On Integrating LLMs Into an Argument Annotation Workflow

Robin Schaefer
Applied Computational Linguistics / University of Potsdam / Germany
Bundesdruckerei GmbH / Berlin / Germany
firstnamelastname@mailbox.org

Abstract

Given the recent success of LLMs across different NLP tasks, their usability for data annotation has become a promising area of research. In this work, we investigate to what extent LLMs can be used as annotators for argument components and their semantic types in German tweets through a series of experiments combining different models and prompt configurations. Each prompt is constructed from modular components, such as class definitions or contextual information. Our results suggest that LLMs can indeed perform argument annotation, particularly of semantic argument types, if provided with precise class definitions. However, a fine-tuned BERT baseline remains a strong contender, often matching or exceeding LLM performance. These findings highlight the importance of considering not only model performance, but also ecological and financial costs when defining an annotation workflow.

1 Introduction

Over the last decade, Argument Mining (AM) has developed into a versatile research area. While early work focused on basic tasks such as claim (Daxenberger et al., 2017), evidence (Rinott et al., 2015) and relation detection (Carstens and Toni, 2015), more recent research focused on the analysis of argument quality (Wachsmuth et al., 2024) and strategies (Schaefer et al., 2023). Text domain can be identified as another dimension of variance. Early AM research was usually applied to rather formal texts, e.g. persuasive essays (Stab and Gurevych, 2017). Following this early trend, the focus somewhat shifted to include user-generated text domains, e.g. ChangeMyView (Al Khatib et al., 2020) or Twitter (Schaefer and Stede, 2021).

While these different subareas and -tasks of AM include their own challenges, they usually have in common a need for reliably annotated data, which is reflected in a substantial amount of work focused at least in part on annotation. With data scarcity being a common bottleneck in NLP tasks, recent research has focused on the question to what extent large language models (LLMs) can be leveraged for data annotation. Although not explicitly designed for classification, LLMs, being autoregressive models, can be prompted to function as annotators in classification settings. Since research has shown that modern LLMs perform well in zero-shot scenarios (Kojima et al., 2022), they are less dependent on annotated corpora compared to encoder-only models like BERT (Devlin et al., 2019), which require task-specific fine-tuning.

In this study, we investigate to what extent LLMs can be utilized as annotators for argumentation. In particular, we focus on Argument Component Type Classification (ACTC) both in a coarse-grained, i.e. claim and evidence, and in a fine-grained sense, i.e. semantic argument types. We use the GerCCT corpus as a starting point, our German tweet dataset, which has previously been expert annotated for argument components and their semantic types (Schaefer and Stede, 2022). We developed an extensive list of *experimental settings* consisting of an LLM and a prompt. We applied three popular open-weight models of different sizes, namely Llama-3.2-3B, Mixtral-8x7B, and Llama-3.3-70B. Each prompt was constructed from a number of modular components, e.g., class definitions or contextual information. We conducted experiments to identify the ideal combination of LLM and prompt to solve the annotation task and conclude this paper with a discussion of our results as well as the necessary aspects to consider when integrating LLMs into an argument annotation workflow.

This paper is structured as follows. In Section 2 we present the related work, before describing the corpus and the original approach in Section 3. In Section 4 we focus on our experiments, methods, and results. We discuss our findings in Section 5, before concluding the paper in Section 6.

Proceedings of the 12th Argument Mining Workshop, pages 87–99

2 Related Work

Our work mainly falls into three areas of study: 1) AM on Twitter, 2) applying LLMs in AM scenarios, and 3) using LLMs as annotators.

AM on Twitter. AM on Twitter has been investigated in a number of studies, usually with a focus on creating datasets. Bosc et al. (2016) annotated 4,000 tweets for argumentativeness as well as for relations between tweets. Addawood and Bashir (2016) annotated 3,000 tweets with a set of evidence types, e.g. news or expert opinion. An SVM approach trained on a mixed feature set performed best in classification experiments. Bhatti et al. (2021) annotated a large tweet corpus with different premise classes with respect to a claim hashtag. Best classification results were obtained using a fine-tuned BERT model. Wührl and Klinger (2021) annotated 1,200 tweets in the biomedical domain for explicit and implicit claims, as well as conducted classification experiments. More recently, Feger and Dietze (2024) applied a pre-classification fine-tuning approach to BERTweet (Nguyen et al., 2020) for the classification of reasoning and factual content in full Twitter conversations. They used contrastive loss and text augmentation in a Siamese network, which yielded high results.

LLMs for AM. Given the recent prominence of LLMs across various NLP tasks, work has been conducted in the field of AM as well. Al Zubaer et al. (2023) used GPT-3.5-Turbo and GPT-4 and few-shot prompting for conclusion and premise detection in a legal context. They found that both models could not compete with a BERT model and argued that this might be due to the LLMs not being domain-specifically fine-tuned and their sensitivity to prompt phrasing. Abkenar et al. (2024) tested the suitability of different Mistral and Llama variants for argument component and relation classification by applying them to previously published AM corpora. They reported that LLMs yielded better results for relation classification. They further found that providing additional context had a mixed effect on model performance. Cabessa et al. (2025) fine-tuned several (quantized) open-weight LLMs and applied them to several AM datasets. They reported state-of-the-art results across different tasks, including argument component and relation classification. Similarly, Gorur et al. (2025) showed that a set of open-weight and proprietary LLMs applied to eleven datasets performed well for relation classification in a few-shot scenario. Mixtral-8x7B yielded best results with Llama2-70B ranging second. Altemeyer et al. (2025) applied GPT-4o to different frameworks of argument summarization and reported good results. They further evaluated the output for *coverage* and *redundancy* using, among other approaches, GPT-4o-mini, which yielded high correlation with human judgments. Favero et al. (2025) investigated the applicability of (fine-tuned) small LLMs to the tasks of argument segmentation, classification and quality assessment in student essays. They showed that fine-tuning improved results for segmentation and classification compared to a few-shot approach without fine-tuning, but worsened results for quality assessment. Also working on essays, Stahl et al. (2024) explored the usability of different zero-shot and few-shot prompts for essay scoring and feedback generation via LLMs. While generated feedback proved to be helpful, it did not appear to have a strong effect on scoring. Wachsmuth et al. (2024) discussed the potential of LLMs for assessing argument quality and proposed to feed models with instructions inspired by argumentation theory during fine-tuning.

LLMs for data annotation. Similar to their application to AM tasks, LLMs also have been used in data annotation scenarios. Early work by Gilardi et al. (2023) showed that ChatGPT exceeded the performance of crowdworkers in tweets and news data across different tasks, e.g., topic annotation. Pavlovic and Poesio (2024) used an LLM to generate opinion distributions for different corpora and found that these distributions notably diverged from human annotations. Bibal et al. (2025) used GPT-4o in an iterative workflow to both annotate a named entity dataset and refine the annotation guidelines based on these annotations, yielding improved inter-annotator agreement compared to the original guidelines. Mirzakhmedova et al. (2024) applied LLMs to the task of argument quality annotation and reported that PaLM 2 produced labels that were moderately consistent with human annotations, compared to GPT-3.5-Turbo which showed a more divergent outcome. Gligorić et al. (2025) used LLM annotations and generated confidence scores to guide human annotation. Both LLM and human annotations were combined to calculate statistical estimates of different quantities of interest. Bavaresco et al. (2024) evaluated the annotation results of eleven LLMs on 20 NLP datasets and found

that models exhibited notable variance with respect to their performance, thus suggesting the need for careful validation of the models' capabilities.

While a certain overlap exists to previous studies, our work differs by 1) applying LLMs to an annotation task in tweets, 2) focusing on German data as opposed to the primary usage of English data in the literature, and 3) conducting extensive experimentation using a set of prompts constructed from a number of relevant modular components, e.g., class definitions and context.

3 Corpus and Original Approach

Starting point for our work is our previously published GerCCT corpus (Schaefer and Stede, 2022). The corpus is an annotated subset of a larger German tweet dataset with a focus on climate change discourse and consists of 1,200 tweet pairs in a *reply to* relationship. While the reply tweet has been annotated, the so-called source tweet has been used as additional context during annotation.

The corpus contains expert annotations on the full tweet level of semantic argument types, called *argument properties* in our original paper, which each fall into the category of either claim or evidence. Claim types are *unverifiable claim* and *verifiable claim*. Evidence types are *reason* and *external evidence*.[1] This is a translated example from the original paper: *"You cannot negotiate with nature. This is why you cannot prepare a climate protection package like a trade agreement. It's about science and its laws are non-negotiable. [...]"*, which has been annotated as containing the types *unverifiable claim*, *verifiable claim*, and *reason*.

We further used the argument type annotations to derive argument component annotations, i.e. claim and evidence, as well as the general +/- argumentative class, thus resulting in three layers of argument annotation consisting of seven classes in total. We use all layers in this work. See Table 1 for an overview of argument classes and their annotation proportions.

In addition to argument annotation, the corpus has also been labeled for toxic language as well as sarcasm. Importantly, in the original approach argumentative and toxic language are considered to be mutually exclusive, that is, a toxic tweet cannot contain argumentation. Given this rule, we consider the detection of toxic language as an important factor. However, in this work, we do not pay attention to sarcasm detection.

Layer	Class	Proportion
1st	Argument	.70
2nd	Claim	.65
2nd	Evidence	.25
3rd	Unverifiable Claim	.59
3rd	Verifiable Claim	.20
3rd	Reason	.11
3rd	External Evidence	.14

Table 1: Argument classes and their proportions as annotated by Schaefer and Stede (2022). Each value represents the proportion of tweets that have been annotated with the respective class, i.e. the proportions do not add up to one.

In the original study, we used the annotated corpus to train models for ACTC. We applied different approaches with a fine-tuned BERT (*bert-base-german-cased*)[2] model yielding best results for argument classes. In this study, we use the majority baseline and the BERT results as baselines.

4 Experiments

In this study, we investigate to what extent LLMs can be used as annotators for ACTC tasks, i.e. for the annotation of argument components and their semantic types. We approach this question via different *experimental settings*. Each setting is defined by an LLM and a prompt. Each prompt is constructed of various modular components, including, for example, the addition of class definitions or context. Our complete list of experimental settings is shown in Table 3.

In the following, we present the models and prompts we used in our experiments. We continue with a description of our inference runs we conducted by applying Mixtral-8x7B to every combination of prompt and class. We then used the macro F1 scores we obtained from these runs to identify the best performing prompts, which we finally used with Llama-3.2-3B and Llama-3.3-70B.[3] We performed permutation tests for statistical significance

[1] Note that the original annotations also include the argument type *internal evidence*. However, given that we did not include it in our previous classification experiments due to it being rarely annotated, we do not use it in this study either.

[2] https://huggingface.co/google-bert/bert-base-german-cased

[3] Given the substantial carbon emissions of LLMs (Wu et al., 2025) we decided to run the full set of experiments only with Mixtral-8x7B.

Model	Vendor	Release
Llama-3.2-3B	Meta AI	Sept 25, 2024
Mixtral-8x7B	Mistral AI	Dec 11, 2023
Llama-3.3-70B	Meta AI	Dec 6, 2024

Table 2: Large language models (ordered by size). The *B* in the model name refers to the number of parameters in billion.

testing and conclude this section with a description of the results.

Models. We made use of three LLMs of different sizes: Llama-3.2-3B, Mixtral-8x7B[4], and Llama-3.3-70B (see Table 2). By adding model size as a variable, we could conduct more fine-grained analyses with respect to the effect of parameter count. All models are open-weight and multilingual including German. We used Groq[5] for inference, which quantizes model weights to 8 bits, while still running calculations in 16 bits. We set the model temperature to 0. For simplicity, we refer to these models as *Llama-3B*, *Mixtral*, and *Llama-70B*.

Prompts. Each prompt can be described as a combination of components (or their absence) that are selected to enable the LLM to perform the task (see Appendix A for an example). In the following, we will describe each component in detail. Every setting is defined as *zero-shot*, i.e. we decided to not add annotation examples to the prompt to support the LLM. Also, in every setting we provide the respective reply tweet and the name of the class at hand and prompt the LLM to binarily annotate a tweet with the label 1 or 0, e.g., +/- claim. Beginning with this general structure, we continue to build a prompt as follows. First, we may insert additional helpful information in the form of context or class definitions. *Context* refers to the source tweet in a tweet pair, i.e. to the tweet that was not annotated but was used as additional context by the expert annotators in Schaefer and Stede (2022). By adding a source tweet we try to simulate the conditions under which the original annotation took place. We add the class definition by providing a translated version of the definitions given in the annotation scheme of the original paper (see Appendix B for the class definitions). While we tried to stay as close to the annotation scheme as possible, we had to perform minor adjustments in order to facilitate the task for the LLM.

In addition to inserting further information to the prompt, we may task the LLM to ignore tweets containing toxic language when performing the annotation. Recall that in our previous study we decided not to annotate argumentation in toxic tweets. As this decision may be somewhat unintuitive, an LLM could benefit from being explicitly prompted to pay attention to toxic language. Finally, as Röttger et al. (2024) showed, outcomes of LLMs may be affected by an open vs forced-choice setting. While we enforce the model to binarily label a tweet, in some prompts we ask it to justify its decision, thereby giving it space to argue its case.

In total, we designed 14 prompts with different characteristics. We used a small subset of the corpus (n: 50) and Mixtral to identify challenges in prompt phrasing, as well as in transferring the annotation scheme into a form that can be leveraged by an LLM. We eventually arrived at a number of *building blocks*, i.e. succinct instructions and placeholders, e.g. the definition of a specific class, which we combined into prompts, depending on the requirements of the respective setting.

Inference. Having constructed our full set of prompts, we proceeded with running inference. We prompted Mixtral to individually label each tweet of the corpus with every class, according to each experimental setting we defined, resulting in $1,200 \times 7 \times 14$ inference calls. We postprocessed the generated output with regular expressions to extract the class label. We applied a simple heuristic of extracting the first integer from the output string. If the first integer was not 0 or 1 we labeled the negative class, i.e. 0. Afterward, we calculated macro F1 by using the original annotations as gold standard.

To identify the best performing prompts, we ranked them according to their performance as reflected by their macro F1 scores. We selected the best performing prompt for the claim classes and the evidence classes, respectively, and utilized them to label the corpus with Llama-3B and Llama-70B, resulting in four additional experimental settings. We again performed postprocessing and evaluation as described. Our F1 scores are shown in Table 3.

[4]Mixtral-8x7B is a Mixture-of-Experts model. Rather than representing a single 56B parameter model, it consists of eight distinct 7B expert models, of which only a subset is activated during inference. The selection of active expert models is governed a *gating network* and depends on the respective input prompt.

[5]https://groq.com/

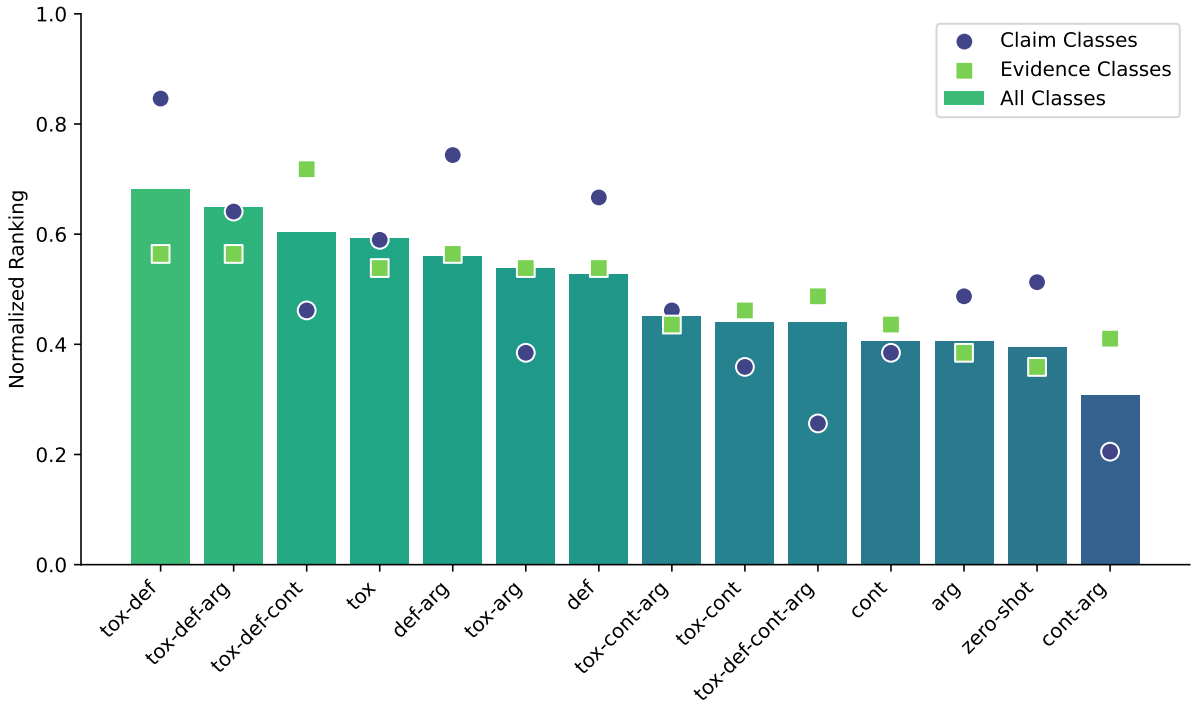

Figure 1: Normalized ranking of prompts for all classes (bars), claim classes (dots), and evidence classes (squares).

Prompt Ranking. In order to identify the most promising prompts to use for the annotation task, we ranked them as follows. Given a set of prompts P and a set of classes C, where $|P| = 14$ and $|C| = 7$, we assign to each prompt a ranking score R_p, which we calculate as:

$$R_p = \sum_{c \in C} r_{p,c}$$

where $r_{p,c}$ is the rank of prompt p for class c. Each rank is assigned with respect to the macro F1 of prompt p for class c, in descending order. We applied min-max normalization to rescale the ranking scores to the range $[0, 1]$. The minimum ranking score is defined as $R_{min} = |C|$ and the maximum ranking score is defined as $R_{max} = |C| \times |P|$. In addition to ranking scores for the entire set of classes, we also calculated scores for claim and evidence subsets, respectively, where $|C| = 3$, e.g., *claim*, *unverifiable claim*, and *verifiable claim*.

The prompt ranking is shown in Figure 1. When analyzing the ranking for the full set of classes, depicted as bars in the figure, we found the following pattern. While both the identification of toxic language and the addition of class definitions tended to have a benefiting effect, simplistic prompts that only contain context or ask the model to argue its decision could not compete. However, just prompting the model to consider toxic language when making a decision resulted in a decent ranking position.

Turning to the analyzes for the claim and evidence class subsets, we found that claim classes showed a substantially higher variance than evidence classes (SD: 0.18 vs 0.09). We further found that *tox-def* performed best for claim classes, while for evidence classes *tox-def-cont* yielded the best ranking position. We thus consider these prompts as the most promising for argument component and type annotation via LLM.[6]

Permutation Testing. We calculated statistical significance by running permutation tests on the comparison of the BERT baseline and the best performing experimental setting per class. To this end, we simulated the output of the BERT model by iteratively flipping labels starting from the gold standard until the desired macro F1 score was obtained, e.g., 0.73 for the claim class. To achieve reliable results, we simulated the output of the BERT model one hundred times per class.

We then ran two-sided permutation tests using each of the one hundred simulations and the output of the best performing experimental setting per

[6]Note that *tox-def-arg* ranked second for the full class set. However, we only utilized the prompts with Llama-3B and Llama-70B that ranked highest for the claim and evidence sets, respectively.

Setting	Model	Argument	Claim	Evidence	UC	VC	Reason	EE
baseline	majority	.41	.40	.43	.37	.44	.47	.46
baseline	BERT	**.70**	**.73**	**.77**	.70	.69	.60	.86
tox-def	Llama-3B	.24	.29	.44	.37	.47	.53	.48
tox-def-cont	Llama-3B	.23	.26	.44	.30	.44	.48	.49
zero-shot	Mixtral	.53	.61	.51	.55	.70	.61	.52
arg	Mixtral	.55	.60	.52	.58	.70	.60	.52
cont	Mixtral	.59	.59	.56	.59	.61	.53	.53
def	Mixtral	.50	.61	.51	.59	.70	.59	.69
tox	Mixtral	.65	.61	.55	.61	.57	.56	.55
cont-arg	Mixtral	.58	.56	.55	.57	.59	.55	.53
def-arg	Mixtral	.48	.62	.54	.59	.71	.58	.68
tox-arg	Mixtral	.66	.59	.56	.62	.54	.55	.54
tox-cont	Mixtral	.64	.61	.58	.39	.57	.48	.54
tox-def	Mixtral	.63	.63	.54	.62	.62	.53	.71
tox-cont-arg	Mixtral	.63	.61	.57	.61	.55	.50	.54
tox-def-arg	Mixtral	.66	.59	.56	.64	.63	.50	.71
tox-def-cont	Mixtral	.65	.60	.58	.61	.59	.51	.70
tox-def-cont-arg	Mixtral	.65	.57	.57	.61	.57	.48	.68
tox-def	Llama-70B	.53	.71	.66	**.72**	.68	**.64**	**.90**
tox-def-cont	Llama-70B	.66	.72	.63	.69	**.72**	.61	.88

Table 3: Macro F1 scores by experimental setting and class. The baseline results are taken from Schaefer and Stede (2022) (UC: unverifiable claim; VC: verifiable claim; EE: external evidence).

class. We conducted 10,000 permutations per test and used the difference in macro F1 as the test statistic. The null hypothesis (H_0) assumes that both BERT and LLM outputs are sampled from the same distribution, i.e. observed differences are due to chance. We report the mean of p-values and the percentage of p-values < 0.05 (see Table 4).

Results. We report macro F1 scores (see Table 3) for comparison with the majority and BERT baselines taken from our previous study. To begin with, we found that in most experimental settings the majority baseline was surpassed. Only the smallest model Llama-3B appeared to be unable to sufficiently solve the task with *tox-def-cont* performing worse than *tox-def*.

Mixtral, on the other hand, showed mixed results with respect to the class at hand. While, for the general argument class, it could compete with Llama-70B *tox-def-cont* in some settings, F1 scores ranging from 0.63 to 0.66, and even outperformed Llama-70B *tox-def* in most settings, the claim and evidence component classes appeared to be more challenging. There Mixtral showed a substantial distance to Llama-70B, especially for the claim class. Turning to the semantic type classes, we found that Mixtral yielded mediocre results for *unverifiable claim* with most settings ranging between 0.57 and 0.62. For *verifiable claim*, however, we found that Mixtral mildly exceeded the BERT baseline using the following comparatively simplistic prompts: *def-arg*, *def*, *arg*, or *zero-shot*. Neither of these scores, however, were statistically significant. With respect to *reason*, some Mixtral settings were able to compete with the BERT baseline, while for *external evidence* Mixtral performed substantially worse than both Llama-70B and BERT.

Llama-70B yielded the best F1 scores of all LLMs. This was achieved primarily by using the *tox-def* prompt. However, *tox-def-cont* notably outperformed *tox-def* for the general argument class (0.66 vs 0.53) and also showed better results for *verifiable claim* (0.72 vs 0.68). With respect to the BERT baseline, Llama-70B surpassed it in all semantic type classes, while BERT yielded better results for the argument component and general argument classes.

We conclude Section 4 with our permutation test results (see Table 4), where we report mean p-values per class as well as the percentage of p-values < 0.05. We found that $p < 0.05$ for both

Class	BERT	LLM	P-Value	
Argument	.70	.66	.035*	100%
Claim	.73	.72	.571	0%
Evidence	.77	.66	≈ 0***	100%
UC	.70	.72	.392	0%
VC	.69	.72	.145	0%
Reason	.60	.64	.048*	67%
EE	.86	.90	.035*	100%

Table 4: Permutation test results: mean of p-value and percentage of p-values < 0.05 (* p < 0.05, *** p < 0.001). For convenience, we show the best LLM results as well as the BERT baseline.

reason and *external evidence*, thus indicating a statistically significant difference in model performance. From the F1 scores we can conclude that this difference is driven by Llama-70B outperforming BERT. However, we failed to reject the null hypothesis for *unverifiable claim* and *verifiable claim*. Considering the argument component classes, we found evidence for a significant effect for the evidence class ($p < 0.001$), while we again failed to reject H_0 for the claim class. The argument class, on the other hand, also yielded $p < 0.05$. Thus, for *argument* and *evidence* we can conclude that BERT significantly surpassed Llama-70B given the respective F1 scores. With respect to the percentages of p-values < 0.05, we found a rather binary pattern. Statistically significant classes showed a percentage of 100% of p-values < 0.05 with the exception of *reason* (67%), thereby indicating a less reliable effect for this class. In those cases where we failed to reject the null hypothesis on average, we did not find any cases of p-values < 0.05.

5 Discussion

LLMs do not necessarily outperform BERT. While we provided evidence for LLMs being able to solve the task of argument annotation in specific experimental settings, we did not find that they outperformed the BERT baseline per se. Furthermore, we observed for the semantic types *unverifiable* and *verifiable claim* that advantages of using an LLM instead of BERT might be actually due to chance, since we failed to reject H_0. We also found that the BERT approach significantly outperformed the best LLM setting for the general argument class as well as the evidence class. Our results are thus in line with mixed results previously reported in the literature (Mirzakhmedova et al., 2024; Stahl et al., 2024).

Importantly, we do not consider a statistically significant effect as a prerequisite to employ an LLM to the annotation task, given that utilizing BERT also failed to significantly exceed the performance of the best LLM setting for most classes. Thus, we interpret the performance of both approaches to be similar enough to warrant their implementation. However, we suggest that the absent dominance of the LLM approach is a strong argument in favor of keeping *the human in the loop*. Since efficiently prompting an LLM is not a trivial task, we argue that precise annotation guidelines, developed by (human) experts as well as thoroughly validated by using annotator agreement metrics, e.g., Krippendorff's α, are necessary to ensure reliability and confidence in the annotations. Provided with these guidelines, an LLM may be capable of performing the remainder of the annotations.

Providing definitions is essential. Our results indicate that providing definitions of classes has a beneficial effect, as shown by the better ranking of prompts that include definitions. This is especially the case for the best performing prompts *tox-def* and *tox-def-cont*. We argue that providing definitions may be especially necessary for argumentation, as argument categories, e.g. claim, tend to have a common meaning which differs from their more formal definition in the context of AM. Further our results suggest that class definitions need to be precise. While using Llama-70B led to best results for all argument properties, i.e. for those classes with rather concise definitions, it performed worse for argument components and the general argument class. We argue that this may be due to their definitions being more complex since they are essentially combinations of the simpler semantic type definitions.

With respect to the other prompt components, we find our assumption confirmed by the prompt ranking that toxic language detection indeed has a positive effect on the results. This is intuitive given the definition of argumentation in the annotation guidelines. In contrast, our prompt ranking further hints that providing additional context does not benefit the results. Although one of the best performing prompts does include context, i.e. *tox-def-cont*, we argue that its good performance mainly results from the combination of toxic language detection and class definitions, given that the prompt *tox-def*

appeared to yield better results for most classes. We suggest that this might be due to the context having a deviating effect on the model, as it needs to process another piece of text, which does not need to be labeled. Finally, we fail to find an effect of prompting an LLM to justify its decision. However, we still consider this to be potentially helpful, as it enables the researcher to interpret the model output.

Model size matters. One main outcome of our experiments is the apparent importance of model size. While the medium-sized Mixtral model yielded good results in some experimental settings, even the majority baseline proved to be a challenge for Llama-3B. Best results were consistently achieved by Llama-70B, which, however, does not imply that it outperforms the BERT baseline, as we have previously discussed.

These findings are in line with previous research indicating that a larger number of parameters enables LLMs to more efficiently capture both semantic nuances which are prevalent in a subjective task such as argumentation as well as complexity of contextual information. It further suggests that better results can be obtained by applying models of size $> 70B$ to the task. However, this is a mere hypothesis and requires rigorous testing given that argument annotation remains a challenging task.

It also raises the question to what extent the completion of a task justifies the added resources associated with employing increasingly large models. An alternative approach would to be to improve the performance of smaller models, so-called *small language models*. While our results show that Llama-3B is not suitable to solve the task, more research in this direction may result in higher model performance, while keeping energy consumption at a lower rate.

Resources should be considered. Data annotation tends to be expensive in terms of financial and ecological resources. The classic approach is to train a group of expert annotators to solve the task by following a set of clearly defined annotation guidelines. As these guidelines need to be validated, several ratings per data point are required, thus potentially rendering the annotation of a corpus a costly endeavor.

On the other hand, training and running an LLM causes a substantial amount of carbon emissions, which requires ethical considerations. Wu et al. (2025) investigated the effect of model size, quantization, and hardware on carbon emissions. They found that smaller models tend to outperform larger ones with respect to carbon emissions with increasing request rates, while larger models benefit the most from quantization. In addition, older hardware tends to contain less embodied carbon than newer hardware. Both types of resource need to be considered in combination with model performance, in order to decide on the ideal approach for the task.

6 Conclusion

In this study, we investigated to what extent LLMs can be integrated into an argument annotation workflow, with a special focus on ACTC on tweets both on the level of argument components and semantic types. To this end, we defined experimental settings consisting of model and prompt and used Mixtral to identify the most promising prompts for the annotation task, i.e. *tox-def* and *tox-def-cont*, before utilizing them with Llama-3B and Llama-70B. In order to run permutation tests for significance testing between the BERT baseline and the best performing experimental settings, we simulated the output of the BERT model.

While we found the annotation task to be challenging for an LLM, we identified specific combinations of prompt and LLM that produced good results, especially for the classification of semantic types. However, we also found that a BERT model fine-tuned on human expert annotations was a strong contender, rendering the choice of the best approach a non-trivial one. We argued in favor of precise guidelines, ideally created and validated by human experts, as well as clear class definitions to facilitate the annotation task. Given the guidelines, an LLM could undertake the main part of the annotation. However, we also suggested considering the required resources, both financial and ecological, alongside model performance when deciding on the best approach to employ.

For future research, we are interested in testing the applicability of advanced prompting techniques such as few-shot and chain-of-thought prompting. In addition, approaching the annotation task in an open setting in combination with another LLM to make the final judgment, similar to the approach carried out in Röttger et al. (2024), could be a fruitful direction to follow. We also aim to extend our approach to capture more complex argumentation structures. Finally, we consider exploring label

variation of LLMs to be a promising next research direction, both with respect to the annotation itself as well as the usability of label variation for classification (Plank, 2022).

Limitations

In our experiments, we made use of a single corpus. Extending the number of corpora both within the task of ACTC and across different AM tasks could give a clearer picture with respect to the usability of LLMs in an argument annotation workflow. Similarly, applying a larger number of LLMs may result in a more comprehensive understanding of the capabilities of these models.

This limitation also extends to the investigation of a single language, i.e. German. While being common in areas of NLP research that are not explicitly multilingual in nature, this raises the question to what extent the results generalize cross-linguistically.

So far, we have produced single LLM annotations and compared them to expert annotations of the corpus, thus creating a scenario of two annotators. The study may benefit from the generation of multiple LLM outputs that simulate the work of multiple human annotators.

Previous research has shown that LLMs tend to be sensitive to exact prompt phrasing (Röttger et al., 2024). While we defined different prompt settings, we did not create prompt variants within a setting. This could lead to more robust results.

Ethical Considerations

Previous research has shown that LLMs produce biases (Gallegos et al., 2024). While this is also true for other models and human annotators, this may result in skewed annotations, especially in argumentative texts which tend to deal with controversial topics. Applying LLMs to annotation tasks in less-resourced languages like German may increase these biases, as well as eventually using the annotated data for fine-tuning purposes, potentially resulting in a self-reinforcing feedback loop.

Furthermore, while automating data annotation via LLMs may be feasible, it also may result in the replacement of paid labor for human annotators, thereby having socioeconomic implications that should be considered when designing an annotation study.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback.

References

Mohammad Yeghaneh Abkenar, Weixing Wang, Hendrik Graupner, and Manfred Stede. 2024. Assessing open-source large language models on argumentation mining subtasks. *Preprint*, arXiv:2411.05639.

Aseel Addawood and Masooda Bashir. 2016. "what is your evidence?" a study of controversial topics on social media. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Khalid Al Khatib, Michael Völske, Shahbaz Syed, Nikolay Kolyada, and Benno Stein. 2020. Exploiting personal characteristics of debaters for predicting persuasiveness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7067–7072, Online. Association for Computational Linguistics.

Abdullah Al Zubaer, Michael Granitzer, and Jelena Mitrović. 2023. Performance analysis of large language models in the domain of legal argument mining. *Frontiers in Artificial Intelligence*, 6.

Moritz Altemeyer, Steffen Eger, Johannes Daxenberger, Tim Altendorf, Philipp Cimiano, and Benjamin Schiller. 2025. Argument summarization and its evaluation in the era of large language models. *Preprint*, arXiv:2503.00847.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2024. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *Preprint*, arXiv:2406.18403.

Muhammad Mahad Afzal Bhatti, Ahsan Suheer Ahmad, and Joonsuk Park. 2021. Argument mining on Twitter: A case study on the planned parenthood debate. In *Proceedings of the 8th Workshop on Argument Mining*, pages 1–11, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adrien Bibal, Nathaniel Gerlek, Goran Muric, Elizabeth Boschee, Steven C. Fincke, Mike Ross, and Steven N. Minton. 2025. Automating annotation guideline improvements using LLMs: A case study. In *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*, pages 129–144, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Tom Bosc, Elena Cabrio, and Serena Villata. 2016. DART: a dataset of arguments and their relations on Twitter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1258–1263, Portorož, Slovenia. European Language Resources Association (ELRA).

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.

Lucas Carstens and Francesca Toni. 2015. Towards relation based argumentation mining. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 29–34, Denver, CO. Association for Computational Linguistics.

Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. 2017. What is the essence of a claim? cross-domain claim identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2066, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Lucile Favero, Juan Antonio Pérez-Ortiz, Tanja Käser, and Nuria Oliver. 2025. Leveraging small llms for argument mining in education: Argument component identification, classification, and assessment. *Preprint*, arXiv:2502.14389.

Marc Feger and Stefan Dietze. 2024. BERTweet's TACO fiesta: Contrasting flavors on the path of inference and information-driven argument mining on Twitter. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2256–2266, Mexico City, Mexico. Association for Computational Linguistics.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Kristina Gligorić, Tijana Zrnic, Cinoo Lee, Emmanuel J. Candès, and Dan Jurafsky. 2025. Can unconfident llm annotations be used for confident conclusions? *Preprint*, arXiv:2408.15204.

Deniz Gorur, Antonio Rago, and Francesca Toni. 2025. Can large language models perform relation-based argument mining? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8518–8534, Abu Dhabi, UAE. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Nailia Mirzakhmedova, Marcel Gohsen, Chia Hao Chang, and Benno Stein. 2024. Are large language models reliable argument quality annotators? In *Robust Argumentation Machines*, pages 129–146, Cham. Springer Nature Switzerland.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Maja Pavlovic and Massimo Poesio. 2024. The effectiveness of LLMs as annotators: A comparative overview and empirical analysis of direct representation. In *Proceedings of the 3rd Workshop on Perspectivist Approaches to NLP (NLPerspectives) @ LREC-COLING 2024*, pages 100–110, Torino, Italia. ELRA and ICCL.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. 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.

Paul Röttger, Valentin Hofmann, Valentina Pyatkin, Musashi Hinck, Hannah Kirk, Hinrich Schuetze, and Dirk Hovy. 2024. Political compass or spinning arrow? towards more meaningful evaluations for values and opinions in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15295–15311, Bangkok, Thailand. Association for Computational Linguistics.

Robin Schaefer, René Knaebel, and Manfred Stede. 2023. Towards fine-grained argumentation strategy

analysis in persuasive essays. In *Proceedings of the 10th Workshop on Argument Mining*, pages 76–88, Singapore. Association for Computational Linguistics.

Robin Schaefer and Manfred Stede. 2021. Argument mining on Twitter: A survey. *it - Information Technology*, 63(1):45–58.

Robin Schaefer and Manfred Stede. 2022. GerCCT: An annotated corpus for mining arguments in German tweets on climate change. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6121–6130, Marseille, France. European Language Resources Association.

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

Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. Exploring LLM prompting strategies for joint essay scoring and feedback generation. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. Association for Computational Linguistics.

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.

Yanran Wu, Inez Hua, and Yi Ding. 2025. Unveiling environmental impacts of large language model serving: A functional unit view. *Preprint*, arXiv:2502.11256.

Amelie Wührl and Roman Klinger. 2021. Claim detection in biomedical Twitter posts. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 131–142, Online. Association for Computational Linguistics.

A Prompt Example

German version:
Du bist ein erfahrener Assistent für die Analyse von Argumentation im Text. Du bekommst einen Text, eine Argumentationskategorie, eine Definition der Argumentationskategorie und einen Kontext. Deine Aufgabe ist es zu entscheiden, ob der Text die Kategorie enthält oder nicht. Antworte mit 1, wenn die Kategorie vorhanden ist. Antworte mit 0, wenn die Kategorie nicht vorhanden ist. Wenn der Text toxische Sprache wie zum Beispiel Beleidigungen enthält, antworte auch mit 0. Berücksichtige den Kontext bei deiner Entscheidung. Nutze die Definition der Kategorie, um eine Entscheidung zu fällen. Begründe deine Antwort.

Kontext: ```{context}```

Argumentationskategorie: {category}

Definition der Argumentationskategorie: ```{definition}```

Text: {text}

English translation:
You are an experienced assistant for analyzing argumentation in text. You will be given a text, an argumentation category, a definition of the argumentation category, and a context. Your task is to decide whether the text contains the specified category or not. Answer with 1 if the category is present. Answer with 0 if the category is not present. If the text contains toxic language, such as insults, also answer with 0. Take the context into account when making your decision. Use the definition of the category to guide your judgment. Argue for your answer.

Context: ```{context}```

Argumentation category: {category}

Definition of the argumentation category: ```{definition}```

Text: {text}

Figure 2: The prompt for setting *tox-def-cont-arg*. It includes 1) the identification of toxic language, makes use of 2) class definitions and 3) context, as well as 4) asks the model to argue for its answer.

B Class Definitions

Class	Definition
Argument	An argument contains at least one claim or piece of evidence. A claim includes unverifiable claims and verifiable claims. An unverifiable claim is a subjective standpoint or positioning. A verifiable claim can potentially be verified by another source, such as scientific papers or statistics. Evidence is proof of an unverifiable claim or a verifiable claim. Types of evidence are external evidence or reason. External evidence includes, for example, news, expert opinions and quotations. External evidence is often provided via links. Evidence can also be reason, which means that it justifies an unverifiable claim or a verifiable claim.
Claim	A claim includes unverifiable claims and verifiable claims. An unverifiable claim is a subjective standpoint or positioning. A verifiable claim can potentially be verified via an external source, such as scientific references or statistics.
Evidence	Evidence is proof of an unverifiable claim or a verifiable claim. Types of evidence are external evidence or reason. External evidence includes, for example, news, expert opinions and quotations. External evidence is often provided via links. Evidence can also be reason, which means that it justifies an unverifiable claim or a verifiable claim.
UC	An unverifiable claim is a subjective standpoint, positioning, interpretation or prognosis. Although such a statement is unverifiable, it can still be sufficiently supported by providing reasons.
VC	A statement is considered a verifiable claim, if it can potentially be verified via an external source. However, it is not sufficient for a statement to be identified as verifiable by linguistic means alone. Potential sources for verifiable claims include, for example, scientific references, statistics, political manifestos and lexicon entries. Verifiable claims do not have to be factually correct.
Reason	Reason is a statement that justifies an unverifiable claim or a verifiable claim. The unverifiable claim or verifiable claim must also be present in the text. An unverifiable claim is a subjective standpoint or positioning. A verifiable claim can potentially be verified by an external source, such as scientific references or statistics. The connection between reason and an unverifiable claim or verifiable claim is often causal.
EE	External evidence is a source of proof for an unverifiable claim or a verifiable claim. An unverifiable claim is a subjective standpoint or positioning. A verifiable claim can potentially be verified by an external source, such as scientific references or statistics. External evidence does not have to be factually correct. External evidence must be explicitly present in the text. It includes, for example news, expert opinions, blog entries, books, petitions, images and quotations. External evidence is often provided via links, which is why links are considered external evidence.

Table 5: The class definitions. Each definition has been translated from the respective German version we used for prompting. (UC: unverifiable claim; VC: verifiable claim; EE: external evidence).