA, a comparison-based framework designed for evaluating argument quality. The architecture of CompAQA is illustrated in Figure 1. To predict the quality score of a given target argument, CompAQA first selects several reference arguments from the training set to form multiple comparison pairs. Next, a pairwise comparison module evaluates each argument pair to analyze the relative quality between the target argument and each reference. Through joint encoding of the target and reference arguments using pre-trained models, the pairwise comparison module can directly capture complex semantic interactions between the two arguments. This enables an explicit modeling of their relative quality. As a result, CompAQA can predict the quality score if each target argument more accurately by considering multiple reference arguments. To further improve the robustness of CompAQA, the pairwise comparison module is enhanced through an argument order-based data augmentation strategy, which mitigates potential biases stemming from the input order of the two arguments. Moreover, compared to previous approaches [21,39,40], our method offers a notable advantage by integrating both pairwise argument quality classification and argument quality ranking tasks into a single, unified framework.

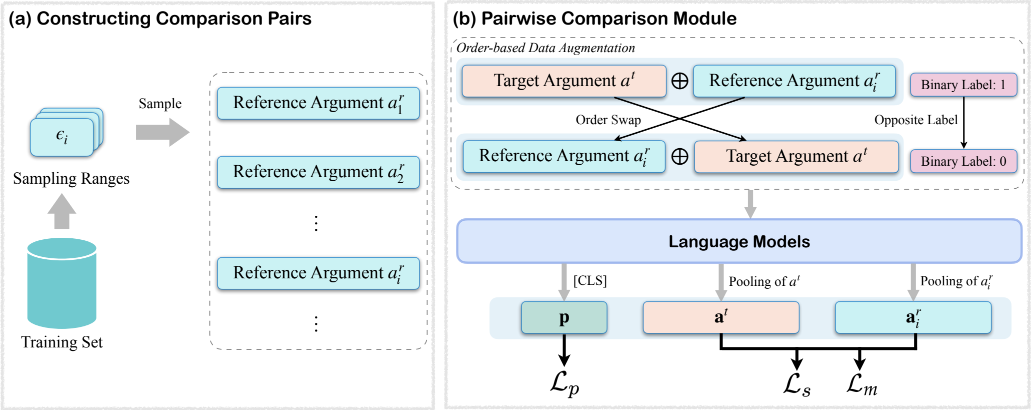

Figure 1. The architecture of CompAQA. For the sake of simplicity, we omit the topic corresponding to each argument. Here, we assume that the quality of the "Target Argument a^t" is superior to that of the "Reference Argument a_i^r".

3.1. Problem Definition

As shown in Table 1, for the argument quality ranking task, the input is a single argument a^t with topic t^t, and the output is a quality score $y_s \in [0,1]$, where a higher score indicates a better quality of a^t. For the pairwise argument quality classification task, the input consists of two arguments, a_1 and a_2, along with their corresponding topics t_1 and t_2. The output is a binary label $y_c \in \{0,1\}$, where 1 and 0, respectively, indicate whether a_1 or a_2 is of better quality.

In the following sections, we mainly describe how our proposed CompAQA addresses the argument quality ranking task. The pairwise comparison module of CompAQA can be directly used to solve the pairwise argument quality classification task.

3.2. Constructing Comparison Pairs

For the input target argument a^t, CompAQA first selects m reference arguments $\{a_1^r, a_2^r, \ldots, a_m^r\}$ from the training set for subsequent comparisons. Intuitively, the quality scores of reference arguments should be uniformly distributed across the valid score range. This distribution would better reflect the scoring criteria, clearly distinguishing between low-quality and high-quality arguments. Therefore, we designed a sampling method that can select reference arguments from the training set with scores that are uniformly distributed across the quality spectrum.

Specifically, we first determine m sampling ranges. For the i-th reference argument to be sampled, its corresponding sampling range ϵ_i is as follows:

$$\epsilon_i = [\mu_i - l, \mu_i + l], \tag{1}$$

$$\mu_i = \frac{1.0}{m+1} \cdot i. \tag{2}$$

where $i \in \{1, 2, \ldots, m\}$, l is a hyperparameter used to control the size of the range. To avoid overlap between different ranges, it must be strictly ensured that $l < \frac{1.0}{m+1}/2$. The impact of the hyperparameters m and l is discussed in detail in Section 5.4. For instance, if we set $m = 3$ and $l = 0.05$, then $\mu_1 = 0.25$, $\mu_2 = 0.50$, and $\mu_3 = 0.75$. The three intervals would be $\epsilon_1 = [0.20, 0.30]$, $\epsilon_2 = [0.45, 0.55]$, and $\epsilon_3 = [0.70, 0.80]$, representing the score ranges for low-, medium-, and high-quality arguments, respectively.

Once the m sampling ranges are determined, we randomly sample one reference argument from each range, thereby obtaining m reference arguments. Importantly, these m reference arguments remain constant throughout the entire training and inference process. In other words, all input target arguments are evaluated through comparison with this fixed set of m reference arguments. This approach ensures consistency in the training and inference processes, and allows for standardized comparisons across different target arguments.

The two-step process of first determining m sampling intervals and then selecting arguments from each interval ensures that the quality scores of the sampled reference arguments are distributed as uniformly as possible. Consequently, this approach better reflects the standards for argument quality evaluation, enabling a more accurate determination of the target argument's quality.

After obtaining m reference arguments $\{a_1^r, a_2^r, \ldots, a_m^r\}$, we pair each with the target argument a^t to form m comparison argument pairs $\{(a^t, a_1^r), (a^t, a_2^r), \ldots, (a^t, a_m^r)\}$. In the training phase, we determine the binary classification labels $\{y_{c,1}, y_{c,2}, \ldots, y_{c,m}\}$ for each argument pair based on the ground-truth quality scores of the two arguments. Here, the ground-truth quality score refers to the manually annotated quality score provided in the training dataset. For instance, for an argument pair (a^t, a_i^r), if the ground-truth quality score of a^t is higher than or equal to the quality score of a_i^r, then its binary classification label is 1; otherwise, it is 0.

3.3. Pairwise Comparison Module

By feeding each of the m comparison argument pairs into the pairwise comparison module, we can derive m predicted quality scores for the target argument a^t, each from the perspective of a different reference argument.

3.3.1. Text Encoding

For a pair of input arguments (a^t, a_i^r), the purpose of text encoding is twofold. First, it aims to obtain an overall vector representation $\mathbf{p}$ for this pair. Second, it generates individual vector representations $\mathbf{a}^t$ and $\mathbf{a}_i^r$ for each argument. Subsequently, $\mathbf{p}$ will be utilized for pairwise argument quality classification, while $\mathbf{a}^t$ and $\mathbf{a}_i^r$ will be used to predict individual quality scores for each argument.

Specifically, for pre-trained encoder-only models like BERT [27], we formalize the pair (a^t, a_i^r) as the following sequence:

$$S = \text{“ [CLS] } a^t \mid t^t \text{ [SEP] } a_i^r \mid t_i^r \text{ [SEP] ”}. \tag{3}$$

where t_i^r and t^t is the topic of a_i^r and a^t. S is then fed into a pre-trained model to encode each token as a contextual embedding vector. Here, we use the embedding of "[CLS]" as the argument pair representation $\mathbf{p}$. Then, we perform max pooling on the embeddings of all tokens in "$a^t \mid t^t$" and "$a_i^r \mid t_i^r$" to capture the most salient features, resulting in the representations $\mathbf{a}^t$ and $\mathbf{a}_i^r$ for the first and second arguments, respectively.

Unlike previous methods [39,40], which encode arguments separately, our approach encodes the target argument and reference argument together through a pre-trained model. This joint encoding enables the better capture of semantic interactions between the two arguments, potentially leading to more nuanced quality comparisons.

3.3.2. Pairwise Classification

During training, we classify each argument pair (a^t, a_i^r) with a binary classification label $y_{c,i}$. This classification task aims to strengthen the pairwise comparison module's understanding of the relative quality between two arguments. Intuitively, the more accurately the model captures relative quality differences, the more precise its predictions of absolute quality scores become. Concretely, the argument pair representation $\mathbf{p}$ is utilized for predicting $y_{c,i}$ via a binary classifier $f_p(\cdot)$. Subsequently, we compute the cross-entropy loss $\mathcal{L}_p$ for pairwise argument quality classification:

$$\mathcal{L}_p = -[y_{c,i} \log(f_p(\mathbf{p})) + (1 - y_{c,i}) \log(1 - f_p(\mathbf{p}))]. \tag{4}$$

3.3.3. Quality Score Prediction

Here, we predict the quality scores for both the target argument a^t and the reference argument a_i^r in each input pair. To be specific, two regressors $f_{s1}(\cdot)$ and $f_{s2}(\cdot)$ are used to individually predict the quality scores of the two arguments, and the mean absolute error (MAE) loss is employed for optimization:

$$\hat{y}_s^t = f_{s1}(\mathbf{a}^t), \tag{5}$$

$$\hat{y}_{s,i}^r = f_{s2}(\mathbf{a}_i^r), \tag{6}$$

$$\mathcal{L}_s = \frac{1}{2}\left(|y_s^t - \hat{y}s^t| + |y_{s,i}^r - \hat{y}_{s,i}^r|\right). \tag{7}$$

where y_s^t and $y_{s,i}^r$ represent the ground-truth quality scores of the target and reference arguments, while $\hat{y}_s^t$ and $\hat{y}_{s,i}^r$ are their predicted scores.

To further enhance the pairwise comparison module's ability to learn the relative quality of multiple arguments, we introduce a margin ranking loss:

$$\mathcal{L}_m = \max(0, -\delta * (\hat{y}_s^t - \hat{y}_{s,i}^r) + \gamma). \tag{8}$$

where δ is 1 when y_s^t is greater than or equal to $y_{s,i}^r$, and -1 when y_s^t is less than $y_{s,i}^r$. γ is the margin hyperparameter, which enforces a minimum difference between the predicted scores of the compared arguments. By incorporating this margin ranking loss, we ensure that the model not only predicts accurate scores but also maintains appropriate relative rankings between arguments of varying quality.

The total loss $\mathcal{L}$ during training is the weighted sum of $\mathcal{L}_p$, $\mathcal{L}_s$, and $\mathcal{L}_m$:

$$\mathcal{L} = \lambda_p \mathcal{L}_p + \lambda_s \mathcal{L}_s + \lambda_m \mathcal{L}_m. \tag{9}$$

where λ_p, λ_s, and λ_m are hyperparameters that control the relative importance of each loss component.

In this manner, the pairwise comparison module can simultaneously learn to achieve two objectives: (1) to compare the relative quality of two arguments through $\mathcal{L}_p$ and $\mathcal{L}_m$, and (2) to predict the specific scores of individual arguments through $\mathcal{L}_s$.

3.4. Order-Based Data Augmentation

Pre-trained language models can produce biased encoding representations due to the order of elements in input or output sequences [43–45]. In our framework, the same issue arises. Specifically, when encoding the sequence S in Section 3.3 using pre-trained language models, we consistently place the target argument before the reference argument. However, there should not be such an order relation between the target and reference arguments. Therefore, to alleviate this issue, we propose an argument order-based data augmentation strategy.

Specifically, we create a new data sample S' by swapping the order of "$a^t \mid t^t$" and "$a_i^r \mid t_i^r$" in S:

$$S' = \text{``[CLS]}\, a_i^r \mid t_i^r\, \text{[SEP]}\, a^t \mid t^t\, \text{[SEP]''}. \tag{10}$$

Consequently, the binary classification label of S' is the opposite of S; that is, 0 becomes 1, and 1 becomes 0. This order-based data augmentation strategy reduces the pairwise comparison module's sensitivity to the order of arguments, thereby enhancing its robustness.

3.5. Inference

Recall that we constructed m argument pairs for each target argument in Section 3.2. With our order-based data augmentation strategy, each argument pair produces two distinct data samples for the pairwise comparison module. As a result, during inference, each target argument receives $2m$ predicted scores. We take the average of all these scores as the final predicted score for the target argument.

4. Experiments

4.1. Datasets

We evaluated CompAQA on three datasets, namely IBM-ArgQ-9.1kPairs, IBM-ArgQ-5.3kArgs [21], and IBM-Rank-30k [13]. IBM-ArgQ-9.1kPairs is designed for pairwise argument quality classification, while IBM-ArgQ-5.3kArgs and IBM-Rank-30k were created for argument quality ranking. IBM-ArgQ-9.1kPairs and IBM-ArgQ-5.3kArgs both contain arguments from 11 different topics, with each sample annotated by 15 to 17 annotators to ensure data quality. IBM-Rank-30k is a larger dataset, comprising 30k arguments spanning 71 different topics. Following the recommendation of Gretz et al. [13], we used the weighted-average score as the ground-truth quality score for each argument, as it demonstrated higher agreement in its annotations. In both the IBM-Rank-30k and IBM-ArgQ-5.3kArgs datasets, argument quality scores are normalized to the range $[0, 1]$, with higher scores indicating higher quality.

As there is no official data split for IBM-ArgQ-9.1kPairs and IBM-ArgQ-5.3kArgs, we randomly selected seven topics as the training set. The remaining four topics were evenly divided, with two topics assigned to the validation set and two to the test set. For IBM-Rank-30k, we adhered to the original data split proposed by Gretz et al. [13], utilizing data samples from 49, 7, and 15 topics for training, validation, and testing, respectively.

4.2. Evaluation Metrics

For the argument quality ranking task, we primarily used Pearson and Spearman correlations as the evaluation metrics, following previous work [13,21]. Additionally, we incorporated Kendall's Tau (TAU) and Normalized Discounted Cumulative Gain (NDCG@15), in line with recent studies [40]. Given that argument quality ranking is essentially a regression task, we also reporedt the Mean Absolute Error (MAE) to provide a direct measure of prediction accuracy. For the pairwise argument quality classification task, we used accuracy, F1 score, and area under curve (AUC).

4.3. Implementation Details

Our main experiments were conducted based on BERT-base, RoBERTa-base, and DeBERTa-base. We set the learning rate to 5×10^{-6} for pre-trained layers, while, for other layers, the rate was 5×10^{-4} for BERT, 5×10^{-5} for RoBERTa, and 1×10^{-5} for DeBERTa. We used a batch size of 32 and a warm-up ratio of 0.05, with all dropout rates consistently set to 0.1. For the loss function weights, we set $\lambda_p = 0.01$, $\lambda_s = 1.0$, and $\lambda_m = 1.0$. The γ parameter in Equation (8) was set to 0 for all experiments. For optimization, we employed AdamW [46] with a weight decay of 0.1. The binary classifier $f_p(\cdot)$ and regressors $f_{s1}(\cdot)$ and $f_{s2}(\cdot)$ were implemented as single-layer Multi-Layer Perceptrons (MLPs). All of the pre-trained language models were obtained from HuggingFace's Transformers library [29]. To construct comparison pairs, we set $m = 3$ and $l = 0.05$. That is, we randomly sampled a reference argument from each of the following three intervals: $[0.20, 0.30]$, $[0.45, 0.55]$, and $[0.70, 0.80]$. We selected the best checkpoint based on the Pearson correlation score on the validation set. All experiments were carried out five times and the mean scores were reported.

For experiments involving ChatGPT, we utilized the gpt-3.5-turbo-instruct model from OpenAI's official API.

4.4. Compared Methods

For the argument quality ranking task, we compared our proposed model with the following baselines:

- SVM BOW [13] is a support vector regression ranker with an RBF kernel and bag-of-words features.

- Bi-LSTM GloVe [13] is a Bi-LSTM model with self-attention mechanism, utilizing GloVe embeddings [47].
- BERT, RoBERTa, and DeBERTa refer to pre-trained language models fine-tuned on each dataset. The specific fine-tuning process followed the work of Gretz et al. [13] and Toledo et al. [21]. After confirmation with the authors, we learned that Toledo et al. [21] did not use a validation set in their BERT-based experiments on IBM-ArgQ-9.1kPairs and IBM-ArgQ-5.3kArgs. Therefore, based on our own data split, we replicated the BERT baseline on IBM-ArgQ-9.1kPairs and IBM-ArgQ-5.3kArgs using the hyperparameters provided by Toledo et al. [21]. Note that the inputs of these models all include the topic of each argument.
- TFR-BERT [39] is an ensemble method. It ensembles multiple BERT models fine-tuned with various ranking losses.
- CI-BERT/RoBERTa/DeBERTa [40] enables contextual interaction via supervised contrastive learning and introduces external discourse knowledge via a Discourse-Aware Graph Network [48].
- ChatGPT is evaluated using in-context learning in few-shot settings. Specifically, we conducted tests on 0-shot, 2-shot, and 4-shot settings.

For the pairwise argument quality classification task, the following baselines were compared:

- BERT/RoBERTa/DeBERTa-Pair-CLS directly applies pre-trained language models for sentence pair classification. Notably, CompAQA distinguishes itself from these baselines by incorporating an order-based data augmentation strategy.

5. Results and Discussions

5.1. Main Results

The main experimental results for the argument quality ranking task on IBM-ArgQ-5.3kArgs and IBM-Rank-30k are shown in Table 2. First, we compared CompAQA with baselines that do not incorporate any external knowledge. It is evident that CompAQA outperforms the BERT/RoBERTa/DeBERTa baselines on both datasets in terms of the Pearson, Spearman, and TAU metrics. Furthermore, we can see that although CI-BERT/CI-RoBERTa integrates discourse knowledge through an additional graph network, CompAQA still outperforms it in most metrics. For NDCG@15, CompAQA is either on par with or slightly worse than the baselines. Since NDCG@15 focuses solely on the top 15 arguments ranked by ground-truth quality score in the test set, we argue that this metric cannot provide a comprehensive evaluation of models' overall performance. Although CompAQA's predictions for the quality scores of the top 15 arguments are slightly inferior to some baselines, all other metrics demonstrate its overall advantage across the entire test set.

Regarding ChatGPT's performance, we observe that both the Pearson and Spearman correlations decrease as the number of examples increases. Wang et al. [40] also observed a similar phenomenon, where an increase in the number of examples leads to a decrease in performance. Their explanation posits that as the number of examples grows, ChatGPT tends to repeat the content of the examples from the prompt, resulting in diminished performance. We observed a similar phenomenon, lending credence to this explanation.

CompAQA can be easily adapted for the pairwise argument quality classification task by simply removing the quality score-related losses, namely $\mathcal{L}_s$ and $\mathcal{L}_m$, while retaining only the pairwise classification loss $\mathcal{L}_p$. Here, we conducted experiments on the IBM-ArgQ-9.1kPairs dataset, with the results presented in Table 3. Compared to the basic sentence pair classification using pre-trained language models, CompAQA exhibits a significantly better performance in terms of accuracy, F1 score, and AUC metrics. Furthermore, employing DeBERTa as the base model enables CompAQA to achieve the best overall performance.

Table 2. The main results for the argument quality ranking task. "Pear." and "Spear." are short for "Pearson" and "Spearman". The best scores for each type of pre-trained model are highlighted in bold. † indicates results that we replicated using the same hyperparameters as in the original studies. "MAE" stands for mean absolute error. "↓" indicates that a lower "MAE" value is better. Additionally, we mainly compared our method against the SOTA baselines that we replicated ("CI-BERT/RoBERTa") using the official source code from Wang et al. [40]. For reference, we also list the results reported in the original study by Wang et al. [40] at the bottom of the table.

Model	**IBM-ArgQ-5.3kArgs**					**IBM-Rank-30k**				
	Pear.	**Spear.**	**TAU**	**MAE ↓**	**NDCG@15**	**Pear.**	**Spear.**	**TAU**	**MAE ↓**	**NDCG@15**
SVM BOW [13]	-	-	-	-	-	0.3200	0.3100	-	-	-
Bi-LSTM GloVe [13]	-	-	-	-	-	0.4400	0.4100	-	-	-
TFR-BERT [39]	0.3500	0.3400	0.2300	-	0.6600	0.5200	0.4700	0.3200	-	0.8800
BERT [13]	-	-	-	-	-	0.5200	0.4800	-	-	-
BERT †	0.3902	0.3755	0.2597	0.1560	0.7565	0.5201	0.4794	0.3301	0.1328	0.9300
CI-BERT †	0.4101	0.3959	0.2703	**0.1544**	0.7388	0.5230	**0.4845**	0.3380	0.1330	0.9487
CompAQA-BERT (Ours)	**0.4563**	**0.4417**	**0.3064**	0.1580	**0.8097**	**0.5282**	0.4830	**0.3390**	**0.1311**	**0.9635**
RoBERTa [40]	-	-	-	-	-	0.5283	0.4858	-	-	0.9427
RoBERTa†	0.4132	0.3908	0.2716	0.1533	**0.7729**	0.5311	0.4872	0.3545	0.1348	0.9507
CI-RoBERTa†	0.4612	0.4377	0.3013	0.1633	0.7385	0.5439	0.5056	0.3554	0.1339	**0.9668**
CompAQA-RoBERTa (Ours)	**0.4681**	**0.4585**	**0.3165**	**0.1517**	0.7630	**0.5642**	**0.5204**	**0.3670**	**0.1299**	0.9543
DeBERTa	0.4181	0.4030	0.2777	**0.1562**	**0.7497**	0.5604	0.5154	0.3643	0.1667	0.9481
CompAQA-DeBERTa (Ours)	**0.4657**	**0.4536**	**0.3127**	0.1652	0.7352	**0.5797**	**0.5373**	**0.3794**	**0.1371**	**0.9500**
CI-BERT (Reported) [40]	-	-	-	-	-	0.5375	0.4949	-	-	0.9388
CI-RoBERTa (Reported) [40]	-	-	-	-	-	0.5604	0.5174	-	-	0.9648
ChatGPT-0-Shot	0.3720	0.4043	0.2890	0.2353	0.7148	0.2496	0.2464	0.1749	0.2109	0.8217
ChatGPT-2-Shot	0.3466	0.3178	0.2194	0.1996	0.7304	0.2421	0.2335	0.1591	0.2081	0.8524
ChatGPT-4-Shot	0.3394	0.3126	0.2166	0.1997	0.6893	0.2315	0.2357	0.1608	0.2107	0.8522

Table 3. The main results for the pairwise argument quality classification task.

Model		**IBM-ArgQ-9.1kPairs**	
	Acc.	**F1.**	**AUC.**
BERT-Pair-CLS	74.26	73.99	82.88
CompAQA-BERT (Ours)	77.49	77.41	85.69
RoBERTa-Pair-CLS	75.29	75.11	84.21
CompAQA-RoBERTa (Ours)	80.00	79.98	88.03
DeBERTa-Pair-CLS	78.78	78.59	87.59
CompAQA-DeBERTa (Ours)	81.17	81.16	88.83

5.2. Ablation Study

Table 4 presents the results of our ablation study for the argument quality ranking task. Removing any of the components—$\mathcal{L}_s$, $\mathcal{L}_m$, or the order-based data augmentation (ODA) strategy—results in a decrease in CompAQA's performance to varying degrees. Notably, ODA has the most significant impact on overall performance. This demonstrates that the bias introduced by the order of input arguments can indeed negatively impact performance, and our proposed ODA effectively mitigates this issue. Furthermore, we present the results of removing either two or all three of the components: $\mathcal{L}_s$, $\mathcal{L}_m$, and ODA. In these scenarios, the overall performance of CompAQA declines even further.

Table 4. Ablation results for the argument quality ranking task. ODA is short for the order-based data augmentation strategy.

Model	**IBM-ArgQ-5.3kArgs**					**IBM-Rank-30k**				
	Pear.	**Spear.**	**TAU**	**MAE ↓**	**NDCG@15**	**Pear.**	**Spear.**	**TAU**	**MAE ↓**	**NDCG@15**
CompAQA-BERT (Ours)	**0.4563**	**0.4417**	**0.3064**	0.1580	0.8097	**0.5282**	**0.4830**	**0.3390**	**0.1311**	0.9635
w/o $\mathcal{L}_p$	0.4472	0.4344	0.3015	0.1561	0.8035	0.5257	0.4795	0.3363	0.1321	0.9588
w/o $\mathcal{L}_m$	0.4475	0.4337	0.3009	0.1823	0.7987	0.5230	0.4790	0.3358	0.1314	0.9681
w/o ODA	0.4235	0.4096	0.2830	0.1558	0.7916	0.5169	0.4716	0.3305	0.1333	0.9477
w/o $\mathcal{L}_p$ and $\mathcal{L}_m$	0.4401	0.4240	0.2939	**0.1543**	0.8134	0.5227	0.4761	0.3339	0.1322	**0.9768**
w/o $\mathcal{L}_p$ and ODA	0.4189	0.4049	0.2802	0.1560	**0.8236**	0.5136	0.4705	0.3300	0.1331	0.9659
w/o $\mathcal{L}_m$ and ODA	0.4168	0.4027	0.2801	0.1566	0.8147	0.5127	0.4700	0.3283	0.1325	0.9601
w/o $\mathcal{L}_p$ and $\mathcal{L}_m$ and ODA	0.4030	0.3971	0.2629	0.1590	0.7755	0.5095	0.4640	0.3227	0.1323	0.9577
CompAQA-RoBERTa (Ours)	**0.4681**	**0.4585**	**0.3165**	0.1517	0.7630	0.5642	**0.5204**	**0.3670**	0.1299	0.9543
w/o $\mathcal{L}_p$	0.4550	0.4480	0.3074	0.1553	0.7399	**0.5647**	0.5192	0.3661	0.1319	0.9589
w/o $\mathcal{L}_m$	0.4626	0.4470	0.3084	0.1510	0.7385	0.5601	0.5125	0.3610	0.1301	0.9325
w/o ODA	0.4646	0.4492	0.3108	**0.1491**	0.7558	0.5605	0.5167	0.3643	0.1310	0.9584
w/o $\mathcal{L}_p$ and $\mathcal{L}_m$	0.4541	0.4448	0.3059	0.1644	0.7417	0.5572	0.5092	0.3589	**0.1286**	0.9234
w/o $\mathcal{L}_p$ and ODA	0.4603	0.4453	0.3067	0.1498	**0.7813**	0.5589	0.5155	0.3633	0.1319	**0.9697**
w/o $\mathcal{L}_m$ and ODA	0.4580	0.4435	0.3063	0.1527	0.7743	0.5567	0.5052	0.3559	0.1287	0.9386
w/o $\mathcal{L}_p$ and $\mathcal{L}_m$ and ODA	0.4548	0.4430	0.3014	0.1507	0.7223	0.5506	0.5047	0.3530	0.1312	0.9276
CompAQA-DeBERTa (Ours)	**0.4657**	**0.4536**	**0.3127**	0.1652	0.7352	**0.5797**	**0.5373**	**0.3794**	0.1371	0.9500
w/o $\mathcal{L}_p$	0.4625	0.4512	0.3109	0.1700	0.7376	0.5768	0.5328	0.3758	**0.1355**	0.9367
w/o $\mathcal{L}_m$	0.4478	0.4370	0.3001	0.1663	**0.7465**	0.5737	0.5299	0.3736	0.1360	0.9367
w/o ODA	0.4068	0.3919	0.2702	**0.1557**	0.7293	0.5632	0.5205	0.3665	0.1527	0.9517
w/o $\mathcal{L}_p$ and $\mathcal{L}_m$	0.4377	0.4278	0.2930	0.1683	0.7103	0.5728	0.5271	0.3714	0.1420	0.9557
w/o $\mathcal{L}_p$ and ODA	0.4013	0.3896	0.2673	0.1679	0.7334	0.5603	0.5117	0.3597	0.1494	**0.9781**
w/o $\mathcal{L}_m$ and ODA	0.4008	0.3888	0.2666	0.1574	0.7145	0.5631	0.5172	0.3646	0.1844	0.9508
w/o $\mathcal{L}_p$ and $\mathcal{L}_m$ and ODA	0.3953	0.3806	0.2595	0.1616	0.7165	0.5562	0.5037	0.3591	0.1387	0.9404

5.3. Results of Fine-Tuning Decoder-Only Pre-Trained Models

CompAQA is also applicable to decoder-only pre-trained models such as Llama. To adapt it, we only need to replace the Text Encoding part in Section 3.3 with the following method. Specifically, we employ the prompt shown in Figure 2 to derive the argument pair representation $\mathbf{p}$ and the individual argument representations $\mathbf{a}^t$ and $\mathbf{a}_i^r$.

Given two arguments and their corresponding topics, your task is to evaluate and compare the quality of the arguments, then assign a quality score to each argument.

Argument 1: a^t
Topic 1: t^t

Argument 2: a_i^r
Topic 2: t_i^r

Evaluation:

The quality of Argument [label] is better.
Quality Score of Argument 1: [score 1]
Quality Score of Argument 2: [score 2]

Figure 2. Prompt for fine-tuning decoder-only language models.

Here, we use the mean-pooled vector representation of the "[label]" tokens as the argument pair representation $\mathbf{p}$. Similarly, the representations obtained from "[score 1]" and "[score 2]" serve as the individual argument representations $\mathbf{a}^t$ and $\mathbf{a}_i^r$, respectively. We then calculate $\mathcal{L}_p$, $\mathcal{L}_s$, and $\mathcal{L}_m$ for fine-tuning, following the method described in Section 3.3.

For the implementation details, we fine-tuned the Llama-3-8B-Instruct model using LoRA [49]. We set the learning rate to 2 $\times 10^{-4}$, trained for a single epoch with a batch size of 32, employed a warm-up ratio of 0.1, and used a weight decay of 1 $\times 10^{-4}$. For the LoRA configuration, we used a rank (r) of 64 and an alpha of 16. The LoRA dropout rate was set to 0.1. We adjust λ_p, λ_s, and λ_m to 0.1, 1.0, and 0.1, respectively. All other experimental settings remained consistent with those described in Section 4.3. As a baseline for comparison, we also simply fine-tuned Llama-3-8B-Instruct using only MSE loss. Another baseline for comparison, CI-Llama, is our implementation of the method proposed by Wang et al. [40], which is also based on Llama-3-8B-Instruct.

The results of fine-tuning Llama are shown in Table 5. As is evident from the results, CompAQA demonstrates an excellent performance when applied to Llama, showing significant improvements over both the baseline Llama and CI-Llama.

Table 5. Results of fine-tuning Llama on the IBM-Rank-30k dataset.

Model	IBM-Rank-30k				
	Pear.	**Spear.**	**TAU**	**MAE ↓**	**NDCG@15**
Llama	0.6103	0.5658	0.4035	0.1343	0.9252
CI-Llama	0.6178	0.5738	0.4095	0.1322	**0.9555**
CompAQA-Llama (Ours)	**0.6270**	**05881**	**0.4190**	**0.1313**	0.9521

5.4. Hyperparameter Analysis

When constructing comparison pairs, CompAQA relies on two important hyperparameters: m and l. Here, m represents the number of reference arguments sampled for each target argument, while l controls the score range during sampling. We conducted separate experimental analyses on different values of m and l, with the results presented in Tables 6 and 7, respectively. The results indicate that CompAQA is not particularly sensitive to changes in the hyperparameters m and l, as the performance fluctuations were relatively small. Based on these results, setting m to 3 and l to 0.05 appears to be a good choice for overall performance.

Table 6. The performance of CompAQA-DeBERTa on the IBM-Rank-30k dataset under the influence of different l in Equation (1). We conducted this experiment on DeBERTa, as it was the best-performing pre-trained model in the main experiment (Table 2).

CompAQA-DeBERTa	IBM-Rank-30k				
	Pear.	**Spear.**	**TAU**	**MAE ↓**	**NDCG@15**
$l = 0.025$	**0.5810**	0.5356	0.3780	**0.1292**	0.9461
$l = 0.05$	0.5797	**0.5373**	**0.3794**	0.1371	**0.9500**
$l = 0.075$	0.5776	0.5334	0.3763	0.1323	0.9496
$l = 0.1$	0.5781	0.5342	0.3768	0.1309	0.9448

Table 7. The performance of CompAQA-DeBERTa on the IBM-Rank-30k dataset under the influence of different m, with the number of reference arguments corresponding to each target.

CompAQA-DeBERTa	IBM-Rank-30k				
	Pear.	**Spear.**	**TAU**	**MAE ↓**	**NDCG@15**
$m = 2$	0.5739	0.5313	0.3749	0.1320	**0.9501**
$m = 3$	**0.5797**	**0.5373**	**0.3794**	0.1371	0.9500
$m = 4$	0.5770	0.5323	0.3759	0.1281	0.9308
$m = 5$	0.5768	0.5335	0.3765	**0.1267**	0.9432

5.5. Threats to Validity

Despite the efficacy of our proposed method, it is important to acknowledge certain limitations:

- Generalizability: When applied to new datasets, our method may require parameter-tuning to achieve optimal performance. Specifically, the values of l and m might need adjusting to accommodate different data characteristics.
- Computational Complexity: Our method necessitates comparing a target argument with multiple reference arguments to predict its quality score. This multi-comparison approach, while effective, inherently demands a higher computational cost.

These limitations present opportunities for future research, potentially focusing on developing more robust methods and optimizing the computational efficiency of the comparison process.

6. Conclusions

In this paper, we propose CompAQA, a comparison-based framework for argument quality assessment. This framework evaluates the quality of target arguments through multiple pairwise comparisons. Additionally, we propose an argument order-based data augmentation strategy to enhance the comparison ability of CompAQA. The effectiveness of CompAQA is validated on two argument quality ranking datasets and one pairwise argument quality classification dataset. Notably, CompAQA achieves promising results, regardless of whether it is based on encoder-only or decoder-only pre-trained models.

Author Contributions: Conceptualization, J.B., B.J. and R.X.; methodology, J.B., B.J. and Y.Z.; software, J.B., B.J. and Y.H.; supervision, R.X.; writing—original draft, J.B. and B.J.; writing—review and editing, J.B., Y.S. and R.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Key Research and Development Program of China (2022YFF0902100), the National Natural Science Foundation of China (62176076), Natural Science Foundation of Guangdong (2023A1515012922), the Shenzhen Foundational Research Funding (JCYJ20210324115614039, JCYJ20220818102415032), the Major Key Project of PCL (PCL2021A06), Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

Data Availability Statement: The data used in this study are publicly available.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Stede, M.; Schneider, J.; Hirst, G. *Argumentation Mining*; Springer: Berlin/Heidelberg, Germany, 2019.
2. Lawrence, J.; Reed, C. Argument Mining: A Survey. *Comput. Linguist.* **2019**, *45*, 765–818. [CrossRef]
3. Vecchi, E.M.; Falk, N.; Jundi, I.; Lapesa, G. Towards Argument Mining for Social Good: A Survey. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, 1–6 August 2021; pp. 1338–1352.
4. Ye, Y.; Teufel, S. End-to-End Argument Mining as Biaffine Dependency Parsing. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, 19–23 April 2021; pp. 669–678.
5. Morio, G.; Ozaki, H.; Morishita, T.; Yanai, K. End-to-end Argument Mining with Cross-corpora Multi-task Learning. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 639–658. [CrossRef]
6. Bao, J.; He, Y.; Sun, Y.; Liang, B.; Du, J.; Qin, B.; Yang, M.; Xu, R. A Generative Model for End-to-End Argument Mining with Reconstructed Positional Encoding and Constrained Pointer Mechanism. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 10437–10449.
7. Jo, Y.; Bang, S.; Reed, C.; Hovy, E.H. Classifying Argumentative Relations Using Logical Mechanisms and Argumentation Schemes. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 721–739. [CrossRef]

8. Sun, Y.; Liang, B.; Bao, J.; Yang, M.; Xu, R. Probing Structural Knowledge from Pre-trained Language Model for Argumentation Relation Classification. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 3605–3615.
9. Saadat-Yazdi, A.; Pan, J.Z.; Kökciyan, N. Uncovering Implicit Inferences for Improved Relational Argument Mining. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, 2–6 May 2023; pp. 2476–2487.
10. Schiller, B.; Daxenberger, J.; Gurevych, I. Aspect-Controlled Neural Argument Generation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6–11 June 2021; pp. 380–396.
11. Saha, S.; Srihari, R.K. ArgU: A Controllable Factual Argument Generator. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 8373–8388.
12. Alshomary, M.; Wachsmuth, H. Conclusion-based Counter-Argument Generation. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, 2–6 May 2023; pp. 957–967.
13. Gretz, S.; Friedman, R.; Cohen-Karlik, E.; Toledo, A.; Lahav, D.; Aharonov, R.; Slonim, N. A Large-Scale Dataset for Argument Quality Ranking: Construction and Analysis. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, 7–12 February 2020; pp. 7805–7813.
14. Marro, S.; Cabrio, E.; Villata, S. Graph Embeddings for Argumentation Quality Assessment. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 4154–4164.
15. Joshi, O.; Pitre, P.; Haribhakta, Y. ArgAnalysis35K: A large-scale dataset for Argument Quality Analysis. In Proceedings of the the 61st Annual Meeting of the Association for Computational Linguistics, ACL 2023, Toronto, ON, Canada, 9–14 July 2023; Volume 1: Long Papers, pp. 13916–13931.
16. Stab, C.; Gurevych, I. Annotating Argument Components and Relations in Persuasive Essays. In Proceedings of the COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 1501–1510.
17. Stab, C.; Gurevych, I. Recognizing Insufficiently Supported Arguments in Argumentative Essays. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, 3–7 April 2017; Volume 1: Long Papers, pp. 980–990.
18. Wachsmuth, H.; Potthast, M.; Khatib, K.A.; Ajjour, Y.; Puschmann, J.; Qu, J.; Dorsch, J.; Morari, V.; Bevendorff, J.; Stein, B. Building an Argument Search Engine for the Web. In Proceedings of the 4th Workshop on Argument Mining, ArgMining@EMNLP 2017, Copenhagen, Denmark, 8 September 2017; pp. 49–59.
19. Persing, I.; Ng, V. Modeling Thesis Clarity in Student Essays. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Sofia, Bulgaria, 4–9 August 2013; Volume 1: Long Papers, pp. 260–269.
20. Ding, Y.; Bexte, M.; Horbach, A. Score It All Together: A Multi-Task Learning Study on Automatic Scoring of Argumentative Essays. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 13052–13063.
21. Toledo, A.; Gretz, S.; Cohen-Karlik, E.; Friedman, R.; Venezian, E.; Lahav, D.; Jacovi, M.; Aharonov, R.; Slonim, N. Automatic Argument Quality Assessment—New Datasets and Methods. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019; pp. 5624–5634.
22. Slonim, N.; Bilu, Y.; Alzate, C.; Bar-Haim, R.; Bogin, B.; Bonin, F.; Choshen, L.; Cohen-Karlik, E.; Dankin, L.; Edelstein, L.; et al. An autonomous debating system. *Nature* **2021**, *591*, 379–384. [CrossRef]
23. Wachsmuth, H.; Naderi, N.; Hou, Y.; Bilu, Y.; Prabhakaran, V.; Thijm, T.A.; Hirst, G.; Stein, B. Computational Argumentation Quality Assessment in Natural Language. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, 3–7 April 2017; Volume 1: Long Papers, pp. 176–187.
24. Wachsmuth, H.; Werner, T. Intrinsic Quality Assessment of Arguments. In Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), 8–13 December 2020; pp. 6739–6745.
25. Habernal, I.; Gurevych, I. Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; Volume 1: Long Papers, p. 2016.
26. Gienapp, L.; Stein, B.; Hagen, M.; Potthast, M. Efficient Pairwise Annotation of Argument Quality. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 5772–5781.
27. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1: (Long and Short Papers), pp. 4171–4186.
28. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.

29. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020, Demos, Online, 16–20 November 2020; pp. 38–45.
30. Clark, D.B.; Sampson, V.D. Analyzing the quality of argumentation supported by personally-seeded discussions. In Proceedings of the Next 10 Years! Proceedings of the 2005 Conference on Computer Support for Collaborative Learning, CSCL '05, Taipei, Taiwan, 30 May–4 June 2005; pp. 76–85.
31. Wachsmuth, H.; Stein, B.; Ajjour, Y. "PageRank" for Argument Relevance. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, 3–7 April 2017; Volume 1: Long Papers, pp. 1117–1127.
32. Swanson, R.; Ecker, B.; Walker, M.A. Argument Mining: Extracting Arguments from Online Dialogue. In Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Prague, Czech Republic, 2–4 September 2015; pp. 217–226.
33. Wachsmuth, H.; Khatib, K.A.; Stein, B. Using Argument Mining to Assess the Argumentation Quality of Essays. In Proceedings of the COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 1680–1691.
34. Johnson, R.H.; Blair, J.A. *Logical Self-Defense*; Mcgraw-Hill: Toronto, ON, Canada, 1977.
35. Hamblin, C.L. Fallacies. *Tijdschr. Voor Filos.* **1970**, *33*, 183–188.
36. Lauscher, A.; Ng, L.; Napoles, C.; Tetreault, J.R. Rhetoric, Logic, and Dialectic: Advancing Theory-based Argument Quality Assessment in Natural Language Processing. In Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), 8–13 December 2020; pp. 4563–4574.
37. Persing, I.; Ng, V. Modeling Argument Strength in Student Essays. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, Beijing, China, 26–31 July 2015; Volume 1: Long Papers, pp. 543–552.
38. Skitalinskaya, G.; Klaff, J.; Wachsmuth, H. Learning From Revisions: Quality Assessment of Claims in Argumentation at Scale. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, 19–23 April 2021; pp. 1718–1729.
39. Favreau, C.; Zouaq, A.; Bhatnagar, S. Learning to Rank with BERT for Argument Quality Evaluation. In Proceedings of the Thirty-Fifth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2022, Hutchinson Island, Jensen Beach, FL, USA, 15–18 May 2022.
40. Wang, Y.; Chen, X.; He, B.; Sun, L. Contextual Interaction for Argument Post Quality Assessment. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, 6–10 December 2023; pp. 10420–10432.
41. Falk, N.; Lapesa, G. Bridging Argument Quality and Deliberative Quality Annotations with Adapters. In Proceedings of the Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, 2–6 May 2023; pp. 2424–2443.
42. Fromm, M.; Berrendorf, M.; Faerman, E.; Seidl, T. Cross-Domain Argument Quality Estimation. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 13435–13448.
43. Zhang, S.; Shen, Y.; Tan, Z.; Wu, Y.; Lu, W. De-Bias for Generative Extraction in Unified NER Task. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, 22–27 May 2022; Volume 1: Long Papers, pp. 808–818.
44. Hu, M.; Wu, Y.; Gao, H.; Bai, Y.; Zhao, S. Improving Aspect Sentiment Quad Prediction via Template-Order Data Augmentation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 7889–7900.
45. Gou, Z.; Guo, Q.; Yang, Y. MvP: Multi-view Prompting Improves Aspect Sentiment Tuple Prediction. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 4380–4397.
46. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
47. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014; A meeting of SIGDAT, a Special Interest Group of the ACL; pp. 1532–1543.
48. Huang, Y.; Fang, M.; Cao, Y.; Wang, L.; Liang, X. DAGN: Discourse-Aware Graph Network for Logical Reasoning. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6–11 June 2021; pp. 5848–5855.
49. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, 25–29 April 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.