ArgInstruct: Specialized Instruction Fine-Tuning for Computational Argumentation

Maja Stahl*
Leibniz University Hannover
m.stahl@ai.uni-hannover.de

Timon Ziegenbein*
Leibniz University Hannover
t.ziegenbein@ai.uni-hannover.de

Joonsuk Park†
University of Richmond
park@joonsuk.org

Henning Wachsmuth†
Leibniz University Hannover
h.wachsmuth@ai.uni-hannover.de

Abstract

Training large language models (LLMs) to follow instructions has significantly enhanced their ability to tackle unseen tasks. However, despite their strong generalization capabilities, instruction-following LLMs encounter difficulties when dealing with tasks that require domain knowledge. This work introduces a *specialized instruction fine-tuning* for the domain of computational argumentation (CA). The goal is to enable an LLM to effectively tackle any unseen CA tasks while preserving its generalization capabilities. Reviewing existing CA research, we crafted natural language instructions for 105 CA tasks to this end. On this basis, we developed a CA-specific benchmark for LLMs that allows for a comprehensive evaluation of LLMs' capabilities in solving various CA tasks. We synthesized 52k CA-related instructions, adapting the self-instruct process to train a CA-specialized instruction-following LLM. Our experiments suggest that CA-specialized instruction fine-tuning significantly enhances the LLM on both seen and unseen CA tasks. At the same time, performance on the general NLP tasks of the SuperNI benchmark remains stable.

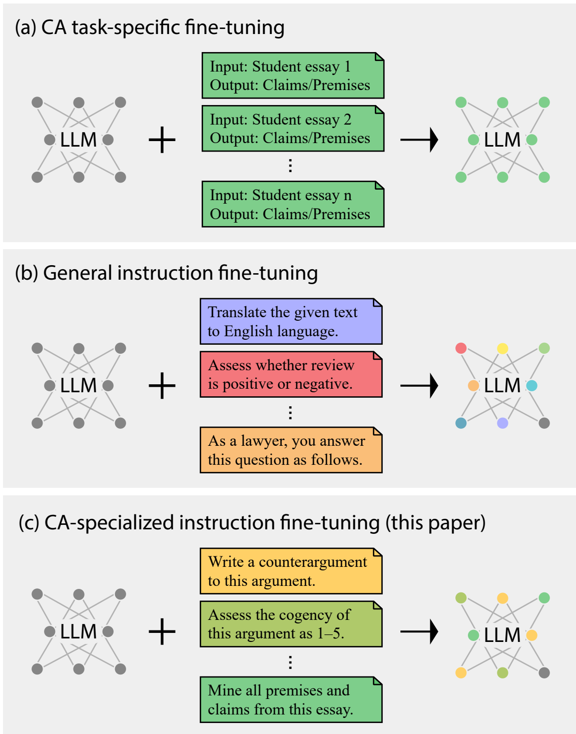

Figure 1: Comparison of fine-tuning methods: (a) Optimizing an LLM for a CA task on input-output pairs. (b) Making an LLM instruction-following on highly diverse tasks. (c) Our method: Making an LLM an instruction-following CA specialist on diverse CA-specific tasks.

1 Introduction

Large language models (LLMs) have proven effective for various NLP tasks, including several tasks from computational argumentation (CA), the computational analysis and synthesis of natural language arguments (Chen et al., 2024a). Initially, it was common to fine-tune pretrained LLMs on input-output pairs for a task (Devlin et al., 2019; Radford et al., 2019). Figure 1(a) illustrates such *task-specific fine-tuning* for the mining of claims and premises from student essays. In contrast, recent LLMs are often *instruction fine-tuned* by exposing them to highly diverse tasks[1] (Ouyang et al., 2022; Taori et al., 2023), as shown in Figure 1(b). This enables them to generate responses aligned with specific task requirements described in the instruction (Wang et al., 2022, 2023).

However, despite their strong generalization abilities, instruction-following LLMs often struggle to solve tasks that require domain knowledge (Lecler et al., 2023; Castro Nascimento and Pimentel, 2023; Yang et al., 2023). This limitation results from the principle of general instruction fine-tuning to prioritize generalizability over specialization. It affects CA tasks in particular, as they often center around

[*] Equal contribution.
[†] Equal advising.
[1] Here, the term *task* refers to a natural language instruction along with one or more input-output pairs that provide contextual guidance (Mishra et al., 2022; Wang et al., 2022).

Findings of the Association for Computational Linguistics: ACL 2025, pages 11103–11127
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

sophisticated context-related concepts from argumentation theory (Wachsmuth et al., 2024).

Specifically, CA research in NLP focuses on the mining, assessment, and generation of natural language arguments (Stede and Schneider, 2018). Despite recent advancements in LLMs, tackling CA tasks remains challenging (Chen et al., 2024a) due to their context-dependent specificities (e.g., in newspaper articles vs. social media posts) (Habernal et al., 2014) and their subjectivity (e.g., in assessing argument quality) (Wachsmuth et al., 2017; Romberg, 2022). In fact, providing context- and argumentation-specific knowledge, such as details about the debate setting and definitions of argumentative concepts, has been stressed to be important for task performance (Lauscher et al., 2022).

In this paper, we study the impact of conflating the two learning paradigms of task-specific fine-tuning and general instruction fine-tuning. That is, we introduce the idea of *specialized instruction fine-tuning* by combining a highly diverse set of general tasks with a diverse set of tasks specific to a given task domain, CA in our case. The goal is to obtain an LLM that is highly proficient in CA while being agnostic to the particular CA task it encounters and maintaining generalization capabilities.

We hypothesize that providing an LLM with argumentation-specific knowledge during instruction fine-tuning in a way that enables joint learning of representations across tasks is key to addressing the limitations of both general-purpose and task-specific LLMs in this domain. Through *CA-specialized instruction fine-tuning* (Figure 1(c)), we enhance the LLM's ability to mine argument structure, assess argument quality, and generate arguments across CA contexts, ensuring both accuracy and versatility for the full spectrum of argumentation. We expect this to require many and diverse CA tasks, exceeding what can be achieved by merely combining existing tasks.

Towards the outlined goal, we create a large CA-specific instruction fine-tuning dataset. Starting from 105 seed tasks, derived from a total of 30 argumentation corpora, we follow the self-instruct process of Wang et al. (2023) to automatically generate a diverse set of 52k CA-specific tasks (instructions plus input-output pairs). By combining these tasks with general instruction fine-tuning data in various ways, we train instruction-following Gemma (Gemma Team et al., 2024) variants for CA.

The seed tasks serve as a new CA benchmark. Our experiments suggest that our specialized instruction fine-tuning method (dubbed *ArgInstruct*) successfully generalizes toward unseen CA tasks, outperforming a wide range of competitive instruction-following LLMs in a zero-shot setting. Moreover, we demonstrate on SuperNI (Wang et al., 2022) that the LLM's general instruction-following abilities remain despite specialization.

Altogether, this paper's main contributions are:

- A general method for specialized instruction fine-tuning, instantiated for CA
- An extensive dataset for CA-specific instruction fine-tuning and benchmarking of LLMs
- Empirical evidence that our CA-specialized instruction-finetuning effectively enhances an LLM's generalizability for unseen CA tasks[2]

2 Related Work

The computational analysis and synthesis of arguments in natural language, often referred to as computational argumentation (CA), has its roots in a long history of philosophical research (Aristotle, ca. 350 B.C.E./ translated 2007), which has gained significant attention from the NLP community in recent years. The three main CA research areas frequently covered are argument mining (Park and Cardie, 2014; Boltužić and Šnajder, 2014; Stab and Gurevych, 2017a), argument assessment (Persing and Ng, 2015; Wachsmuth et al., 2017; Gretz et al., 2020), and argument generation (Syed et al., 2021; Schiller et al., 2021; Wachsmuth et al., 2018a).

Although the contributions to each area are plentiful, most works focus on one or a few tasks within or across the areas. Many methods rely on supervised learning and single-domain datasets, limiting generalizability (Waldis et al., 2024). Recently, Chen et al. (2024a), Elaraby et al. (2024), Rescala et al. (2024), and Cabessa et al. (2025) studied the potential of LLMs to tackle a selection of mining, assessment, and generation tasks. They obtained promising results, showing that LLMs can address multiple CA tasks, sometimes even without explicit training. Beyond that, Wachsmuth et al. (2024) propose to systematically instruct LLMs for argument quality assessment with argumentation-specific knowledge to enable knowledge sharing across tasks and contexts. Despite these advances, there remains a notable gap: no study has yet comprehensively evaluated LLMs across all three main

[2]Our dataset and experiment code can be found under: https://github.com/webis-de/ACL-25

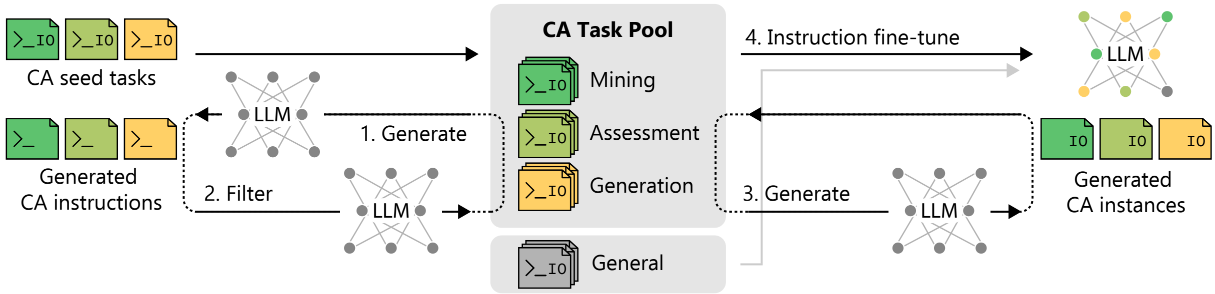

Figure 2: Overview of our methodology: We manually craft CA-specific seed tasks and prompt an LLM to generate new CA-specific tasks in a loop by (1) generating new instructions, (2) filtering them for CA relevance and novelty, and (3) generating corresponding instances. (4) After postprocessing, the generated CA-specific tasks from the task pool are combined with existing general tasks to specialize an LLM for CA using instruction fine-tuning.

CA areas or operationalized a systematic framework such as the one suggested by Wachsmuth et al. (2024) to tackle CA tasks holistically.

To fill this gap, we use instruction fine-tuning as it allows training a task-agnostic LLM for CA. In general, instruction fine-tuning is just a supervised training process. The key is the data used for fine-tuning: By having instances with instructions, and by diversifying these instances as much as possible, the LLM learns that task specificities should be abstracted from while instructions should always be followed (Wang et al., 2024a).

Over the last years, multiple instruction fine-tuning datasets have been collecting from existing NLP tasks (Mishra et al., 2022; Wang et al., 2022). The datasets usually consist of natural language instructions and example instances, which are either written manually by humans (Sanh et al., 2021; Ouyang et al., 2022; Longpre et al., 2023) or created synthetically (Honovich et al., 2023; Wang et al., 2023; Taori et al., 2023; Chen et al., 2024b). The importance of data diversity and selection was further emphasized by Bukharin and Zhao (2023), Li et al. (2024), and Wang et al. (2024a). Besides evaluating instruction following capabilities in terms of text generation performance on unseen tasks (Chia et al., 2024; Dubois et al., 2024), new ways of assessment were developed in which LLMs compete against each other (Zheng et al., 2023; Chiang et al., 2024), sometimes even replacing human annotators as evaluators (Zheng et al., 2023).

Several instruction-following LLMs have been developed and evaluated on these benchmarks (Ouyang et al., 2022; Muennighoff et al., 2023; Chung et al., 2024). However, standard instruction fine-tuning is purely generalization-oriented, whereas we aim to balance between generalization and CA specialization. By changing what data is used, the LLM learns that not all task-specificities should be abstracted from, but the domain specialization should be kept while following instructions. This makes the LLM a computational argumentation expert, rather than a general talk solver as in standard instruction fine-tuning.

Our method builds on the ideas of *Self-Instruct* (Wang et al., 2023) and *Alpaca* (Taori et al., 2023). Both generate 52k tasks starting from a small set of manually-written seed tasks, showing that larger amounts of diverse instructions can help improve the performance of LLMs. However, we do not focus on or start from general instructions but CA-specific ones. Hence, we combine the CA-specific knowledge of the structure, quality, and writing of arguments with meta-knowledge of how to solve tasks acquired by following general instructions.

3 Methodology

This section presents our methodology for the training of a specialized LLM for computational argumentation (CA). We start by deriving a seed set of instruction fine-tuning tasks from existing CA tasks and datasets. This manually-annotated seed data serves as a reliable basis for generating a large set of diverse CA-specific tasks. Combining these tasks with general NLP tasks, we then specialize an LLM using instruction fine-tuning, dubbed *ArgInstruct* (*Arg*umentation-specialized *Instruct*ion Fine-Tuning). Whereas standard instruction fine-tuning is, by concept, fully generalization-oriented, we diversify instructions only within the task domain. The CA-specific instructions enable the LLM to deal specifically with any task from computational argumentation. By still mixing in general instruction fine-tuning data, we further achieve that generalization capabilities are widely preserved. Figure 2 illustrates the methodology.

3.1 Task Generation

Following instruction fine-tuning literature (Mishra et al., 2022; Wang et al., 2022, 2023) we define a task $T = (I, S)$ to consist of a natural language instruction I and $m \geq 1$ input-output instances $S = \{x_j, y_j\}_{j=1}^{m}$. To instruction fine-tune a specialized LLM for CA, we propose to create a large instruction fine-tuning dataset $\mathcal{T} = \{T_1, \ldots, T_n\}$ containing a large set of CA-specific but diverse tasks by (1) generating new instructions, (2) filtering for CA relevance and diversity, and (3) generating corresponding input-output instances.

Instruction Generation Building on research in CA, we curate a collection of seed tasks, $\mathcal{T}_0$, across the three main research areas of CA: argument mining, argument assessment, and argument generation (see Section 4.1). First, we manually craft a set of $k \gg 0$ natural language instructions $\mathcal{I}_0 = \{I_1, \ldots, I_k\}$ by extracting task and term definitions from the papers and annotation guidelines. For each I_j, we obtain the input-output instances from the corresponding datasets to construct $\mathcal{T}_0$.

Following self-instruct (Wang et al., 2023), we use $\mathcal{T}_0$ as the initial CA task pool, and generate new CA-specific instructions $\mathcal{I}_i$ using a pretrained LLM. Self-instruct is an iterative process that uses prompting to create an instruction fine-tuning dataset, where an LLM generates new training data by leveraging its own previous outputs as few-shot examples. In each generation step i, we randomly sample a subset $\tilde{\mathcal{I}}$ of size $l \geq 1$ from the instructions $\mathcal{I}_0 \cup \mathcal{I}_{<i}$ in $\mathcal{T}_0$ where $\mathcal{I}_{<i}$ contains all instructions generated previously (in Section 4, we set $l = 8$). $\tilde{\mathcal{I}}$ is used as few-shot examples to generate a new set of instructions $\mathcal{I}_i$. To ensure that the instructions in $\mathcal{I}_{<i}$ remain focused on CA, we modify the self-instruct prompt to: "Come up with a series of *computational argumentation* tasks."

Instruction Filtering To filter the generated instructions $\mathcal{I}_i$ by CA relevance, we utilize the same LLM using the prompt "Does the following task fall into the field of computational argumentation?" together with few-shot examples. For this, we randomly sample CA-specific examples $\mathcal{I}_+$ from our seed instructions $\mathcal{I}_0$ as positive examples and not CA-specific instructions $\mathcal{I}_-$ from established instruction following datasets as negative examples and append them to the prompt. Following the instruction filtering used in self-instruct (Wang et al., 2023), a generated instruction that has been deemed CA-relevant is added to the CA task pool only when its ROUGE-L F_1-score similarity with any existing instruction from $\mathcal{I}_0 \cup \mathcal{I}_{<i}$ falls below a predefined threshold τ (we use $\tau = 0.7$ below). The instruction generation and filtering process is repeated until we reach a predefined number of generated instructions. In Section 5, we set this number to 52,445, roughly matching Taori et al. (2023).

Instance Generation To obtain a task $T = (I, S)$ for each generated instruction $I \in \mathcal{I}_{<i}$, we mostly follow Wang et al. (2023), using the LLM to first identify the task type for I and then generating the corresponding instances S based on the task type. Duplicates in S and instances with the same input but different outputs are filtered out.

Beyond the two task types *generation* and *classification* covered by Wang et al. (2023), we further distinguish *regression* tasks to enable more fine-grained evaluation with type-specific metrics. We identify the task types for each generated I using templated prompts with instructions sampled from $\mathcal{I}_0$ as examples for each task type. Instances are also generated using a templated prompt, providing tasks from $\mathcal{T}_0$ representative of each task type.

3.2 LLM Instruction Fine-Tuning

To create our ArgInstruct model, we fine-tune a pretrained LLM on both the entire CA task pool and general tasks, aiming to specialize in CA while maintaining the generalization idea of instruction fine-tuning. We format task instances into a prompting template for training and mask input tokens during cross-entropy loss calculation, focusing solely on the generated output tokens to help the model retain pre-trained input interpretation skills while ensuring accurate output generation (i.e., mapping from input to output).

4 Data

This section first details the selection process for CA seed datasets and tasks that serve as the basis for ArgInstruct. Then, we present the resulting CA-specific instruction fine-tuning dataset, which integrates the seed tasks and newly generated tasks.

4.1 Task Generation using ArgInstruct

CA Seed Dataset Selection Initially, we manually collected 71 CA datasets from CA literature. We started from datasets used in shared tasks of the Argument Mining workshop series (6), and extended them based on CA papers from the survey of

	Source	Text Genre	Tasks Covered by the Dataset	# Tasks
Argument Mining	Boltužić and Šnajder (2014)	online user comments	relation type classification	1
	Peldszus and Stede (2015)	short argumentative texts	claim extraction, relation identific., function classific.	4
	Stab and Gurevych (2017a)⋆	student essays	argumentative span/relation identific., stance classific.	4
	Habernal and Gurevych (2017)	user-generated web content	persuasiveness detection, toulmin component extract.	2
	Stab et al. (2018)	diverse web documents	supporting/opposing argument detection	1
	Reimers et al. (2019)	web crawl sentences	argument similarity prediction	1
	Poudyal et al. (2020)	court decisions	clause/premise/conclusion recognition, relation predict.	4
	Hautli-Janisz et al. (2022)⋆	broadcast political debates	propositional/illocutionary relation identification	2
	Chen et al. (2022)	amazon reviews	unit segmentation/classific., helpfulness/relation predict.	4
	Kuznetsov et al. (2022)⋆	peer reviews	pragmatic category tagging	1
Argument Assessment	Persing and Ng (2015)	student essays	argument strength prediction	1
	Habernal and Gurevych (2016a)	online debates	reason for convincingness prediction	18
	Abbott et al. (2016)⋆	online debates	agreement/attack/emotion/hostility/sarcasm prediction	5
	Wachsmuth et al. (2017)	online debates	argument quality rating, argumentativeness detection	15
	Habernal et al. (2018a)	newspaper editorials	warrant selection (argument reasoning comprehension)	1
	Gretz et al. (2020)⋆	crowd-sourced arguments	argument quality rating, stance prediction	2
	Friedman et al. (2021)	crowd-sourced arguments	key point generation/matching	2
	Stein et al. (2021)	online debates	same side stance classification	1
	Heinisch et al. (2022)	online political debates	(relative) novelty/validity classification	4
	Ziegenbein et al. (2023)	reviews, Q&A, debates	inappropriateness (reason) classification	14
Argument Generation	Hasan and Ng (2014)	ideological online debates	reason identification	1
	Skeppstedt et al. (2018)	short argumentative texts	argument generation	1
	Wachsmuth et al. (2018a)⋆	short argumentative texts	argument synthesis	1
	Wachsmuth et al. (2018b)	online debates	counter argument generation	4
	Roush and Balaji (2020)⋆	competitive formal debates	extractive debate summarization	1
	Schiller et al. (2021)⋆	diverse web documents	aspect-based generation	1
	Skitalinskaya et al. (2021)	online debates	suboptimal claim detection/improvement	4
	Syed et al. (2021)⋆	online debates	conclusion generation	1
	Alshomary et al. (2021)	online debates	belief-based generation, stance prediction	2
	Stahl et al. (2023)	learner essays	enthymeme reconstruction, enthymeme detection	2

Table 1: Overview of the 30 selected CA seed datasets, categorized into argument mining, argument assessment, and argument generation. The table includes the corresponding paper, text genre, and the kinds and numbers of extracted CA seed tasks. The tasks from the 9 CA datasets marked with "⋆" are reserved for testing.

Lauscher et al. (2022) (30), as well as more recent datasets found through searches in the ACL Anthology and Google Scholar (35). We then categorized each dataset into one or more CA areas based on their usage in the literature: argument mining, argument assessment, and argument generation.[3] For each subarea, we selected a subset of ten datasets, consisting of the five most cited datasets along with five additional, lesser-known datasets that cover a diverse range of data sources. This process ensures comprehensive coverage of CA and results in the 30 seed datasets listed in Table 1. The list of all 71 datasets can be found in Appendix A.

CA Seed Task Collection Given the seed data, we obtained the set of CA seed tasks, $\mathcal{T}_0$, from the respective papers. Following our task definition, we consider a task $T = (S, I)$ to model a relation S between inputs and outputs, which can be described in natural language in the form of an instruction I. Intermediate steps (e.g., feature extraction or data preprocessing) are not seen as individual CA tasks. If papers contained multiple tasks (e.g., mining and relation identification), each task was treated separately. We wrote the corresponding instructions based on annotation guidelines or, when unavailable, the task descriptions in the papers. To ensure tasks are self-contained, we included relevant term definitions (e.g., class definitions) in the instructions (examples in Appendix B). In total, we obtained a set of 105 seed tasks, $\mathcal{T}_0 = \{T_1, \ldots, T_{105}\}$, from the 30 seed datasets.

We categorized all see tasks into three task types: *classification*, *regression*, and *generation*. There is a noteworthy connection between task types and CA subareas: argument mining tasks are typically categorized as classification, argument assessment tasks as regression, and argument generation tasks as generation. However, exceptions exist; for instance, argument mining can be modeled as a classification task at the sentence level or as a generation task at the span level.

[3]We categorized datasets containing tasks from multiple subareas based on their main focus and then the task modeling.

CA Task Generation Based on the 105 seed tasks, we generated 52,445 additional CA tasks. By roughly matching the size of the general dataset of Taori et al. (2023), we ensure comparability to their results. Concretely, we used *Meta-Llama-3-70B* as the LLM to generate new CA instructions (step 1 in Figure 2), filter them by CA relevance (step 2), and generate instances (step 3).

4.2 The ArgInstruct Dataset

Table 2 compares the statistics for the seed tasks and the generated CA tasks. It can be seen that the seed tasks cover rather few instructions (105) but many instances (4.5M), while the generated tasks have many diverse instructions (52,445) with a single instance each. Classification (30,204) and generation (20,071) dominate the generated tasks, with fewer regression tasks (2,170), likely due to LLMs' limited exposure to regression tasks. Although the generated instructions are shorter on average (28.2 vs. 48.1 words), input lengths are similar (50.7 vs. 64.3). The longer output length (25.2 vs. 7.7) in the generated data likely stems from the higher proportion of generation tasks.

Specialization To assess whether the generated and filtered instructions are indeed CA-specific, we follow Wang et al. (2023) and extract the root verb and its first direct noun object for each generated instruction using the Berkeley Neural Parser (Kitaev et al., 2019). Figure 3 shows the 20 most common root verbs and their four most common direct noun objects, which constitute 14% of all generated instructions. Overall, we see that most direct noun objects are indeed argumentation-related, e.g., "argumentation", "(counter-)argument" and "claim".

Diversity To analyze the diversity of the generated instructions, we compute the ROUGE-L F_1 similarity between each generated and seed instruction. The average similarity to the closest seed instruction is 0.28, with a maximum of 0.70, which was the similarity threshold τ taken from Wang et al. (2023). Examples of generated instructions within the 10% closest to seed instructions are:

I_1: "Extract the central claim from the following argumentative text and predict its stance (pro or con) with respect to the given topic."

I_2: "Determine which of these statements is true: comment 1 attacks argument 2. comment 1 supports argument 2. comment 1 makes no use of argument 2."

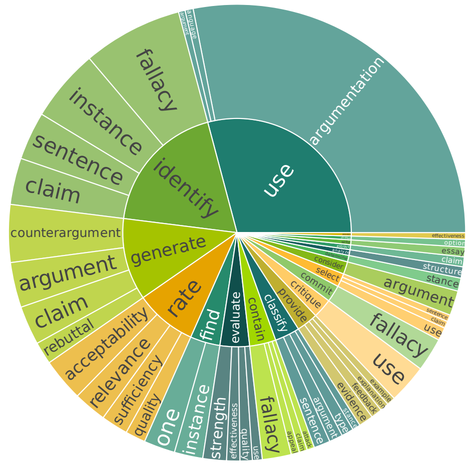

Figure 3: The 20 most common root verbs (inner circle) and their top four direct noun objects (outer circle) in our generated instructions highlight their CA focus.

While these instructions are close to existing CA tasks, namely thesis extraction and stance detection (I_1), and relation detection (I_2), they introduce new wordings that will likely lead to more robust fine-tuning. Exemplary instructions with the lowest maximal similarity to the seed instructions are:

I_3: "For each elementary unit x, choose one proposition y such that there exists a support relation between x and y."

I_4: "Generate a list of possible explanations for why someone might believe something based on their background and experiences."

I_3 is a form of argumentative relation extraction, while I_4 is an argument generation task that, to our knowledge, has not yet been studied. The final task involves identifying various fallacies, which are related to CA but were not covered by the seed tasks. For more examples, see Appendix C.

Quality To evaluate the quality of the generated tasks, two authors of this paper manually evaluated a random sample of 200 generated tasks. The annotation protocol was adapted from the questionnaire of Wang et al. (2023), assessing three key aspects:

Q_1: Does the instruction describe a valid CA task?

Q_2: Is the input appropriate for the instruction?

Q_3: Is the output a correct and acceptable response to the instruction and input?

The tendencies of our results are similar to those reported by Wang et al. (2023), with 87% of the instructions corresponding to valid CA tasks (Q_1),

	# Instructions				# Instances				Average Length		
	Classif.	**Regr.**	**Generat.**	**Total**	**Classif.**	**Regr.**	**Generat.**	**Total**	**Instruct.**	**Input**	**Output**
Seed tasks	60	23	22	105	3,997,517	86,870	465,652	4,550,039	48.08	64.28	7.70
Generated tasks	30,204	2,170	20,071	52,445	30,204	2,170	20,071	52,445	28.17	50.74	25.21

Table 2: Number of instructions and instances in total and per task type (classification, regression, generation), and the average length in words of instructions, non-empty inputs, and outputs for our CA seed and generated tasks.

75.5% of the input instances deemed appropriate for their instructions (Q_2), and 62.5% of the outputs satisfying all correctness criteria (Q_3). Hence, we conclude that, although the generated dataset contains some noise, most of the generated tasks are entirely or at least partially correct.

5 Evaluation

We now present our experiments to evaluate the impact of specialized instruction fine-tuning. LLM variants were trained on (a) CA seed tasks, (b) generated CA tasks, (c) general tasks, and (d) combinations thereof. We then assessed their performance on unseen instances of known CA tasks and their generalizability to entirely unseen CA tasks.

5.1 Experimental Setup

Data We reserved 21 seed tasks (20% of 105) from nine CA datasets as unseen test tasks (Table 1), balancing across argument mining, assessment, and generation. The remaining 84 seed tasks were split into training, validation, and test sets, using prior splits where available or a 7:1:2 random split otherwise. For evaluation, we sampled 100 test instances per task, balancing labels for classification and covering the full range for regression, while generation tasks were sampled randomly. The same sampling method was used to assemble 52,445 instances from the training seed tasks for our (*seedCA*) dataset.

Evaluation We use guided generation (Willard and Louf, 2023) for classification and regression tasks and *open* generation (up to 512 new tokens) for generation tasks. For classification and regression, a finite state machine decodes model outputs for direct comparison with ground-truth values.

Metrics To ensure meaningful comparisons, we use task-specific evaluation metrics. For classification, we report the micro-averaged F_1-score on balanced test sets. For regression, we use mean absolute scaled error (MASE), which normalizes errors with respect to a naïve mean baseline, with scores less than 1 indicating better performance and above 1 indicating worse (Hyndman and Athanasopoulos, 2021). For generation, we report the ROUGE-L F_1-score. Additionally, we calculate the mean rank for each model, reflecting its performance across datasets and tasks, providing a comprehensive measure of overall CA performance.

Models As base models, we use the recent LLM *Gemma-2-9B* (Gemma Team et al., 2024) and an instruction fine-tuned variant, *Gemma-2-9B-General*. The latter is obtained by fine-tuning *Gemma-2-9B* on the general instruction dataset of Taori et al. (2023), ensuring that it has not been exposed to the SuperNI benchmark for general instruction following (Wang et al., 2022). Both models are then instruction fine-tuned on equally-sized combinations of seed CA tasks, generated CA tasks, and general tasks (52k instances each). Details on hyperparameter tuning can be found in Appendix D.

5.2 Results of ArgInstruct on CA Tasks

Table 3 reports the CA performance of the two base LLMs and the LLMs trained with *ArgInstruct* and its ablations, averaged across datasets. We evaluate performance on (a) unseen test instances of the training tasks and (b) test instances of the completely unseen test tasks. Note that for the base models, all instances and tasks are unseen.

On unseen instances, LLMs fine-tuned on the respective training tasks (+*seedCA*) work best (ranks 2.3 and 2.7), as expected. For unseen tasks, combining *seedCA*, *genCA*, and *general* performs best with either base model (both rank 2.0), outperforming ablated variants. This strongly supports our hypothesis that specialized instruction fine-tuning enhances generalization on CA tasks. Whether general instruction fine-tuning occurs prior (*Gemma-2-9B-General+seedCA,genCA*) or alongside CA-specific fine-tuning (*Gemma-2-9B+seedCA,genCA,general*) has little effect on performance, although the latter performs slightly better on unseen instances. Appendix E reports the results of the latter variant on all 105 tasks.

	Fine-Tuning Data			(a) Unseen CA Instances				(b) Unseen CA Tasks			
Approach	**seedCA**	**genCA**	**general**	**F_1 ↑**	**MASE↓**	**R-L ↑**	**Rank↓**	**F_1 ↑**	**MASE↓**	**R-L ↑**	**Rank↓**
Gemma-2-9B (baseline)	○	○	○	.45	1.6	.39	9.0	.45	4.2	.15	10.7
+ seedCA	●	○	○	**.65**	1.1	**.50**	**2.3**	**.65**	2.5	.23	5.7
+ genCA	○	●	○	.51	2.7	.45	8.7	.52	3.0	.30	6.3
+ general	○	○	●	.50	2.1	.32	10.0	.51	2.6	.29	6.7
+ seedCA, genCA	◑	◑	○	.61	1.6	.49	4.0	.57	2.9	.26	5.7
+ genCA, general	○	◑	◑	.51	1.7	.39	8.7	.50	3.0	.17	8.3
+ seedCA, general	◑	○	◑	.59	**1.0**	.48	4.0	.60	2.3	.30	4.0
+ seedCA, genCA, general	◔	◔	◔	.61	1.5	.49	4.3	**.65**	2.5	**.32**	**2.0**
Gemma-2-9B-General (baseline)	○	○	◍	.48	2.3	.34	11.0	.50	2.1	.24	7.0
+ seedCA	●	○	◍	.64	1.2	.49	2.7	.63	2.1	**.32**	4.3
+ genCA	○	●	◍	.49	2.6	.44	9.3	.52	2.6	.30	6.7
+ seedCA, genCA (ArgInstruct)	◑	◑	◍	.57	1.3	.49	4.0	**.65**†	**1.9**†‡	.31†	**2.0**

Table 3: Main CA results on (a) unseen CA instances and (b) unseen CA tasks: Gemma-2-9B instruction fine-tuned on 52k instances of CA seed (+*seedCA*), generated CA (+*genCA*), general (+*general*) tasks and their combinations. The symbols represent the proportion of fine-tuning data coming from each source (○: 0%, ◔: 33%, ◑: 50%, ●: 100%). ◍ indicates prior use for general instruction fine-tuning. The best values are bold, the best per base model underlined. Overall, both full specialized instruction fine-tuning variants (bold) achieve the best mean *rank*. † and ‡ denote significant improvements over the baseline and +*seedCA* respectively (Wilcoxon signed-rank test, $p < .05$).

Approach	**F_1 ↑**	**R-L ↑**	**Rank ↓**
Gemma-2-9B	.55	**.43**	3.0
+ seedCA, genCA, general	.61	.40	3.5
Gemma-2-9B-General	**.62**	.37	2.5
+ seedCA, genCA (ArgInstruct)	**.62**	**.43**	**1.0**

Table 4: Generalization results on SuperNI: Zero-shot performance of *Gemma-2-9B*, its instruction fine-tuned variant *Gemma-2-9B-General*, and our CA-specialized LLMs. *ArgInstruct* performs best across all metrics.

5.3 Results of ArgInstruct on General Tasks

To assess if specialized instruction fine-tuning preserves generalization capabilities, Table 4 shows the performance of both base models and the full CA-specialized LLMs on the SuperNI benchmark of general NLP tasks (Wang et al., 2022). Unlike above, the timing of general instruction fine-tuning has an effect here: Our specialized LLM, fine-tuned on general instructions first (*Gemma-2-9B-General+seedCA,genCA*), performs best across all metrics. Given its strong results on both CA and general tasks, we designate it as our final model, now referred to as the *ArgInstruct* model.

5.4 Comparison to Task-Specific Fine-Tuning

To assess the actual strength of our *ArgInstruct* model, we compare it on specific CA tasks against the current state-of-the-art (SOTA) on the entire original test sets of six tasks – two each from argument mining, assessment, and generation (one seen during training and one entirely unseen). This reveals the trade-off between specializing in CA as a whole and developing task-specific approaches.

As shown in Table 5, the SOTA approaches win on all six tasks, but ArgInstruct achieves comparable performance in three of them. We speculate that the performance gap arises because the SOTA models (a) benefit from a larger number of task-specific training instances and/or (b) are able to better adjust to the single task and data source. Nonetheless, *ArgInstruct* is a strong and versatile model, offering broad generalization across CA tasks in a zero-shot setting. However, depending on the CA task, additional task-specific fine-tuning may further enhance the performance of *ArgInstruct* for optimal results.

5.5 Comparison to General LLMs

To understand the instruction-following abilities of our base model, we finally compare it against competitive instruction-following LLMs of similar size (Taori et al., 2023; Jiang et al., 2023; Gemma Team et al., 2024; Grattafiori et al., 2024; OpenAI et al., 2024). Table 6 shows the zero-shot performance of all models alongside *Majority* and *Random* baselines. ArgInstruct outperforms all others in terms of F_1 (.65). However, for regression tasks, no model proves reliable, as all MASE scores are worse than predicting the mean. In generation (R-L), *GPT-4o-mini* appears slightly superior to our model (.32 vs. .31), though its comparability may be limited due to its unknown size. Overall, *ArgInstruct* achieves the strongest result on unseen CA tasks, achieving the best mean rank (2.33) across all models.

Area	Task (Source)	Unseen	Metric	ArgInstruct	Task-Specific SOTA
Mining	Argument Detection & Classif. (Stab et al., 2018)	Instances	$F_1 \uparrow$	.62	.73 (Wang et al., 2024b)
	Relation Detection (Stab and Gurevych, 2017a)	Task	$F_1 \uparrow$	.47	.84 (Cabessa et al., 2025)
Assessment	Inappropriateness Classif. (Ziegenbein et al., 2023)	Instances	$F_1 \uparrow$	.74	.75 (Ziegenbein et al., 2023)
	Argument Quality Rating (Gretz et al., 2020)	Task	MAE $\downarrow$	.25	.13 (Bao et al., 2024)
Generation	Enthymeme Reconstruction (Stahl et al., 2023)	Instances	R-L $\uparrow$	.16	.17 (Stahl et al., 2023)
	Argument Summarization (Roush and Balaji, 2020)	Task	R-L $\uparrow$	.56	.57 (Roush and Balaji, 2020)

Table 5: Performance comparison of our ArgInstruct LLM with the state-of-the-art (SOTA) upper bound on six CA seed tasks: three tasks were included in our training data (instances unseen), and three were entirely new to our LLM (task unseen). In contrast, the SOTA models are trained in a supervised manner on the single task.

Model	$F_1 \uparrow$	MASE $\downarrow$	R-L $\uparrow$	Rank $\downarrow$
Majority	.38	**1.2**	.18	6.33
Random	.34	1.4	.17	6.33
Alpaca-7B-it	.44	2.3	.18	5.67
Gemma-2-9B-it	.62	3.0	.22	5.33
LLaMA-3-8B-it	.48	2.5	.22	6.00
Ministral-8B-it	.50	2.6	.24	5.00
Mistral-7B-it	.61	2.1	.26	3.33
GPT-4o-mini	.59	1.5	**.32**	2.67
ArgInstruct (Ours)	**.65**	1.9	.31	**2.33**

Table 6: Zero-shot evaluation of our *ArgInstruct* model compared to recent instruction fine-tuned models of similar size (besides GPT-4o-mini), on our CA test tasks.

6 Conclusion

Despite their strong generalization capabilities, instruction-following LLMs struggle with tasks that require domain knowledge. We propose *ArgInstruct*, a new specialized instruction fine-tuning method to address this issue for the domain of computational argumentation (CA).

As a starting point, we have collected 105 CA tasks from the literature and crafted natural instructions for each that serve as a benchmark for LLM-based CA. Additionally, we have generated 52k CA-specific tasks, adapting the self-instruct process to bridge between generalization and CA specialization. We have then trained CA-specialized instruction-following LLMs, combining the collected and generated CA tasks with general instruction fine-tuning data. Our experiments suggest that an LLM fine-tuned on the combined data performs best on unseen CA tasks without losing its general instruction-following capabilities. While the LLM did not fully reach single-task SOTA results, it is on par in half the tasks. At the same time, it outperforms several existing instruction-following models, including the proprietary GPT-4o-mini.

We conclude that our *ArgInstruct* method denotes a substantial step towards overcoming the domain challenges of LLMs, providing a benchmark dataset and a task-agnostic model for CA. We expect that our method may be well-transferable to other specialized NLP domains, for example, to the educational domain. There, our method could involve collecting seed tasks such as essay scoring, feedback generation, text suggestion, and rewriting, but we leave this to future work.

7 Limitations

The research proposed in this paper may have limitations with respect to four aspects that we discuss in the following: (1) Model training, (2) model evaluation, (3) quality of the generated data, and (4) generalizability to other domains.

Model Training We only use a subset of training instances for the training of our models. While this is a strength of our method in that we only require ≈ 500 instances per task, using all training instances could further increase the performance of models, and a different sampling of instances may lead to slightly different results also depending on the varying quality of training data instances.

Model Evaluation We point to the common problem of evaluating generative models with automatic metrics (here, ROUGE-L). Beyond related research, we at least used different measures to give adequate insights into task-related differences. In addition, we decided on a balanced evaluation of 100 instances per task to achieve a uniform, systematic setting. This is, of course, only an approximation of evaluating the full datasets, such that some values could change in tasks that are heavily instance-dependent. While our comparison of the models mentioned in this paper is fair, we do not recommend taking these values blindly and comparing them directly to approaches from related work that are evaluated on full test sets. However, we hope that our analysis in Section 4, which uses the entire test sets, gives readers an idea about the transferability. Finally, given the generally poor performance of all LLMs on regression tasks, we do not recommend using them for such tasks.

Quality of Generated Data Although we did our best to select a representative subset of datasets, we covered only a subset of the CA datasets found during our literature search (30 out of 71). The reason for this is the high manual effort required to manually craft the instructions and load and parse the respective datasets. However, we believe that our methodology would benefit from having more of these existing datasets, as it potentially decreases the amount of data that needs to be generated and could lead to an increased quality of the data in terms of diversity and instance quality. While a manual examination of the generated data, the statistics provided in the paper, and the manual evaluation of the self-instruct method by Wang et al. (2023) suggest a good quality of the data, we ultimately do not know the quality of the generated tasks, and whether their instances correctly match their corresponding instructions.

Generalizability The success of the *ArgInstruct* model depends significantly on the availability of a diverse set of tasks to be used for instructions. Its performance may be limited in domains where task data is scarce or difficult to collect, affecting the model's generalizability. While we only instantiated *ArgInstruct* for computational argumentation, we expect and encourage future work to apply our proposed methodology to other NLP areas that require specific domain knowledge.

Acknowledgments

The presented work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the project ArgSchool, project number 453073654, and within the project OASiS, project number 455913891, which is part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999). Furthermore, this work was supported by the Federal Ministry of Education and Research (BMBF), Germany, under the AI service center KISSKI (grant number 01IS22093C).

References

Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. 2016. Internet argument corpus 2.0: An SQL schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4445–4452, Portorož, Slovenia. European Language Resources Association (ELRA).

Yamen Ajjour, Milad Alshomary, Henning Wachsmuth, and Benno Stein. 2019. Modeling frames in argumentation. 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)*, pages 2922–2932, Hong Kong, China. Association for Computational Linguistics.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery.

Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. 2016a. Cross-domain mining of argumentative text through distant supervision. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1395–1404, San Diego, California. Association for Computational Linguistics.

Khalid Al-Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016b. A news editorial corpus for mining argumentation strategies. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3433–3443, Osaka, Japan. The COLING 2016 Organizing Committee.

Tariq Alhindi and Debanjan Ghosh. 2021. "Sharks are not the threat humans are": Argument component segmentation in school student essays. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 210–222, Online. Association for Computational Linguistics.

Milad Alshomary, Wei-Fan Chen, Timon Gurcke, and Henning Wachsmuth. 2021. Belief-based generation of argumentative claims. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 224–233, Online. Association for Computational Linguistics.

Aristotle. ca. 350 B.C.E./ translated 2007. *On Rhetoric: A Theory of Civic Discourse*. Oxford University Press, Oxford, UK. Translated by George A. Kennedy.

Jianzhu Bao, Bojun Jin, Yang Sun, Yice Zhang, Yuhang He, and Ruifeng Xu. 2024. A comparison-based framework for argument quality assessment. *Electronics*, 13(20).

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261, Valencia, Spain. Association for Computational Linguistics.

Tilman Beck, Ji-Ung Lee, Christina Viehmann, Marcus Maurer, Oliver Quiring, and Iryna Gurevych. 2021. Investigating label suggestions for opinion mining in German covid-19 social media. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–13, Online. Association for Computational Linguistics.

Filip Boltužić and Jan Šnajder. 2014. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58, Baltimore, Maryland. Association for Computational Linguistics.

Alexander Bukharin and Tuo Zhao. 2023. Data diversity matters for robust instruction tuning. *CoRR*, abs/2311.14736.

Jérémie Cabessa, Hugo Hernault, and Umer Mushtaq. 2025. Argument mining with fine-tuned large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6624–6635, Abu Dhabi, UAE. Association for Computational Linguistics.

Cayque Monteiro Castro Nascimento and André Silva Pimentel. 2023. Do large language models understand chemistry? a conversation with chatgpt. *Journal of Chemical Information and Modeling*, 63(6):1649–1655.

Guizhen Chen, Liying Cheng, Anh Tuan Luu, and Lidong Bing. 2024a. Exploring the potential of large language models in computational argumentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2309–2330, Bangkok, Thailand. Association for Computational Linguistics.

Zaiqian Chen, Daniel Verdi do Amarante, Jenna Donaldson, Yohan Jo, and Joonsuk Park. 2022. Argument mining for review helpfulness prediction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8914–8922, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024b. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2024. InstructEval: Towards holistic evaluation of instruction-tuned large language models. In *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models (SCALE-LLM 2024)*, pages 35–64, St. Julian's, Malta. Association for Computational Linguistics.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25(1).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. 2024. Length-controlled alpacaeval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*.

Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. 2015. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2242, Lisbon, Portugal. Association for Computational Linguistics.

Lilach Eden, Yoav Kantor, Matan Orbach, Yoav Katz, Noam Slonim, and Roy Bar-Haim. 2023. Welcome to the real world: Efficient, incremental and scalable key point analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 483–491, Singapore. Association for Computational Linguistics.

Liat Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, Yonatan Bilu, Ranit Aharonov, and Noam Slonim. 2020. Corpus wide argument mining - A working solution. In

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, February 7-12, 2020, pages 7683–7691. AAAI Press.

Mohamed Elaraby, Diane Litman, Xiang Lorraine Li, and Ahmed Magooda. 2024. Persuasiveness of generated free-text rationales in subjective decisions: A case study on pairwise argument ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14311–14329, Miami, Florida, USA. Association for Computational Linguistics.

Marc Feger and Stefan Dietze. 2024. TACO – Twitter arguments from COnversations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15522–15529, Torino, Italia. ELRA and ICCL.

Roni Friedman, Lena Dankin, Yufang Hou, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2021. Overview of the 2021 key point analysis shared task. In *Proceedings of the 8th Workshop on Argument Mining*, pages 154–164, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Martin Gleize, Eyal Shnarch, Leshem Choshen, Lena Dankin, Guy Moshkowich, Ranit Aharonov, and Noam Slonim. 2019. Are you convinced? choosing the more convincing evidence with a Siamese network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 967–976, Florence, Italy. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen,

Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lind-

say, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2020. A large-scale dataset for argument quality ranking: Construction and analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7805–7813.

Giulia Grundler, Piera Santin, Andrea Galassi, Federico Galli, Francesco Godano, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. 2022. Detecting arguments in CJEU decisions on fiscal state aid. In *Proceedings of the 9th Workshop on Argument Mining*, pages 143–157, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Argumentation mining on the web from information seeking perspective. In *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*, TUD-CS-2014-0886. CEUR-WS.

Ivan Habernal and Iryna Gurevych. 2016a. What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223, Austin, Texas. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2016b. 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 (Volume 1: Long Papers)*, pages 1589–1599, Berlin, Germany. Association for Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018a. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940, New Orleans, Louisiana. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018b. Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 386–396, New Orleans, Louisiana. Association for Computational Linguistics.

Shohreh Haddadan, Elena Cabrio, and Serena Villata. 2019. Yes, we can! mining arguments in 50 years of US presidential campaign debates. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4684–4690, Florence, Italy. Association for Computational Linguistics.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 751–762, Doha, Qatar. Association for Computational Linguistics.

Annette Hautli-Janisz, Zlata Kikteva, Wassiliki Siskou, Kamila Gorska, Ray Becker, and Chris Reed. 2022. QT30: A corpus of argument and conflict in broadcast debate. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3291–3300, Marseille, France. European Language Resources Association.

Philipp Heinisch, Anette Frank, Juri Opitz, Moritz Plenz, and Philipp Cimiano. 2022. Overview of the 2022 validity and novelty prediction shared task. In *Proceedings of the 9th Workshop on Argument Mining*, pages 84–94, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. 2017. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Rob J. Hyndman and George Athanasopoulos. 2021. *Forecasting: Principles and Practice*, 3rd edition. OTexts, Melbourne, Australia.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yohan Jo, Seojin Bang, Emaad Manzoor, Eduard Hovy, and Chris Reed. 2020. Detecting attackable sentences in arguments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–23, Online. Association for Computational Linguistics.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. 2022. Revise and resubmit: An intertextual model of text-based collaboration in peer review. *Computational Linguistics*, 48(4):949–986.

Anne Lauscher, Goran Glavaš, and Simone Paolo Ponzetto. 2018. An argument-annotated corpus of scientific publications. In *Proceedings of the 5th Workshop on Argument Mining*, pages 40–46, Brussels, Belgium. Association for Computational Linguistics.

Anne Lauscher, Henning Wachsmuth, Iryna Gurevych, and Goran Glavaš. 2022. Scientia potentia Est—On the role of knowledge in computational argumentation. *Transactions of the Association for Computational Linguistics*, 10:1392–1422.

Augustin Lecler, Loïc Duron, and Philippe Soyer. 2023. Revolutionizing radiology with gpt-based models: Current applications, future possibilities and limitations of chatgpt. *Diagnostic and Interventional Imaging*, 104(6):269–274.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Matthias Liebeck, Katharina Esau, and Stefan Conrad. 2016. What to do with an airport? mining arguments in the German online participation project tempelhofer feld. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 144–153, Berlin, Germany. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based Argument Mining for Healthcare Applications. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, Santiago de Compostela / Online, Spain.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Nathan Ong, Diane Litman, and Alexandra Brusilovsky. 2014. Ontology-based argument mining and automatic essay scoring. In *Proceedings of the First Workshop on Argumentation Mining*, pages 24–28, Baltimore, Maryland. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing*

Systems, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Joonsuk Park and Claire Cardie. 2014. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland. Association for Computational Linguistics.

Joonsuk Park and Claire Cardie. 2018. A corpus of eRulemaking user comments for measuring evaluability of arguments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon*, volume 2, pages 801–815.

Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239, Cambridge, MA. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2013. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269, Sofia, Bulgaria. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543, Baltimore, Maryland. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2015. 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 (Volume 1: Long Papers)*, pages 543–552, Beijing, China. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2016. Modeling stance in student essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2174–2184, Berlin, Germany. Association for Computational Linguistics.

Prakash Poudyal, Jaromir Savelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. ECHR: Legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.

Paula Rescala, Manoel Horta Ribeiro, Tiancheng Hu, and Robert West. 2024. Can language models recognize convincing arguments? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8826–8837, Miami, Florida, USA. Association for Computational Linguistics.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show me your evidence - an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Lisbon, Portugal. Association for Computational Linguistics.

Julia Romberg. 2022. Is your perspective also my perspective? enriching prediction with subjectivity. In *Proceedings of the 9th Workshop on Argument Mining*, pages 115–125, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Allen Roush and Arvind Balaji. 2020. DebateSum: A large-scale argument mining and summarization dataset. In *Proceedings of the 7th Workshop on Argument Mining*, pages 1–7, Online. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *CoRR*, abs/2110.08207.

Nils-Jonathan Schaller, Andrea Horbach, Lars Ingver Höft, Yuning Ding, Jan Luca Bahr, Jennifer Meyer, and Thorben Jansen. 2024. DARIUS: A comprehensive learner corpus for argument mining in German-language essays. In *Proceedings of the 2024 Joint*

International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 4356–4367, Torino, Italia. ELRA and ICCL.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. 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*, pages 380–396, Online. Association for Computational Linguistics.

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605, Melbourne, Australia. Association for Computational Linguistics.

Eyal Shnarch, Leshem Choshen, Guy Moshkowich, Ranit Aharonov, and Noam Slonim. 2020. Unsupervised expressive rules provide explainability and assist human experts grasping new domains. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2678–2697, Online. Association for Computational Linguistics.

Maria Skeppstedt, Andreas Peldszus, and Manfred Stede. 2018. More or less controlled elicitation of argumentative text: Enlarging a microtext corpus via crowdsourcing. In *Proceedings of the 5th Workshop on Argument Mining*, pages 155–163, Brussels, Belgium. Association for Computational Linguistics.

Gabriella Skitalinskaya, Jonas Klaff, and Henning Wachsmuth. 2021. 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*, pages 1718–1729, Online. Association for Computational Linguistics.

Parinaz Sobhani, Diana Inkpen, and Stan Matwin. 2015. From argumentation mining to stance classification. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 67–77, Denver, CO. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2017a. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Christian Stab and Iryna Gurevych. 2017b. Recognizing insufficiently supported arguments in argumentative essays. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 980–990, Valencia, Spain. Association for Computational Linguistics.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium. Association for Computational Linguistics.

Maja Stahl, Nick Düsterhus, Mei-Hua Chen, and Henning Wachsmuth. 2023. Mind the gap: Automated corpus creation for enthymeme detection and reconstruction in learner arguments. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4703–4717, Singapore. Association for Computational Linguistics.

Maja Stahl, Nadine Michel, Sebastian Kilsbach, Julian Schmidtke, Sara Rezat, and Henning Wachsmuth. 2024. A school student essay corpus for analyzing interactions of argumentative structure and quality. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2661–2674, Mexico City, Mexico. Association for Computational Linguistics.

Manfred Stede and Jodi Schneider. 2018. *Argumentation Mining*. Number 40 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

Benno Stein, Yamen Ajjour, Roxanne El Baff, Khalid Al-Khatib, Philipp Cimiano, and Henning Wachsmuth. 2021. Same side stance classification. In *Same Side Shared Task 2019: Same Side Stance Classification Shared Task 2019*, pages 1–7. CEUR Workshop Proceedings (CEUR-WS. org).

Shahbaz Syed, Khalid Al Khatib, Milad Alshomary, Henning Wachsmuth, and Martin Potthast. 2021. Generating informative conclusions for argumentative texts. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3482–3493, Online. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Assaf Toledo, Shai Gretz, Edo Cohen-Karlik, Roni Friedman, Elad Venezian, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. 2019. 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)*, pages 5625–5635, Hong Kong, China. Association for Computational Linguistics.

Orith Toledo-Ronen, Matan Orbach, Yonatan Bilu, Artem Spector, and Noam Slonim. 2020. Multilingual argument mining: Datasets and analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 303–317, Online. Association for Computational Linguistics.

Dietrich Trautmann. 2020. Aspect-based argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 41–52, Online. Association for Computational Linguistics.

Dietrich Trautmann, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. Fine-grained argument unit recognition and classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9048–9056.

Jannis Vamvas and Rico Sennrich. 2020. X-stance: A multilingual multi-target dataset for stance detection. *CoRR*, arXiv:2003.08385.

Jacky Visser, John Lawrence, Jean Wagemans, and Chris Reed. 2019. An annotated corpus of argument schemes in us election debates. In *Proceedings of the 9th Conference of the International Society for the Study of Argumentation (ISSA), 3-6 July 2018*, pages 1101–1111.

Henning Wachsmuth, Gabriella Lapesa, Elena Cabrio, Anne Lauscher, Joonsuk Park, Eva Maria Vecchi, Serena Villata, and Timon Ziegenbein. 2024. Argument quality assessment in the age of instruction-following large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1519–1538, Torino, Italia. ELRA and ICCL.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017. Computational argumentation quality assessment in natural language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 176–187, Valencia, Spain. Association for Computational Linguistics.

Henning Wachsmuth, Manfred Stede, Roxanne El Baff, Khalid Al-Khatib, Maria Skeppstedt, and Benno Stein. 2018a. Argumentation synthesis following rhetorical strategies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3753–3765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018b. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

Andreas Waldis, Yufang Hou, and Iryna Gurevych. 2024. How to handle different types of out-of-distribution scenarios in computational argumentation? a comprehensive and fine-grained field study. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14878–14898, Bangkok, Thailand. Association for Computational Linguistics.

Marilyn Walker, Jean Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 812–817, Istanbul, Turkey. European Language Resources Association (ELRA).

Thiemo Wambsganss, Christina Niklaus, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2020. A corpus for argumentative writing support in German. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 856–869, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. 2024a. A survey on data selection for llm instruction tuning. *Preprint*, arXiv:2402.05123.

Shaokang Wang, Fuhui Sun, Xiaoyan Wang, and Li Pan. 2024b. Incorporating target-aware knowledge into prompt-tuning for few-shot stance detection. *Information Processing & Management*, 61(5):103815.

Yizhong Wang, Yeganeh Kordi, Alisa Mishra, Swaroop and/ Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Brandon T Willard and Rémi Louf. 2023. Efficient guided generation for llms.

Fangkai Yang, Pu Zhao, Zezhong Wang, Lu Wang, Bo Qiao, Jue Zhang, Mohit Garg, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2023. Empower large language model to perform better on industrial domain-specific question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 294–312, Singapore. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Timon Ziegenbein, Shahbaz Syed, Felix Lange, Martin Potthast, and Henning Wachsmuth. 2023. Modeling appropriate language in argumentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4344–4363, Toronto, Canada. Association for Computational Linguistics.

A Collected CA Datasets

Table 7 shows the complete list of the 71 CA datasets considered.

B Exemplary Seed Instructions

This section provides examples of manually crafted seed instructions for different tasks and datasets. While listing all 105 instructions here would be impractical, the complete set of seed instructions can be found in the provided code.

Argument Mining

- **Argument Detection & Classification** (Stab et al., 2018): "Given a sentence and a topic, classify the sentence as a "supporting argument" or "opposing argument" if it includes a relevant reason for supporting or opposing the topic, or as a "non-argument" if it does not include a reason or is not relevant to the topic."
- **Argument Component Classification** (Stab and Gurevych, 2017a): "Given the following essay as context, and a list of argumentative components extracted from the essay. Label each argumentative component as "major claim", "claim", or "premise"."

Argument Assessment

- **Inappropriateness Detection** (Ziegenbein et al., 2023): "An argument is appropriate if the used language supports the creation of credibility and emotions as well as if it is proportional to its topic. Given the following argument and the topic of the debate the argument appeared in. Decide whether the argument is Appropriate or Inappropriate."
- **Overall Quality Rating** (Wachsmuth et al., 2017): "How would you rate the overall quality of the author's argumentation on the scale "1" (Low), "2" (Average) or "3" (High)?"

Argument Generation

- **Enthymeme Reconstruction** (Stahl et al., 2023): "An enthymeme is defined here as any missing argumentative discourse unit (ADU) that would complete the logic of a written argument. Is there a problematic enthymematic gap at the position marked with "<mask>" in the following argument?"
- **Argument Summarization** (Roush and Balaji, 2020): "Create a word-level extractive summary of the argument by "underlining" and/or "highlighting" the evidence in such a way to support the argument being made."

C Exemplary Generated Instructions

Examples from the 10% generated instructions with the highest maximal similarity to the seed instructions are:

I_5: "Consider the following arguments (argument a and argument b). Would you agree with the following statement? Argument a has worse reasoning because it presents facts without explaining their relevance to the claim."

I_6: "Given a question, stance (yes vs. no) towards this question and a premise, your task is to form a counterargument against the given stance using the given premise."

I_7: "Given two arguments, determine whether they have the same stance towards their common topic."

Further exemplary instructions from the 10% generated instructions with the lowest maximal similarity to the seed instructions are:

Argument Mining	Argument Assessment	Argument Generation
Al-Khatib et al. (2016b)	Abbott et al. (2016)♡	Alshomary et al. (2021)♡
Al-Khatib et al. (2016a)	Ajjour et al. (2019)	Eden et al. (2023)
Alhindi and Ghosh (2021)	Beck et al. (2021)	Hasan and Ng (2014)♡
Bar-Haim et al. (2017)	Friedman et al. (2021)♡	Jo et al. (2020)
Boltužić and Šnajder (2014)♡	Gleize et al. (2019)	Roush and Balaji (2020)♡
Chen et al. (2022)♡	Gretz et al. (2020)♡	Schiller et al. (2021)♡
Eckle-Kohler et al. (2015)	Habernal et al. (2018a)♡	Skeppstedt et al. (2018)♡
Ein-Dor et al. (2020)	Habernal et al. (2018b)	Skitalinskaya et al. (2021)♡
Feger and Dietze (2024)	Habernal and Gurevych (2016b)	Stahl et al. (2023)♡
Grundler et al. (2022)	Habernal and Gurevych (2016a)♡	Syed et al. (2021)♡
Habernal and Gurevych (2017)♡	Heinisch et al. (2022)♡	Wachsmuth et al. (2018a)♡
Haddadan et al. (2019)	Persing et al. (2010)	Wachsmuth et al. (2018b)♡
Hautli-Janisz et al. (2022)♡	Persing and Ng (2013)	
Hidey et al. (2017)	Persing and Ng (2014)	
Kuznetsov et al. (2022)♡	Persing and Ng (2015)♡	
Lauscher et al. (2018)	Persing and Ng (2016)	
Liebeck et al. (2016)	Sobhani et al. (2015)	
Mayer et al. (2020)	Stab and Gurevych (2017b)	
Ong et al. (2014)	Stein et al. (2021)♡	
Park and Cardie (2018)	Toledo et al. (2019)	
Peldszus and Stede (2015)♡	Vamvas and Sennrich (2020)	
Poudyal et al. (2020)♡	Wachsmuth et al. (2017)♡	
Reimers et al. (2019)♡	Walker et al. (2012)	
Rinott et al. (2015)	Ziegenbein et al. (2023)♡	
Schaller et al. (2024)		
Shnarch et al. (2018)		
Shnarch et al. (2020)		
Stab et al. (2018)♡		
Stab and Gurevych (2017a)♡		
Stahl et al. (2024)		
Toledo-Ronen et al. (2020)		
Trautmann (2020)		
Trautmann et al. (2020)		
Visser et al. (2019)		
Wambsganss et al. (2020)		

Table 7: The list of all 71 considered CA datasets. The 30 datasets selected as seed datasets for generating CA tasks are marked with "♡".

I_8: "Classify the type of logical fallacy committed in the argument (ad hominem, appeal to emotion, appeal to ignorance, appeal to popularity, appeal to tradition, circular reasoning, confirmation bias, false dichotomy, genetic fallacy, post hoc ergo propter hoc, red herring, slippery slope, straw man, tu quoque)."

I_9: "This task requires you to identify whether each statement below expresses a subjective opinion or objective fact. please read all instructions carefully before starting! subjective opinions are personal judgments about things that cannot be proven true or false objectively. they reflect individual preferences and values rather than empirical observations or established knowledge. objective facts are verifiable pieces of information derived from observation, experimentation, measurement, calculation, etc... they describe physical reality independent of human perception or interpretation. examples include historical events, scientific laws, mathematical formulas, etc."

I_{10}: "Given a hypothetical scenario wherein there exists conflict over ownership rights pertaining to certain property located within disputed territory between two neighboring countries; construct logical sequence of steps leading up towards resolution through negotiation process involving both parties concerned alongside third party mediator appointed mutually agreed upon basis taking into account all relevant factors such as historical background surrounding issue at hand."

D Hyperparameter-Tuning

We use adapter-based low-rank adaptation (LoRA) (Hu et al., 2021) with $r = 16$, an amplification factor of 32, and a dropout rate of 0.05 to enhance training efficiency. To determine an optimal learning rate, number of epochs, and batch size, we use Optuna (Akiba et al., 2019) for hyperparameter optimization. Ultimately, all models are trained for 7 epochs with a learning rate of 9.88×10^{-5}, an effective batch size of 64, cosine learning rate decay, and a warmup ratio of 0.05.

E ArgInstruct: Task Results

Table 8 and 9 show the performance of our *ArgInstruct* model for the 100 sampled test instances for all 105 CA seed tasks.

F ArgInstruct for CA Tasks: Dual General Instruction-Finetuning

Table 10 presents the CA performance of the base models and LLMs trained with *ArgInstruct* and its ablations. For completeness, we also include variants using the already instruction-finetuned *Gemma-2-9B-General* as the base model, along with additional general instruction finetuning (+*general*) integrated into CA-specific finetuning. However, this does not lead to further improvements in CA performance.

Source	Task	Split	$F_1 \uparrow$	MASE $\downarrow$	R-L $\uparrow$
Abbott et al. (2016)	Predict Agreement	test	-	2.28	-
	Predict Respect	test	-	2.09	-
	Predict Factuality	test	-	1.94	-
	Predict Nice	test	-	2.16	-
	Predict Sarcasm	test	-	3.15	-
Alshomary et al. (2021)	Stance Prediction	train	0.42	-	-
	Belief Based Claim Generation	train	-	-	0.13
Boltužić and Šnajder (2014)	Stance Detection	train	0.39	-	-
Chen et al. (2022)	Review Helpfulness Prediction	train	-	1.02	-
	Relation Detection	train	0.63	-	-
	Unit Segmentation Prediction	train	-	-	0.95
	Init Classification Prediction	train	-	-	0.69
Friedman et al. (2021)	Key Point Matching	train	0.76	-	-
	Key Point Generation	train	-	-	0.31
Gretz et al. (2020)	Quality Assessment	test	-	1.54	-
	Stance Prediction	test	0.92	-	-
Habernal et al. (2018a)	Argument Reasoning Comprehension	train	-	-	0.89
Habernal and Gurevych (2016a)	Classify More Convincing Argument	train	0.83	-	-
	Classify More Details Argument	train	0.53	-	-
	Classify More Balanced Argument	train	0.62	-	-
	Classify More Credible Argument	train	0.56	-	-
	Classify More Clear Argument	train	0.53	-	-
	Classify More On-Topic Argument	train	0.46	-	-
	Classify More Provoking Argument	train	0.57	-	-
	Classify More Smart Argument	train	0.50	-	-
	Classify Less Attacking Argument	train	0.75	-	-
	Classify Less Language-issues Argument	train	0.66	-	-
	Classify Less Unclear Argument	train	0.55	-	-
	Classify Less Facts Argument	train	0.45	-	-
	Classify Less Reasoning Argument	train	0.53	-	-
	Classify Less Relevant-reasons	train	0.52	-	-
	Classify Not An Argument	train	0.70	-	-
	Classify Nonsense Argument	train	0.55	-	-
	Classify Off-topic Argument	train	0.79	-	-
	Classify Generally weak Argument	train	0.51	-	-
Habernal and Gurevych (2017)	Detect Persuasive Documents	train	0.64	-	-
	Extract Toulmin Components	train	-	-	0.52
Hasan and Ng (2014)	Reason Identification	train	-	-	0.56
Hautli-Janisz et al. (2022)	Propositional Relations Identification	test	0.45	-	-
	Illocutionary Relations Identification	test	0.23	-	-
Heinisch et al. (2022)	Novelty Classification	train	0.53	-	-
	Validity Classification	train	0.75	-	-
	Relative Novelty Classification	train	0.39	-	-
	Relative Validity Classification	train	0.43	-	-
Kuznetsov et al. (2022)	Pragmatic Tagging	test	-	-	0.24
Peldszus and Stede (2015)	Argumentative Role Determination	train	0.79	-	-
	Function of Segment Determination	train	0.36	-	-
	Unit Attachment Identification	train	0.63	-	-
	Argumentative Text Creation	train	-	-	0.21
	Central Claim Extraction	train	-	-	0.79
Persing and Ng (2015)	Classifying Argument Strength	train	-	2.05	-

Table 8: Performance of our *ArgInstruct* model on all 105 CA seed tasks. The split indicates whether the task was seen during training (train) or is a completely unseen task (test). The performance is always measured on the 100 sampled instances from the test split of the respective task data. (Part 1/2)

Source	Task	Split	$F_1 \uparrow$	MASE ↓	R-L ↑
Poudyal et al. (2020)	Argument Clause Recognition	train	0.60	-	-
	Clause Relation Prediction	train	0.58	-	-
	Premise Recognition	train	0.52	-	-
	Conclusion Recognition	train	0.66	-	-
Reimers et al. (2019)	Argument Similarity	train	0.51	-	-
Schiller et al. (2021)	Aspect Controlled Argument Generation	test	-	-	0.11
Skitalinskaya et al. (2021)	Claim Revision Improvement	train	0.51	-	-
	Suboptimal Claim Detection	train	0.50	-	-
	Claim Improvement Suggestions	train	0.37	-	-
	Claim Optimization	train	-	-	0.72
Stab et al. (2018)	Argument Identification	train	0.70	-	-
Stab and Gurevych (2017a)	Identifying Argumentative Relations	test	0.67	-	-
	Stance Recognition	test	0.70	-	-
	Identifying Argument Components	test	-	-	0.67
	Classifying Argument Components	test	-	-	0.67
Stahl et al. (2023)	Detect Enthymemes	train	0.50	-	-
	Reconstruct Enthymemes	train	-	-	0.17
Stein et al. (2021)	Same Side Stance Classification	train	0.54	-	-
Syed et al. (2021)	Conclusion Generation	test	-	-	0.21
Wachsmuth et al. (2017)	Rate Local Acceptability	train	-	0.95	-
	Rate Local Relevance	train	-	0.88	-
	Rate Local Sufficiency	train	-	0.94	-
	Rate Cogency	train	-	1.00	-
	Rate Credibility	train	-	0.55	-
	Rate Emotional Appeal	train	-	0.83	-
	Rate Clarity	train	-	0.99	-
	Rate Appropriateness	train	-	0.63	-
	Rate Arrangement	train	-	1.21	-
	Rate Effectiveness	train	-	1.05	-
	Rate Global Acceptability	train	-	1.11	-
	Rate Global Relevance	train	-	0.87	-
	Rate Global Sufficiency	train	-	1.02	-
	Rate Reasonableness	train	-	0.96	-
	Rate Overall Quality	train	-	0.88	-
Wachsmuth et al. (2018a)	Synthesize Argument	train	-	-	0.20
Wachsmuth et al. (2018b)	Same Debate Opposing Counters	test	-	-	0.46
	Same Debate Counters	test	-	-	0.24
	Same Debate Opposing Argument	test	-	-	0.30
	Same Aebate Argument	test	-	-	0.19
Ziegenbein et al. (2023)	Inappropriateness Detection	train	0.72	-	-
	Toxic Emotions Detection	train	0.70	-	-
	Missing Commitment Detection	train	0.65	-	-
	Missing Intelligibility Detection	train	0.70	-	-
	Other Inappropriateness Detection	train	0.80	-	-
	Excessive Intensity Detection	train	0.78	-	-
	Emotional Deception Detection	train	0.78	-	-
	Missing Seriousness Detection	train	0.62	-	-
	Missing Openness Detection	train	0.64	-	-
	Unclear Meaning Detection	train	0.71	-	-
	Missing Relevance Detection	train	0.76	-	-
	Confusing Reasoning Detection	train	0.72	-	-
	Detrimental Orthography Detection	train	0.83	-	-
	Reason Unclassified Detection	train	0.62	-	-

Table 9: Performance of our *ArgInstruct* model on all 105 CA seed tasks. The split indicates whether the task was seen during training (train) or is a completely unseen task (test). The performance is always measured on the 100 sampled instances from the test split of the respective task data. (Part 2/2)

Approach	Fine-Tuning Data: seedCA	genCA	general	(a) Unseen CA Instances: F$_1$↑	MASE↓	R-L ↑	Rank↓	(b) Unseen CA Tasks: F$_1$↑	MASE↓	R-L↑	Rank↓
Gemma-2-9B	○	○	○	.45	1.6	.39	10.0	.45	4.2	.15	13.7
+ seedCA	●	○	○	**.65**	1.1	**.50**	**2.3**	**.65**	2.5	.23	7.7
+ genCA	○	●	○	.51	2.7	.45	9.7	.52	3.0	.30	6.7
+ general	○	○	●	.50	2.1	.32	11.7	.51	2.6	.29	8.0
+ seedCA, genCA	◑	◑	○	.61	1.6	.49	4.3	.57	2.9	.26	7.0
+ genCA, general	○	◑	◑	.51	1.7	.39	9.7	.50	3.0	.17	10.7
+ seedCA, general	◑	○	◑	.59	**1.0**	.48	4.3	.60	2.3	.30	4.3
+ seedCA, genCA, general	◔	◔	◔	.61	1.5	.49	4.3	**.65**	2.5	**.32**	2.7
Gemma-2-9B-General	○	○	◎	.48	2.3	.34	12.3	.50	2.1	.24	9.3
+ seedCA	●	○	◎	.64	1.2	.49	2.7	.63	2.1	**.32**	5.0
+ genCA	○	●	◎	.49	2.6	.44	10.7	.52	2.6	.30	6.7
+ general	○	○	●	.51	2.1	.35	11.3	.50	2.3	.27	8.7
+ seedCA, genCA (ArgInstruct)	◑	◑	◎	.57	1.3	.49	4.7	**.65**	**1.9**	.31	**1.7**
+ genCA, general	○	◑	◑	.48	1.8	.43	10.0	.50	2.9	.31	9.3
+ seedCA, general	◑	○	◑	.59	1.5	.49	4.7	.63	2.3	.31	5.3
+ seedCA, genCA, general	◔	◔	◔	.55	1.3	.48	6.0	.57	2.2	.29	5.7

Table 10: Full version of Table 3 containing all dataset combinations for the *Gemma-2-9B-General* approach. Performance on unseen CA instances and unseen CA tasks: Evaluation of Gemma-2-9B trained to follow instructions on 52k instances of CA seed tasks (+*seedCA*), generated CA tasks (+*genCA*), general tasks (+*general*) and combinations of these. The symbols represent the proportion of fine-tuning instances coming from each source (○: 0%, ◔: 33%, ◑: 50%, ●: 100%). ◎ indicates that the data was used beforehand to perform general instruction fine-tuning. Performance is evaluated on (a) unseen CA instances from the training tasks and (b) unseen CA test tasks. The best values are **bold**, the best per base model underlined.