LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
CENTRUM FÜR INFORMATIONS- UND SPRACHVERARBEITUNG
STUDIENGANG COMPUTERLINGUISTIK

Masterarbeit

im Studiengang Computerlinguistik

an der Ludwig- Maximilians- Universität München

Fakultät für Sprach- und Literaturwissenschaften

Beyond Noise: Detecting Annotation Error from Human Label Variation using LLMs

vorgelegt von
Longfei Zuo

Betreuer:	Dr. Siyao (Logan) Peng
Prüfer:	Prof. Dr. Barbara Plank
Bearbeitungszeitraum:	11. März - 28. Juli 2025

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, den 28. Juli 2025

.....
Longfei Zuo

Erklärung der verwendeten KI-Tools

Ich versichere, dass ich diese Arbeit eigenständig, ohne jede externe Unterstützung, außer den unten aufgeführten Ressourcen, angefertigt habe.

Purpose	**Section(s)**	**Tool**
Grammar check	All document	ChatGPT 4o, Grammarly
Translations	Abstract	ChatGPT 4o
Literature search	Related work	Perplexity AI
Code review suggestions	Github repository	ChatGPT 4o, Claude Sonnet 4

München, den 28. Juli 2025

. . . . . .
Longfei Zuo

Abstract

High-quality labeled datasets are essential for Natural Language Processing (NLP) research, but ensuring data quality remains a major challenge. Human Label Variation (HLV) is prevalent in tasks such as Natural Language Inference (NLI), where multiple labels may be valid for the same instance. This inherent ambiguity makes it even more difficult to distinguish annotation errors from plausible variations . In this thesis, we propose a framework that leverages large language models (LLMs) to detect annotation errors through explanation-based validation. Specifically, the LLM first generates diverse, label-specific explanations for each instance and then validate them by assigning a validity score to each explanation. If none of the explanations under a given label are validated, the label is considered erroneous.

We perform a comprehensive analysis comparing human and LLM explanations across distribution, validation results, and impact on model learning. Our experiments show that LLM-generated explanations align well with human annotations in terms of label distributiong and removing LLM-detected errors from the training data leads to improved performance on downstream tasks. These demonstrate that LLMs can detect annotation errors and offer complementary insights to human annotators, highlighting the potential of explanation-based pipelines to scale validation with minimal human effort, offering a practical approach to improving dataset quality in the presence of label variation.

Hochwertig annotierte Datensätze sind wichtig für die Forschung im Bereich Natural Language Processing (NLP), doch die gute Datenqualität bleibt eine große Herausforderung. Besonders im Bereich Natural Language Inference (NLI) ist die Human Label Variation (HLV) weit verbreitet, dabei können für dieselbe Instanz mehrere Labels gleichermaßen gültig sein. Diese inhärente Mehrdeutigkeit erschwert es zusätzlich, echte Annotationsfehler von plausiblen Abweichungen zu unterscheiden. In dieser Arbeit schlagen wir ein Framework vor, das große Sprachmodelle (LLM) zur Erkennung von Annotationsfehlern durch erklärungsbasierte Validierung einsetzt. Das LLM generiert zunächst vielfältige, label-spezifische Erklärungen zu jeder Instanz und bewertet diese anschließend durch die Vergabe von Gültigkeitsscores. Wird keine der Erklärungen zu einem Label validiert, so wird das Label als fehlerhaft eingestuft.

Wir führen eine umfassende Analyse durch, in der wir menschliche und LLM-generierte Erklärungen hinsichtlich ihrer Verteilung, Validierungsergebnisse und ihres Einflusses auf das Modelllernen vergleichen. Unsere Experimente zeigen, dass die von LLMs generierten Erklärungen hinsichtlich der Labelverteilung gut mit menschlichen Annotationen übereinstimmen und dass das Entfernen der vom LLM erkannten Fehler die Leistung bei nachgelagerten Aufgaben verbessert. Diese Ergebnisse belegen, dass LLMs zuverlässig Annotationsfehler erkennen und Menschen bei der Annotation helfen können. Diese Studie unterstreicht das Potenzial erklärungsbasierter Pipelines für eine skalierbare, effiziente und interpretierbare Validierung von Datensätzen bei gleichzeitig geringem menschlichem Aufwand.

Acknowledgement

I would like to sincerely thank my advisor, Dr. Siyao (Logan) Peng, for his consistent support and encouragement throughout my thesis and studies. I also thank Prof. Dr. Barbara Plank for her helpful advice and academic guidance during my thesis and throughout my master's program. Their insights and mentorship have helped shape my understanding of research and deepened my interest in the field.

I am also thankful to the Center for Information and Language Processing (CIS) for offering me the opportunity to transition into the field of NLP. It opened a new path for me and allowed me to find direction in an area I now feel passionate about. I would also like to thank MaiNLP, I have not only learned knowledge but also gained inspiring experience in academic research.

I am deeply grateful to my family and friends. Though some of them may be 8,000 kilometers away, their unwavering support, trust, and encouragement have always been by my side. Without their presence and understanding, this journey would have been much more difficult.

Lastly, I would like to thank myself for staying committed through challenges, maintaining curiosity, and continuing to move forward step by step. This thesis represents not only an academic achievement, but also a personal journey I'm proud of.

Contents

1 Introduction

1.1 Background and Motivation

Datasets play a fundamental role in all aspects of research, with label quality being particularly crucial. Human Label Variation (HLV, Plank 2022) refers to the plausible variation in annotation, where multiple labels can be assigned to a single instance. This variation often offers richer and more nuanced information than a single ground-truth label. HLV has gained significant attention across many NLP tasks like Natural Language Inference (NLI) (Pavlick and Kwiatkowski, 2019; Nie et al., 2020b), where multiple plausible labels can be valid for the same premise-hypothesis pair (Jiang and de Marneffe, 2022; Weber-Genzel et al., 2024; Jayaweera and Dorr, 2025).

While HLV better reflects real-world ambiguities, it also introduces a challenge: annotation errors may be obscured within the variation. Annotation errors refer to labels assigned for invalid or inconsistent reasons that do not reflect plausible interpretations of the instance. Since model training heavily depends on high-quality labeled data, annotation errors can undermine model performance and reliability. Thus, ensuring high data quality is essential for building trustworthy NLP systems.

1.2 Human-involved vs. LLM-involved Error Detection

Many approaches to detecting such errors have been introduced in previous research (Klie et al., 2023; Weber-Genzel et al., 2024; Weber et al., 2024). Among them, Weber-Genzel et al. (2024) propose a two-round annotation procedure in which human experts first assign labels and provide corresponding explanations. In the second round, after reviewing all explanations from their own and those from other annotators for the instance, annotators assess the validity of their original explanations. With a comprehensive understanding of different labeling rationales, annotators can make more informed validity judgments.

An explanation is considered *self-validated* if the annotator finds their original reasoning sound, and *peer-validated* if the majority of other annotators think the explanation makes sense for the label in Round 2. Research shows that considering both types of validation is beneficial, as each provides complementary insights. This method leverages self-correction to detect annotation errors in a realistic and intuitive way: if no explanation under a given label is self-validated, the label is deemed erroneous. Since such errors naturally arise in real-world annotation settings, this approach provides a more comprehensive and faithful evaluation for error detection.

However, a key limitation of this approach is its reliance on human experts to generate and validate explanations, making it resource-intensive and time-consuming. To mitigate this challenge, studies have shown that explanations generated by large language models (LLMs) are comparable to human explanations in approximating human judgement distribution (HJD) on NLI (e.g., in the first round) (Chen et al., 2025b).

This raises the following research questions in our study:

1. RQ1: How good are the LLM-generated explanations, do they signal valid HLVs or are they more error-prone than humans?

2. RQ2: Can LLMs detect errors using their own explanations if a second validation round is introduced?

Building on these, we propose an **LLM-based two-round annotation error detection framework** to minimize human expert involvement while maintaining both efficiency and accuracy. Specifically, we adapt the original human-in-the-loop pipeline by replacing human experts with an LLM for both explanation generation and validation.

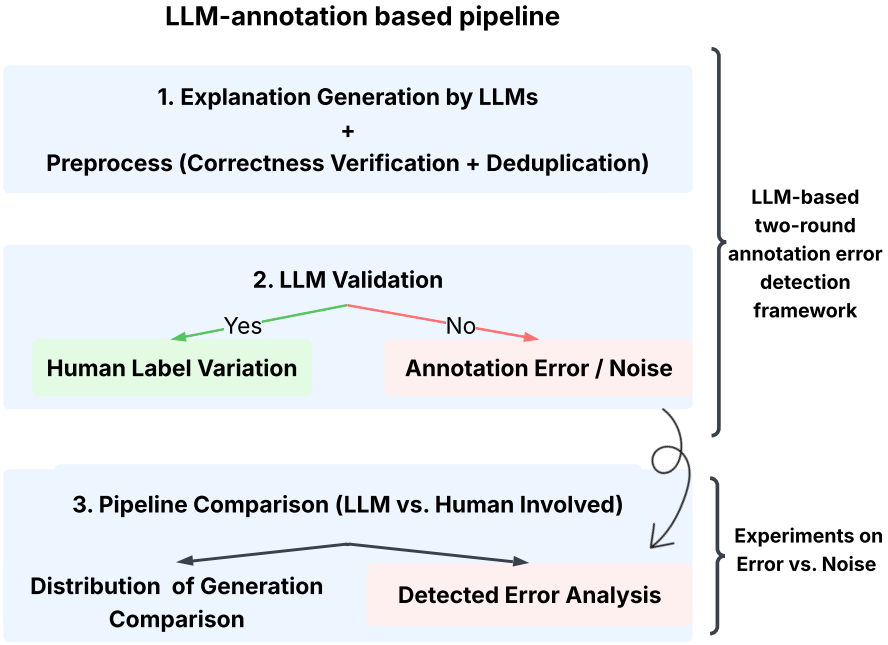


Figure 1.1: The pipeline used in this study. Both explanation generation and validation are performed by LLMs. The comparison step includes analysis of the distribution of explanations generated by humans and LLMs, as well as detected error analysis.

The working pipeline is shown in Figure 1.1. In the first round, given a premise–hypothesis pair and a candidate label from the three NLI classes: Entailment (E), Neutral (N), or Contradiction (C), the LLM is prompted to generate a corresponding explanation (§3). To ensure the quality and diversity of the generated explanations, we apply a series of preprocessing steps, which aim to filter out redundant or inconsistent responses, thereby preserving a more precise distribution of generated explanations. In the second round, the same LLM is prompted to assign a validity score to each explanation it previously generated, reflecting how well it supports the given label for the premise–hypothesis pair (§4). Explanations that receive low scores are considered "not-validated", and following previous research, a label is classified as erroneous if all of its associated explanations are not validated, closely mirroring the self-validation criterion used in the original human-based pipeline.

This process emulates the pipeline proposed by Weber-Genzel et al. (2024), requiring much less human involvement at both stages. For example, while human annotators typically provide only one or two explanations per instance for a label, LLMs can easily generate 5 to 10 diverse explanations.

1.3 Evaluation and Contributions

To evaluate the effectiveness of our framework (§5), we first compare the label explanation distributions before and after LLM validation against human annotations either directly by measuring distributional similarity, or by converting the distributions into rankings and comparing their orderings. We further assess the quality of the detected annotation errors from multiple perspectives: 1. **Ranking-based evaluation**: Explore the consistency of average validity scores with human judgments; 2. **Error distribution alignment**: Com-

pare the distribution of detected errors from LLMs and humans; 3. **Fine-tuning impact**: A key question that arises is whether annotation errors behave similar to random noise. We define random noise as labels generated through a stochastic process, without correlation with semantics or other textual factors. If errors exhibit similar effects as random noise, their informativeness would be low, suggesting that they can be systematically removed to enhance dataset quality without losing meaningful information. To assess this, we measure model performance changes after fine-tuning on cleaned datasets by removing LLM-detected errors, and compare the results to models fine-tuned on the cleaned version by removing human-detected errors.

Our results demonstrate that the label distribution after LLM validation aligns more closely with that of crowdworkers in the ChaosNLI dataset (Nie et al., 2020b), which is often considered to reflect HLV in NLI tasks. Removing errors identified by the LLM and using this cleaned dataset for fine-tuning downstream models yields improved performance. This performance gain is not even observed when removing human-detected errors. This suggests that the LLM-based pipeline provides cleaner supervision and enhances generalization for model training.

Although the annotation errors identified by our LLM-based framework show little overlap with those flagged by human annotators, LLM results provide additional perspectives. Therefore, it is still beneficial to incorporate LLM-based validation for future studies.

2 Related Work

2.1 Natural Language Inference (NLI)

Natural Language Inference (NLI) is the task of determining whether a hypothesis can be logically inferred from a premise, labeled as entailment, neutral, or contradiction (Williams et al., 2018).

NLI task has long been used for evaluating models' capabilities in reasoning and understanding semantic relationships between sentence pairs (Bowman et al., 2015; Storks et al., 2020) It is also known to exhibit a high degree of human label variation (HLV) (Pavlick and Kwiatkowski, 2019; Nie et al., 2020b; Jiang and de Marneffe, 2022; Jiang et al., 2023; Madaan et al., 2025), where multiple labels (entailment, neutral, contradiction) may appear plausible depending on the annotator's perspective, world knowledge, or interpretative bias.

There are many related NLI datasets that contain multiple plausible labels for each instance that reflect HLV:

The Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018) is one of the most widely used NLI datasets and forms the basis for many downstream evaluations. However, its single-label annotations can miss the whole human interpretations. To address this, ChaosNLI (Nie et al., 2020b) provides distributional annotations from 1000 crowdworkers over a subset of MNLI and SNLI (Bowman et al., 2015) examples, offering a more realistic and ecologically valid representation of NLI labels. Building on this, VariErr (Weber-Genzel et al., 2024) extends ChaosNLI by incorporating four annotators who provide both labels and explanations. It also introduces a two-stage human annotation process, where annotators not only label and explain their decisions but also validate both their own and others' explanations. This validation step enables the identification of annotation errors, making VariErr a valuable resource for studying not only HLV but also systematic annotation errors.

Other benchmark works in capturing HLV in NLI include WANLI (Liu et al., 2022), which uses worker and AI collaboration to set up datasets, and Adversarial NLI (ANLI) (Nie et al., 2020a), which collects data via an iterative, adversarial human-and-model-in-the-loop procedure. These datasets further highlight that not all NLI disagreements arise from random noise, but often reflect deeper ambiguities or reasoning gaps.

2.2 Human Label Variation

Human label variation (HLV) refers to plausible variations in annotation, which is the phenomenon where different annotators assign different, but plausible, labels to the same data instance (Plank, 2022). Such variation is increasingly recognized as an inherent property of language understanding tasks, offering rich insight into the ambiguity and subjectivity of human interpretation (Nie et al., 2020b; Weber-Genzel et al., 2024).

Aroyo and Welty (2015) challenges the traditional notion of a single truth in human annotation and proposes crowd truth as a more realistic framework that captures the subjective and diverse nature of human interpretation. Pavlick and Kwiatkowski (2019) show that disagreements in NLI annotations are often systematic rather than random noise. This finding questions the idea that there is only one correct label and suggests that we may need to think differently about how models should learn from data.

Other studies have investigated how to train models under HLV. Uma et al. (2022)

showed that training on soft label distributions (rather than majority-vote labels) improves generalization when high-quality multi-annotator data is available.

2.3 Annotation Error Detection (AED)

Annotation Error Detection (AED) has a long history in NLP (Larson et al., 2020; Weber et al., 2024; Weber-Genzel et al., 2024). Numerous studies have shown that even widely-used benchmarks may also contain annotation errors (Northcutt et al., 2021; Klie et al., 2023; Bernier-Colborne and Vajjala, 2024), like CoNLL 2003 for Named Entity Recognition (Wang et al., 2019; Reiss et al., 2020; Rücker and Akbik, 2023)

Klie et al. (2023) conduct a comprehensive survey of AED methods, categorizing approaches into two categories: flaggers and scorers. Flaggers make binary decisions by directly identifying whether an instance contains an annotation error, while scorers assign a probability score reflecting the likelihood of an error, allowing the top-ranked instances to be flagged for review or correction.

Annotation errors are not limited to tasks with one single correct label, they also occur in cases where multiple labels may be valid for the instance. Such ambiguous instances, often reflecting HLV, tend to be particularly challenging for machine learning models to learn from and predict (Klie et al., 2023).

Weber-Genzel et al. (2024) have proposed a two-round annotation procedure in which experts firstly annotate and provide explanations for each label and then do the validity judgement. In the second round, an annotation is classified as erroneous when all label-explanation pairs are not self-validated. This approach is intuitive and comprehensive for using self-correction as the final criterion for error detection. Since these errors arise naturally in real-world scenarios, this method offers a more realistic evaluation for error detection methods.

2.4 LLM-generated Explanations

LLM-generated explanations have been applied across a variety of tasks. Li et al. (2022) propose using LLM-generated explanations to train smaller reasoning models through a multi-task learning framework, demonstrating that such explanations can transfer reasoning abilities effectively. From a different angle, Kunz and Kuhlmann (2024) analyze the intrinsic properties of LLM-generated explanations, showing that their certian behavior patterns are influenced by the training data. Lubos et al. (2024) leverage LLMs to generate high-quality explanations aimed at helping users in different recommendation approaches.

A growing line of research also investigates whether LLMs can replicate or approximate human annotation distributions. Jiang et al. (2023) examine GPT-3's ability to generate explanations for label variation in NLI, finding that while its outputs are often fluent and coherent, the generated explanations can sometimes fail to support the given label. Furthermore, Chen et al. (2025b) explore the potential of LLMs to replace human annotators in generating explanations that approximate the human judgement distribution. These studies collectively underscore the strong potential of LLMs to generate explanations, while also revealing key areas for further improvement.

2.5 Error and Noise

In prior work, incorrect labels are often associated with noisy annotations. For instance, Mao et al. (2021) describe noisy annotations as including incorrect class labels in the context of object detection. Similarly, Rehbein and Ruppenhofer (2017) propose error detection methods for automatically annotated text, using the terms annotation error and annotation noise interchangeably.

Beigman Klebanov and Beigman (2009) define annotation "annotation noise" as the expected proportion of hard instances, which are ambiguous examples where annotators exhibit high personal bias, among the items where annotators appear to agree. This framing aligns closely with the concept of HLV, recognizing that disagreement may not stem from error, but from intrinsic ambiguity. The same authors further elaborate on this perspective in follow-up work (Beigman Klebanov and Beigman, 2010).

3 Explanation Generation and Preprocessing

We introduce a fully automated annotation error detection framework that eliminates the need for human involvement. Instead of relying on human-provided label distributions or human explanations, the pipeline prompts LLMs to generate as many distinct explanations as possible for each label and uses their distributions for later studies. In this chapter, we first introduce the models used in this study (§3.1), followed by the explanation generation process (§3.2). Two preprocessing approaches, correctness verification and explanation deduplication, are described in detail in §3.3 and §3.4, respectively. Finally, we present an ablation study examining the impact of the output token limit on the model's generation behavior (§3.5).

3.1 Models

Recent studies have explored the use of LLMs for AED and explanation generation in annotation tasks. Weber-Genzel et al. (2024) compare *GPT-3.5* (Brown et al., 2020) and *GPT-4* (OpenAI, 2023) for AED on the VariErr dataset. Similarly, Chen et al. (2025b) evaluate both open-source and closed-source instruction-tuned LLMs for the explanation generation process, including *Llama3-Chat-70b* (Grattafiori et al., 2024), *Mixtral8x7b-Instruct-v0.1* (Jiang et al., 2024), and *GPT-4o* (OpenAI, 2023).

Building on these studies, we adopt a comparable approach by incorporating both open-source and closed-source models. Specifically, we select *Llama-3.1-8B-Instruct*, *Llama-3.3-70B-Instruct*, and *GPT-4.1* due to their demonstrated capabilities in instruction following, which is critical in both generation and validation processes in our experiments. Our choice of GPT-4.1 is further justified by its superior performance over GPT-4o, particularly in coding and instruction following [1].

3.2 Explanation Generation

This section outlines the explanation generation process, covering both the setup (§3.2.1) and the generated outputs (§3.2.2).

3.2.1 Generation Setups

We base our experiments on the 500 examples from the VariErr dataset. These instances are the intersection of ChaosNLI and VariErr datasets, which contain both label distributions from 100 crowdworkers from ChaosNLI and annotations with explanations provided by four human experts from VariErr. This setup provides a rich ground truth for evaluating the quality and distribution of labels.

We follow the prompt design of Chen et al. (2025b) with modification, as shown in Figure 3.1. An LLM is prompted to generate all distinct explanations supporting a specified label for a given premise-hypothesis pair. Critically, in order to emulate human-written explanations, the prompt explicitly prohibits introductory phrases and disallows semantic repetitions, requiring each explanation to present a genuinely unique justification. The model is also given the option to abstain from responding if the requested label cannot be reasonably supported by the premise-hypothesis pair. This design balances thoroughness in explanation generation, while maintaining output validity.

[1] `https://openai.com/index/gpt-4-1/`

```
EXPLANATION GENERATION PROMPT
"role": "system", "content":
You are an expert in Natural Language Inference (NLI). List every distinct
explanation for why the statement is {relationship} given the context below
without introductory phrases.
If you think the relationship is false given the context, you can choose
not to provide explanations. Do not repeat or paraphrase the same idea in
different words. End your answer after all reasonable distinct explanations
are listed.
Format your answer as a numbered list (e.g., 1., 2., 3.)

"role": "user", "content":
Context: {promise}
Statement: {hypothesis}
```

Figure 3.1: Explanation generation prompt used on Llama models.

3.2.2 Generation Results

The statistics of the LLM-generation on 500 VariErr examples are shown in Table 3.1. We observe notable differences in generation behavior across models: while Llama-8B model tends to produce more explanations per label (on average 5.92), Llama-70B and GPT-4.1 generate relatively fewer explanations, averaging 2.69 and 2.19, respectively. Moreover, the average length of explanations also differs: Llama-70B and GPT-4.1 generate longer, more detailed explanations (23.34 and 22.75 tokens on average), whereas Llama-8B explanations are more concise (18.57 on average).

Model	# Expl.	Avg. Expl./Label	Avg. Length
Llama-3.3-70B-Instruct	4039	2.69	23.34
Llama-3.1-8B-Instruct	8884	5.92	18.57
GPT-4.1	3278	2.19	22.75

Table 3.1: Generation statistics on 500 VariErr examples. #Expl.: the total number of explanations. Avg. Expl./Label: average number per label. Avg. Length: average number of space-separated words per explanation.

3.3 Fallback Responses during Generation

We find two main issues with these LLM-generated explanations: the generation of fallback responses and repetitive explanations. Specifically, some explanations fail to justify the given label with valid reasons, while others repeat the same meaning across multiple explanations. As our pipeline aims not only to evaluate error detection performance but also to compare the alignment of model-generated distributions with human annotations, we intend to remove the low-quality and repetitive explanations to ensure a more accurate and cleaner distribution. This is based on the assumption that such issues occur infrequently in human-written explanations and may otherwise distort the intended distribution.

To address these issues, we first want to find explanations that potentially lack validity by screening for specific keywords. These selected explanations are then manually reviewed to determine whether they should be discarded for not supporting the corresponding label.

3.3.1 Identifying Non-Explanations in Generations

During our experiments with the original generation prompts proposed by Chen et al. (2025b), we observed frequent correctness issues in the generated explanations: the model may generate explanations despite internally disagreeing with the label under the premise-hypothesis pair, often resulting in explanations that are irrelevant or even contradictory to the label.

To mitigate these issues, we modify the prompt by adding the instruction: *"If you think the relationship is false given the context, you can choose not to provide explanations".* This adjustment partially alleviates the problem, as the models begin to refrain from generating misleading explanations when they disagree. However, rather than omitting the output entirely, models tend to respond with fallback statements like *"No explanations can be provided."* or *"Note: Since the statement is not supported by the context, there are no explanations for why the statement is true."*

This behavior is found across all three models. To detect such "non-explanation" outputs, we define a set of indicative keywords and use them to filter out a subset of the explanations that contain these keywords, as these keywords (*Note, None, No, no, explanation(s), distince, why*) are frequently found in responses where the model explicitly states that no explanation can be provided. We then manually verify whether all selected instances indeed correspond to non-explanations.

The verification step is performed before deduplication to prevent low-quality or irrelevant generations from negatively affecting the deduplication process, for instance, they may skew sentence embeddings and lead to the erroneous removal of valid explanations.

3.3.2 Fallback Responses Removal

Table 3.2 shows the number of identified non-explanation generations. GPT-4.1 generates significantly more outputs that indicate an explicit refusal to generate further explanations. In contrast, such refusal scenarios are much rarer in Llama models: Llama-70B yields only 33 filtered cases, and Llama-8B has the fewest filtered instances (only 3), despite producing the highest total number of generations.

This discrepancy may reflect differences in instruction-following capabilities across models. Smaller models like Llama-8B may struggle to correctly recognize when they have exhaustively listed all plausible justifications, leading to continued generation even when meaningful content is exhausted. In contrast, GPT-4.1 appears to interpret the instruction better to stop when no further valid explanation exists.

Model	**Non-Explanation**			
	E	**N**	**C**	**Sum**
Llama-3.3-70B-Instruct	11	7	15	33
Llama-3.1-8B-Instruct	2	1	0	3
GPT-4.1	165	0	31	196

Table 3.2: Number of non-explanation outputs identified through keyword filtering and manual verification.

Additionally, we observe label-specific trends: the majority of non-explanation generations are associated with the entailment (E) label, while neutral (N) has the fewest. This could be attributed to the nature of the labels themselves. Both "Entailment" and "Contradiction" labels have relatively strict and well-defined logical conditions, allowing the model to more confidently assess whether it has completely generated all reasonable explanations. In contrast, "Neutral" indicates a more ambiguous and looser relationship, making it easier to rationalize but harder for the model to determine if sufficient reasoning has been provided, thus reducing the chance of generating a non-explanation output.

Table 3.3 presents an example of the correctness check process, where the majority label for this instance is "N" (Neutral) in both the VariErr and ChaosNLI dataset and the model is asked to generate explanations for the label "E" (Entailment). Explanations marked in *italics* are removed after correctness check. Both Llama-70B and GPT-4.1 models generate non-explanation outputs, which are successfully identified and filtered out.

id	98621n
Context	In other cases, we must rely on survey approaches to estimate WTP, usually through a variant of the contingent valuation approach, which generally involves directly questioning respondents for their WTP in hypothetical market situations.
Statement	Hypothetical market situations are uniform across all respondents.
Label Distribution	`VariErr_round_2: {E: 0, N: 2, C: 1}` `ChaosNLI: {E: 7, N: 54, C: 39}`
Prompted Label	Entailment
`Llama-3.3-70B-Instruct`	1. The context does not support this statement, as it does not imply uniformity across respondents. *2. (Note: Since the relationship appears to be false given the context, no further explanations are provided.)*
`Llama-3.1-8B-Instruct`	1. The contingent valuation approach is standardized to ensure uniformity across all respondents. 2. The hypothetical market situations are designed to be identical for all respondents in the survey. 3. The survey is structured to present the same hypothetical market situations to all respondents. 4. The respondents are presented with the same set of hypothetical market scenarios. 5. The survey aims to minimize variability by using the same hypothetical market situations for all respondents.
GPT-4.1	*1. No explanations.*

Table 3.3: Examples of Correctness Verification. *Italic explanations* are removed after correctness check

3.3.3 Inconsistent Explanations with Label

However, current fallback responses filtering primarily addresses the "non-explanation" issue, there still exist certain explanations that are in contradiction with the given label upon closer human inspection (see explanation 1 in the Llama-70B generations in Table 3.3). These cases fall outside the defined filtering criteria and are therefore difficult to detect using the above filtering steps. We leave this challenge for future work, which may be addressed through more detailed prompting strategies and the incorporation of richer input contexts.

3.4 Explanation Deduplication

Although the instruction explicitly requests semantically distinct explanations, LLM generations frequently display high levels of repetition, as observed in our preliminary experiments.

Table 3.4 presents some examples of repetition at different levels: lexical, syntactic, and semantic. At the lexical level, explanations tend to reuse similar phrases or wordings. In this example, the four generated explanations all begin with identical phrasing *The*

id	109278n
Context	Lawyers in their first three years of practice or who are inactive pay $90, and retired lawyers pay nothing.
Statement	Lawyers pay $90 to be included in the directory.
Prompted Label	Contradiction
Llama-3.1-8B-Instruct	1. The statement does not specify that the lawyers must be active. *2. The statement does not specify that the lawyers must not be retired.* 3. The statement does not specify that the lawyers must have more than three years of practice experience. *4. The statement does not specify that the lawyers must not be inactive.*

(a) Examples of **lexical** repetitive explanations generated by Llama-3.1-8B-Instruct model.

id	5193n
Context	EPA estimates that 5.6 million acres of lakes, estuaries, and wetlands and 43,500 miles of streams, rivers, and coasts are impaired by mercury emissions.
Statement	The release of mercury has an impact on rivers, streams, and lakes
Prompted Label	Entailment
GPT-4.1	1. EPA data shows that a large number of acres of lakes, estuaries, and wetlands are impaired by mercury emissions, directly indicating an impact on lakes. *2. EPA data indicates that tens of thousands of miles of streams, rivers, and coasts are impaired by mercury emissions, directly indicating an impact on rivers and streams.*

(b) Examples of **syntactic** repetitive explanations generated by GPT-4.1 model.

id	47404e
Context	do you really romance
Statement	Do you really have an affair?
Prompted Label	Entailment
`Llama-3.3-70B-Instruct`	1. The term “romance” can imply a close, intimate relationship, which may not be limited to a committed partnership, potentially overlapping with the concept of having an affair. *2. “Romance” can be used to describe extramarital or non-committed intimate relationships, which is synonymous with having an affair.* *3. The word ”romance” might be used colloquially or euphemistically to refer to any kind of intimate or sexual relationship outside of a primary partnership, thereby encompassing the idea of an affair.*

(c) Examples of **semantic** repetitive explanations generated by Llama-3.3-70B-Instruct model.

Table 3.4: Examples of repetitive explanation. *Italic explanations* are considered repetitive and removed in each deduplication stage.

statement does not specify that the lawyers must.... At the syntactic level, explanations share similar sentence structures and always express the same meaning using different words. As in this example, *a large number of* and *tens of thousands of miles of* are synonyms. At the semantic level, repetition is more subtle. Explanations may appear different on the surface but express similar meanings. Here, all the explanations essentially highlight the semantic similarity between the terms *romance* and *affair*.

To mitigate redundancy, we implement a three-stage filtering process operating at the lexical, syntactic, and semantic levels on the generated explanations after correctness verification (§3.4.1). We then present the deduplication results (§3.4.2) and provide an analysis of representative examples (§3.4.3).

3.4.1 Deduplication Process

Previous studies (Giulianelli et al., 2023; Chen et al., 2025b; Hong et al., 2025) adopt a three-level framework (lexical, syntactic, and semantic) to assess similarity between texts. Building on this approach, we apply the same comparison process to the LLM-generated explanations in a pairwise manner after removing invalid fallback responses, in order to filter out redundant generations and ensure both diversity and quality.

1. **Lexical deduplication** Measured by 1-, 2-, and 3-gram word overlap. Two explanations e_1 and e_2 are considered lexically similar if:

$$1 - \frac{|\text{n-gram}(e_1) \cap \text{n-gram}(e_2)|}{|\text{n-gram}(e_1) \cup \text{n-gram}(e_2)|} < 0.5$$

   where n-gram(e) denotes the set of word-level n-grams. We average the similarity scores across $n = 1, 2, 3$. Higher scores indicate greater dissimilarity, pairs with lower dissimilarity (i.e., higher overlap) are filtered.

2. **Syntactic deduplication** Measured by 1-, 2-, and 3-gram part-of-speech (POS) tag overlap using spaCy's `en_core_web_md` (Honnibal et al., 2020). The same computation as lexical deduplication is used:

$$1 - \frac{|\text{POS-n-gram}(e_1) \cap \text{POS-n-gram}(e_2)|}{|\text{POS-n-gram}(e_1) \cup \text{POS-n-gram}(e_2)|} < 0.5$$

   where POS-n-gram(e) denotes the set of POS tag n-grams.

3. **Semantic deduplication** Measured by cosine similarity between sentence embeddings using the *sentence-transformers/all-MiniLM-L6-v2* model. A pair of explanations is considered semantically similar if:

$$\cos(\vec{e}_1, \vec{e}_2) > 0.9$$

   where $\vec{e}_i$ is the embedding vector of the explanation. In this case, lower similarity (i.e., cosine value ≤ 0.9) is desirable.

All the thresholds are manually selected based on empirical analysis. Deduplication is performed pairwisely among all explanations for the same prompted label generated by the same model. In cases where a pair of explanations is considered repetitive, the first one is retained and the second is discarded. This decision is based on the assumption that earlier generated outputs better represent the preferences of LLMs and are therefore more likely to capture the most salient reasons (Chen et al., 2025b).

3.4.2 Deduplication Results

Table 3.5 presents the number of retained explanations for each label category and their total counts at each stage of the preprocessing pipeline. The stages include the original generation output, after correctness verification, and after three deduplication steps—lexical, syntactic, and semantic.

Initially, explanations from all three models exhibit a near uniform distribution across the three NLI classes, as shown in Figure 3.2. Notably, these distributions remain relatively stable throughout the deduplication process.

Stage \ Model	**Llama-3.3-70B-Instruct**				**Llama-3.1-8B-Instruct**				GPT-4.1			
	E	N	C	Sum	E	N	C	Sum	E	N	C	Sum
Original	1289	1476	1274	**4039**	2823	3092	2969	**8884**	857	1326	1095	**3278**
Correctness	1278	1469	1259	**4006**	2821	3091	2969	**8881**	692	1326	1064	**3082**
Lexical	1278	1469	1259	**4006**	2795	3033	2925	**8753**	692	1326	1064	**3082**
Syntactic	1278	1469	1256	**4003**	2641	2841	2768	**8250**	691	1325	1064	**3080**
Semantic	1263	1454	1252	**3969**	2470	2716	2648	**7834**	685	1314	1060	**3059**

Table 3.5: Explanation counts (E/N/C) and total numbers across correctness verification and deduplication stages on the 500 VariErr examples.

Similar deduplication patterns can be observed in both the Llama-70B and GPT-4.1 models: no explanations are removed during the first step based on n-gram word overlap, and only a small number are filtered out in the second step using n-gram POS tag overlap. In contrast, the smaller Llama-8B model exhibits a greater reduction in explanations during the first two stages, likely due to its larger and more redundant outputs. It tends to include more superficial lexical or syntactical repetition that is easily captured.

This difference may reflect that the Llama-70B, which is an instruction-tuned model optimized for dialogue use cases[2] and the GPT-4.1, which is also shown to excel at instruction following, produce higher-quality outputs, particularly in following constraints such as avoiding repetition.

3.4.3 Deduplication Example Analysis

Table 3.6 and Table 3.7 present two examples of explanation deduplication process for both models. The selected examples are representative cases with low diversity scores (i.e., < 0.5), illustrating instances where redundancy is more prevalent. For each example, we include the corresponding label distributions from the VariErr and ChaosNLI datasets, in which the majority label is obvious.

Notably, explanations of "non-majority" labels tend to exhibit a lower diversity score. For instance, in example (a) in Table 3.6, the majority label is clearly "E", while the prompted label is "C", which conflicts with the more plausible choice. As a result, the generated explanations show really low diversity scores. A similar pattern appears in Table 3.7, where "E" is again the majority label, and low diversity scores are observed when the prompted label is "C" or "N".

This scenario can be attributed to the behavior of the models during generation: although the prompt explicitly allows models to refrain from generating explanations for implausible reasons, the models still generate explanations for ALL labels given and give a similar number of outputs. This often results in lower-quality generations, especially when the prompted label is less plausible. Consequently, the over-generation tends to yield more repeated or less informative explanations for these "non-majority" labels, which are then more likely to be filtered out during the deduplication process.

In the final (third) step, semantic similarity is assessed using cosine similarity of sentence embeddings. Figure 3.3 and figure 3.4 show the similarity matrices for the respective

[2] `https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct`

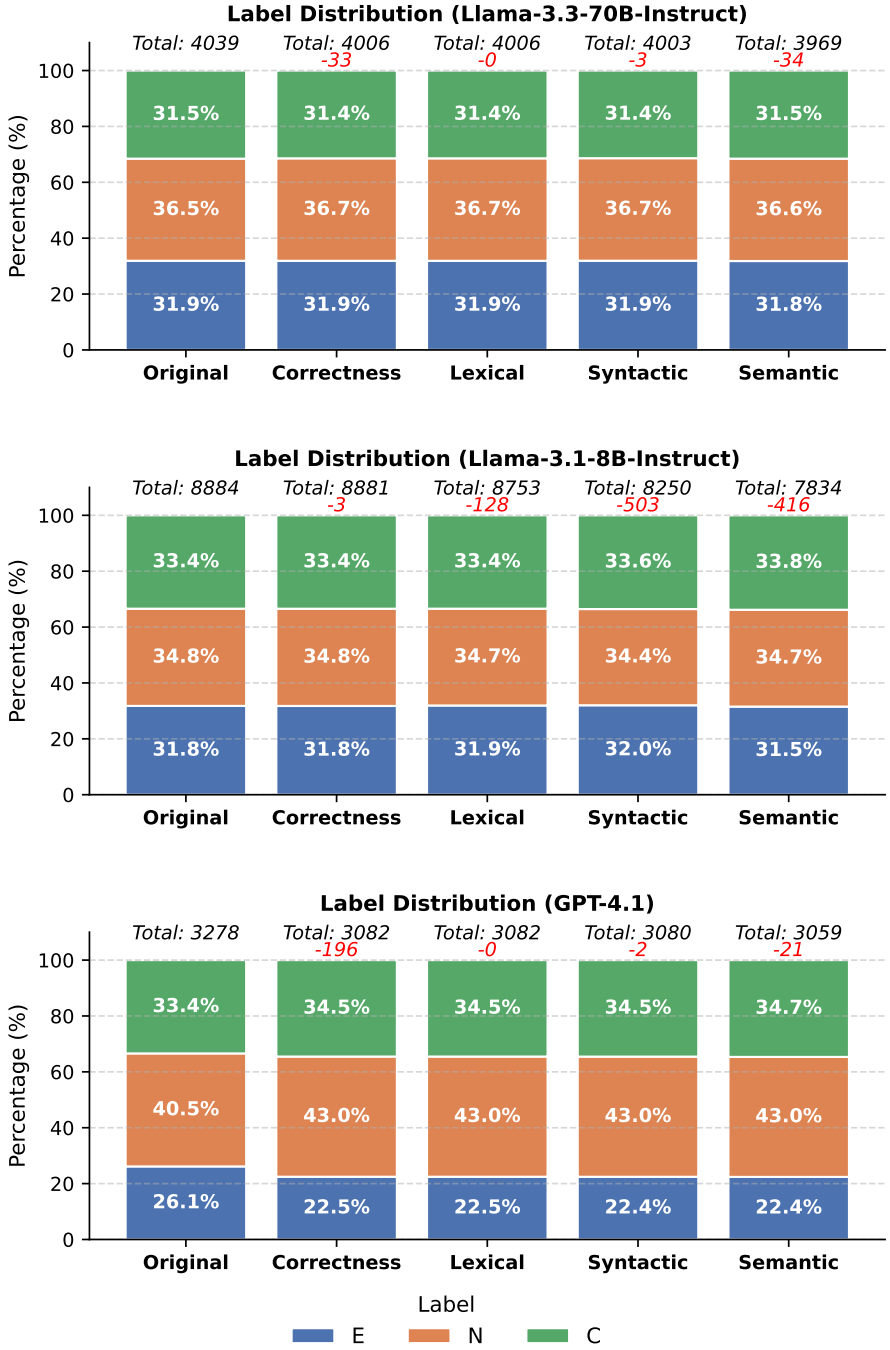


Figure 3.2: Label distribution across preprocessing stages, marked with the number of removed explanations.

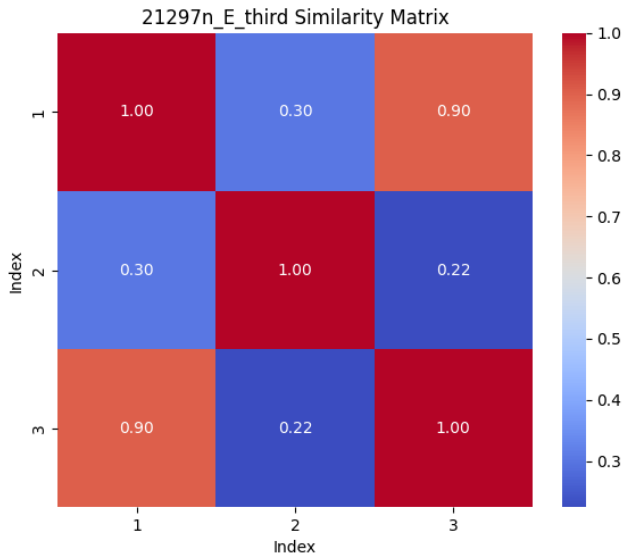


Figure 3.3: Sentence embedding similarity matrix for deduplication of explanations generated by the Llama-3.3-70B-Instruct model (instance `21297n`, label: *Entailment*).

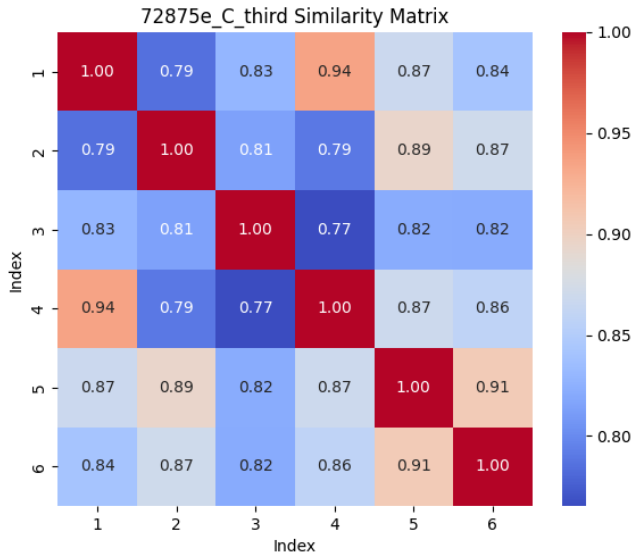


Figure 3.4: Sentence embedding similarity matrix for deduplication of explanations generated by the Llama-3.1-8B-Instruct model (instance `72875e`, label: *Contradiction*).

id	22587e	
Context	Classic Castilian restaurant.	
Statement	The restaurant is based off a classic Castilian style.	
Label Dist	`VariErr: {E: 3, N: 0, C: 0}` `ChaosNLI: {E: 90, N: 9, C: 1}`	
Step	Syntactic filtering: $\text{diversity}_{\text{n-gram}} < 0.5$	
Prompted Label: C	**Removed explanation:** The restaurant's name or branding does not reference Castilian roots.	**Diversity ↑** **1-gram:** 0.000 **2-gram:** 0.000 **3-gram:** 0.462
	Similar to explanation: The restaurant's decor does not reflect Castilian architecture or design.	

(a) Syntactic filtering based on n-gram diversity.

id	21297n	
Context	He was crying like his mother had just walloped him.	
Statement	He was crying like his mother hit him with a spoon.	
Label Dist	`VariErr: {E: 2, N: 2, C: 0}` `ChaosNLI: {E: 24, N: 74, C: 2}`	
Step	Semantic filtering: sentence embedding cosine similarity > 0.9	
Prompted Label: E	**Removed explanation:** "Walloped" can colloquially mean hit with an open hand or a blunt object, which could include a spoon.	See Figure 3.3
	Similar to explanation: The phrase "walloped" implies being hit, which is consistent with being hit with a spoon.	

(b) Semantic filtering using sentence embedding cosine similarity.

Table 3.6: Examples of syntactic and semantic deduplication process for the Llama-70B model. No explanations are removed in the lexical filtering process.

models. When a pair of explanations exceeds a cosine similarity threshold of 0.9, the later-generated explanation is removed to ensure semantic diversity.

3.5 Does Output Token Limit Affect Explanation Exhaustiveness?

When we compare the explanation distributions, it is crucial to ensure that the number of generated explanations is as complete and accurate as possible. However, when using Llama-8B for generation, we observe that the last explanation in some instances is truncated. This raises the question about whether "exhaustive" truly reflects the model's full range of possible explanations, or merely what can fit within the output token limit. Besides that, prior observations suggest that the number and average length of explanations differ a lot between different LLMs (Table 3.1), and one plausible factor contributing to this variation is the output token limit parameter during the generation process.

To examine the influence of output token limits, we compare generations with two different parameter settings: the default limit of 256 tokens used in prior experiments,

id	72875e	
Context	The policy succeeded, and I was fortunate to have had the opportunity to make that contribution to my people.	
Statement	Because the policy was a success, I was able to make a contribution to my people.	
Label Dist	`VariErr: {E: 3, N: 0, C: 1}` `ChaosNLI: {E: 88, N: 11, C: 1 }`	
Step	Lexical filtering: diversity$_{n\text{-gram}} < 0.5$	
Prompted Label: C	**Removed explanation:** The speaker's contribution to their people was a result of the policy's success, but it was not a significant contribution.	**Diversity ↑** **1-gram:** 0.190 **2-gram:** 0.320 **3-gram:** 0.333
	Similar to explanation: The speaker's contribution to their people was a result of the policy's success, but it was not a unique or notable contribution.	
Step	Syntactical filtering: diversity$_{n\text{-gram}} < 0.5$	
Prompted Label: N	**Removed explanation:** The statement's conclusion is too focused on the speaker's ability; the policy's success could have been achieved regardless of the speaker's contribution.	**Diversity ↑** **1-gram:** 0.000 **2-gram:** 0.250 **3-gram:** 0.300
	Similar to explanation: The statement's conclusion is too dependent on the speaker's role; the policy's success could have been achieved by others.	
Step	Semantic filtering: sentence embedding cosine similarity > 0.9	
Prompted Label: C	**Removed explanation:** The speaker's contribution to their people was a result of the policy's success, but it was not a unique or notable contribution.	see Figure 3.4
	Similar to explanation: The speaker's contribution to their people was a result of the policy's success, but the speaker did not have a significant role in the policy's success.	

Table 3.7: Example of explanation deduplication process for the Llama-3.1-8B-Instruct model. Lower diversity scores reflect greater n-gram overlap. In the final semantic step, similarity is assessed through a matrix rather than diversity metrics.

and an extended limit of 512 tokens. All other generation prompts and the parameters are kept constant to ensure a fair comparison. Two models are selected for this analysis: Llama-8B and Llama-70B.

For the Llama-8B model, we regenerate outputs for all 500 instances in the VariErr dataset. In comparison, for the Llama-70B model, we perform the same experiment only on a randomly selected subset of 25 examples of the VariErr dataset due to computational constraints.

Model	Token Limit	# Expl.	Avg. Expl./Label	Avg. Length
Llama-70B	256	4039	2.69	23.34
Llama-70B (subset of 25)	512	194	2.59	24.53
Llama-8B	256	8884	5.92	18.57
Llama-8B	512	8843	5.90	18.66

Table 3.8: Ablation study on the influence of output token limit.

Results are shown in Table 3.8. We can see that increasing the output token limit from 256 to 512 does not lead to an increase in the total number of generated explanations, the average number of explanations per label, or the average token number of each explanation. This indicates that the output token limit is not a crucial factor in determining explanation exhaustiveness. A limit of 256 tokens appears sufficient for the model to fully express its reasoning without interruption.

3.6 Interim Summary

In this chapter, we demonstrate that prompting large language models to generate explanations is effective. The generated content varies across models, each exhibiting distinct characteristics. However, we observe common issues such as fallback responses and repetitive content in the generations, which have been addressed in our study using keyword filtering and deduplication on different linguistic levels. Future work may further explore how to handle inconsistencies between the generated explanation and the prompted label.

Moreover, our ablation study shows that increasing the output token limit has minimal impact on the number or length of generated explanations, suggesting that tuning this parameter may not be necessary for future study.

4 LLM-Validation

In the LLM validation process, the LLM is provided with a premise, a hypothesis, and a label–explanation pair. Given the input context, it is prompted to assess how well the explanation justifies the label. The model outputs a plausibility score, and a predefined threshold is used to determine whether the explanation reflects an HLV (above threshold) or constitutes a potential annotation error to be discarded (below threshold). Notably, the same LLM is used for both generating and validating explanations, thereby simulating the self-validation process similar to human annotator behavior.

In this chapter, we introduce the experimental setup of the validation process (§4.1), followed by the presentation of validation results (§4.2) and a comparison of time efficiency between the LLM-involved and human-involved pipelines (§4.3).

4.1 Validation Setups

After the explanations are preprocessed and filtered for accuracy, we proceed to the validation stage using the same LLM. Given a premise–hypothesis pair, a label, and a single explanation, we prompt the LLMs to assign a validity score (ranging from 0.0 to 1.0) that reflects the extent to which the explanation justifies the label in light of the input pair.

We consider an EXPLANATION as "validated" if its validity score exceeds a predefined threshold; otherwise, it is labeled as "not-validated". To determine whether a LABEL is erroneous, we follow the criterion from VariErr: a label is considered "not-validated" if all associated explanations receive low validity scores.

Notably, both the generation and validation of explanations are performed by the same LLM, enabling a self-validation mechanism similar to that used in VariErr. This is important because the original definition of annotation errors in VariErr is also grounded in the self-validation process.

We adopt the prompt design proposed by Weber-Genzel et al. (2024) with some modifications. First, we omit the requirement to provide a reason for the score. Furthermore, unlike VariErr, our approach presents the model with only one explanation per evaluation instance for scoring due to limited input tokens, without access to other explanations. This adjustment is necessary as LLM-generated explanations tend to be longer and the numbers are usually more than the four human-written ones used in VariErr. The prompt is illustrated in Figure 4.1.

4.2 Validation Results

In this section, we examine how to set the threshold for determining whether an explanation is considered validated (§4.2.1) and provide some examples of explanations with a low score (§4.2.2).

4.2.1 Validation Threshold

As shown in Figure 4.2, model-assigned validity scores are generally high, with most explanations receiving scores of 0.8 or above. Based on this distribution, we set the validation threshold to 0.8: explanations with scores greater than or equal to 0.8 are marked as "validated", while those below are considered "not-validated".

We also find out that GPT-4.1 has the highest average score (0.9201) compared to Llama-70B (0.8481) and Llama-8B (0.7999). Moreover, GPT-4.1 assigns no explanation

```
EXPLANATION VALIDATION PROMPT
"role": "system", "content":
You are an expert linguistic annotator.
"role": "user", "content":
We have collected annotations for an NLI instance together with reasons for
the labels. Your task is to judge whether the reasons make sense for the
label. Provide the probability (0.0 - 1.0) that the reason makes sense for
the label. Give ONLY the probability, no other words or explanation.

For example:
Probability: <the probability between 0.0 and 1.0 that the reason makes
sense for the label, without any extra commentary whatsoever; just the
probability!>

Context: {promise}
Statement: {hypothesis}
Reason for label {label}: {reason_text}
Probability:
```

Figure 4.1: Explanation validation prompt used in this study.

a score of 0.0, in contrast to the Llama models, which still have several zero scores. This suggests that GPT-4.1 exhibits a lower tendency for self-rejection of explanations, possibly reflecting a more nuanced evaluation behavior or higher confidence in its outputs.

4.2.2 Examples of Explanations with Low Score

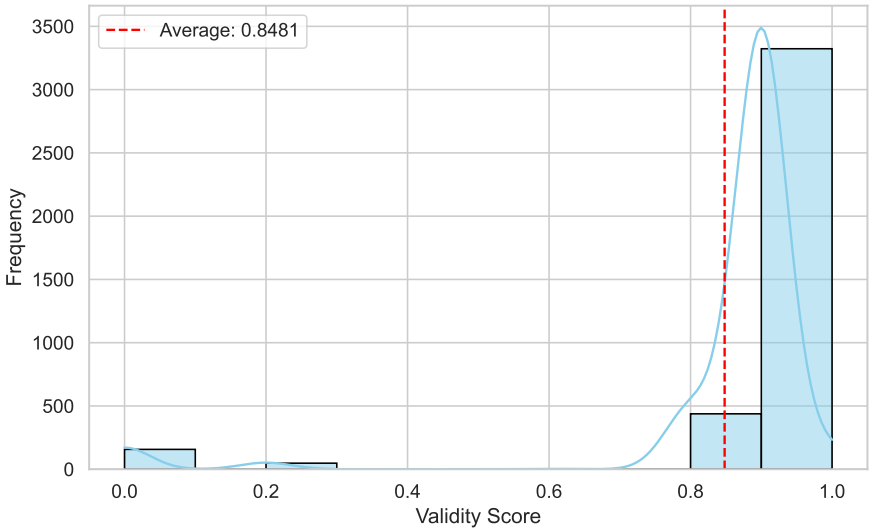
We check several examples with low validity scores. Example (b) in Table 4.1 receives the lowest score assigned by GPT-4.1 across all the generations. The explanation in this case suffers from a correctness issue: despite providing an extended analysis, it ultimately concludes that there are *"no reasonable distinct explanations for the statements"* This instance slipped through the preprocessing stage, as the misleading length makes it harder to detect during manual inspection. However, the low score still suggests that particular model is capable of identifying these inconsistencies by assigning a low validity score.

Besides that, we also observe that some explanations receive unexpectedly low scores, even though their semantic meaning appeared to align well with human explanations. The three explanations generated by Llama-70B, as well as the human-written examples in example (a) of Table 4.1, all convey the same idea that the asymmetry in quantity prevents the relationship from being determined, but the model still assigns low validity scores for the model-generated explanations. These cases are challenging to interpret, as the explanations have no obvious logical flaws or content errors. A possible reason is that the model may sometimes assign low scores due to subtle phrasing differences, which indicates a potential limitation in the model's interpretability.

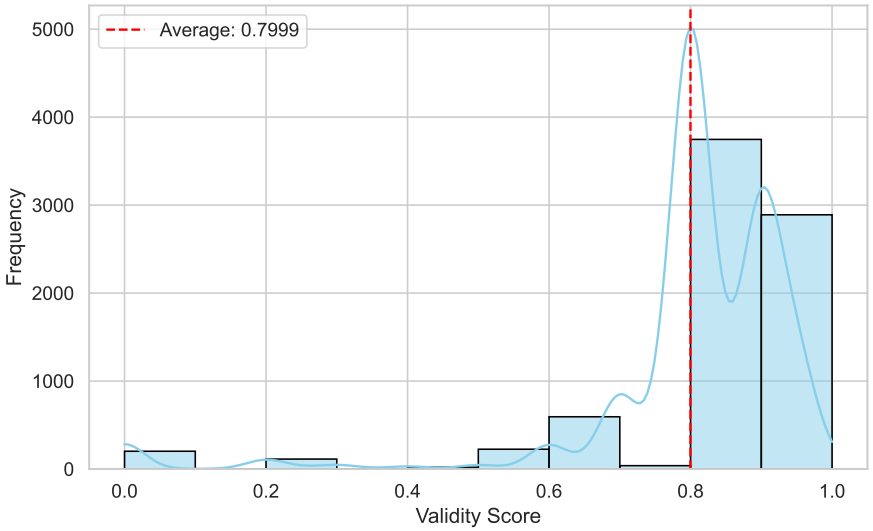
4.3 Computing Efficient

The replacement of human involvement with LLMs substantially reduces the overall time cost. For explanation generation, both Llama-8B and GPT-4.1 are able to generate explanations for 500 instances within one hour, while Llama-70B takes approximately three hours.

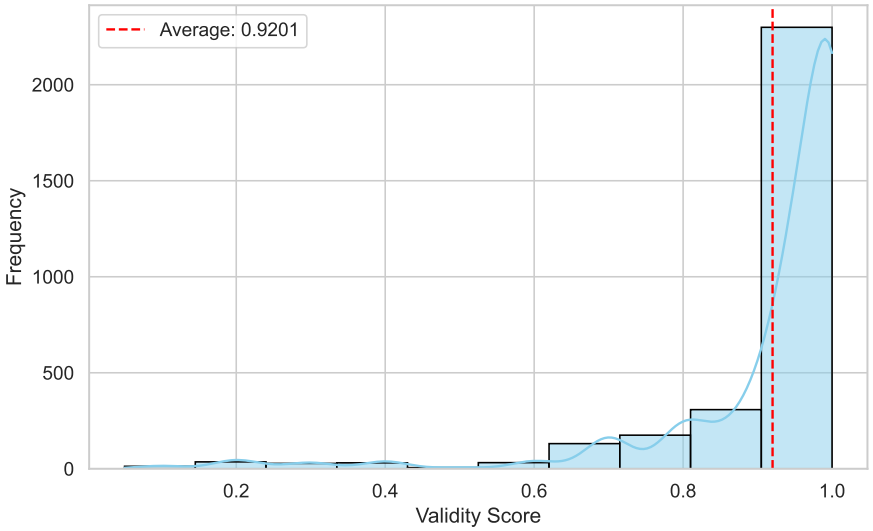
For explanation validation, Llama-8B processed around 8,000 instances in 15 minutes, whereas GPT-4.1 takes 30 minutes to validate around 3,000 instances. The more computationally intensive Llama-70B required about 30 hours to validate around 4,000 instances. In this study, GPT-4.1 is accessed via the OpenAI API, while the Llama models are run



(a) Llama-3.3-70B-Instruct validation score distribution.



(b) Llama-3.1-8B-Instruct validation score distribution.



(c) GPT-4.1 validation score distribution.

Figure 4.2: Validation score distributions across different models.

on NVIDIA A100 GPUs.

This represents a significant reduction in annotation and validation time compared to human experts. According to Weber-Genzel et al. (2024), each human annotator spent approximately 26 hours to annotate and about 15 hours to validate the same 500 items.

Our experiments demonstrate that, even with large models, LLM-based pipelines can largely reduce annotation time while maintaining high-quality output.

4.4 Interim Summary

Validation through LLM scoring proves to be a feasible approach. We observe that LLMs have strong confidence by assigning relatively high scores to their own generated explanations. Based on that, we set the validation threshold close to the average score (0.8).

However, a closer examination of unvalidated cases reveals certain issues. Some explanations receive low scores despite being comparable to human-written ones, which underscores the need for more interpretable mechanisms. Others show inconsistencies between the explanation and the prompted label. These inconsistencies are not always reliably identified by the model's scoring, highlighting the need for more careful examination before relying on LLM validation.

id	**13133n**	
Context	The newspaper publishes just one letter a week from a reader, always with an editorial riposte at the bottom.	
Statement	There are many letters submitted each week, but only one is chosen.	
Label Dist	`VariErr: {E: 0, N: 4, C: 0}` `ChaosNLI: {E: 50, N: 47, C: 3}`	
Prompted Label	Neutral	
Model Generated Explanations		Score
Llama-3.3-70B-Instruct	1. The context does not provide information about the number of letters submitted, only that one is published.	0.0
	2. It is possible that only one letter is submitted each week, which would be the one published.	0.0
	3. The presence of an editorial riposte does not imply multiple submissions, only a response to the published letter.	0.2
Human Written Explanations		
1. The newspaper publishes only one letter a week, either because they receive only one per week or because they receive many but choose only one. We don't know which is true.		
2. The context does not say anything about how many letters are submitted.		
3. We don't know whether there are many letters submitted.		
4. It's not known if many letters are submitted, but only known that only one is published.		

(a) Explanations with low score that semantically align with human-written explanations

id	**118403n**	
Context	oh really it wouldn't matter if we plant them when it was starting to get warmer	
Statement	It is better to plant when it is colder.	
Label Dist	`VariErr: {E: 0, N: 2, C: 2}` `ChaosNLI: {E: 7, N: 52, C: 41}`	
Prompted Label	Entailment	
Model Generated Explanations		Score
GPT-4.1	1. The phrase "it wouldn't matter if we plant them when it was starting to get warmer" implies that planting timing relative to temperature is not important, which does not directly support the claim that it is better to plant when it is colder. Therefore, there are no reasonable distinct explanations for the statement being true given this context.	0.05

(b) Incorrect explanations assigned with low validity score

Table 4.1: Examples of explanations with low validity scores.

5 Pipeline Comparison

After generation and validation, we compare the LLM-based and human-based pipelines from different angles. We first assess the distribution similarity between LLM-generated and human-written explanations, both before and after validation (§5.1), and also compare label rankings based on explanation counts for a smoother alignment measure (§5.2). We then adopt the ranking evaluation protocol from Weber-Genzel et al. (2024) (§5.3)and analyze the distribution and some examples of detected errors (§5.4).

5.1 Distribution Comparison

We compare the distribution of explanations between model-generated outputs and human annotations. Specifically:

- For the **VariErr** dataset, we compare the explanation distributions before and after LLM-validation with those from the first and second rounds of human annotation, respectively. This allows us to assess whether the distributions at each stage of the LLM pipeline align closely with human behaviors, indicating similarity across the entire process.
- For the **ChaosNLI** dataset, we compare the model distributions, before and after validation, against the annotation distribution from 100 human annotators, examining whether the validation step helps align model outputs more closely with human consensus.

To measure the discrepancy between distributions, we employ Jensen-Shannon Divergence (JSD, Endres and Schindelin 2003) and Kullback-Leibler Divergence (KLD, Kullback and Leibler 1951) following previous works (Chen et al., 2024, 2025b), which respectively measure the symmetric and asymmetric differences between probability distributions.

Results are shown in Table 5.1. We can observe that the model-generated distributions are more similar to those of ChaosNLI than to VariErr, as indicated by consistently lower scores in both JSD and KLD in all experiment settings. This is understandable, as the label distributions in ChaosNLI are softer with more annotators, while the distributions in VariErr tend to be more sparse and potentially extreme, often exhibiting dominant or zero-count labels for individual instances, which makes alignment more difficult.

Notably, decreases in JSD and KLD scores are observed in most comparisons with ChaosNLI after removing such errors, but not in the case of VariErr. This indicates that the validation process helps refine the label distribution to better reflect the natural variation found in human annotations.

5.2 Ranking Comparison

Inspired by Chen et al. (2025a), we also compare the relative rankings of labels instead of their absolute counts. For each instance, we derive a ranking over the labels E, N, C based on the number of associated explanations. In the presence of ties, we assign the same rank to the tied labels. For instance, if a instance has {“E”: 2, “N”: 1, “C”: 1} explanations, the resulting ranking would be {“E”: 1, “N”: 2, “C”: 2}. This approach mitigates the difference between the sparsity distribution in datasets like VariErr and the dense distribution in datasets like ChaoNLI. In VariErr, extreme values such as zero can

Model	Reference	Stage	JSD ↓	KLD ↓
Llama-3.3-70B-Instruct	chaosNLI	before	0.1352	2.0122
	chaosNLI	after	**0.1286**	**1.7961**
	VariErr_round_1	before	0.2680	14.8138
	VariErr_round_2	after	0.3138	17.2190
Llama-3.1-8B-Instruct	chaosNLI	before	0.1558	2.2362
	chaosNLI	after	**0.1505**	**2.1065**
	VariErr_round_1	before	0.2864.	15.4817
	VariErr_round_2	after	0.3455	18.5590
GPT-4.1	chaosNLI	before	0.1513	1.6277
	chaosNLI	after	0.1618	**1.4745**
	VariErr_round_1	before	0.2487	12.8860
	VariErr_round_2	after	0.2758	14.7455

Table 5.1: Distribution comparison (JSD and KLD) between LLM-generated explanations and human references across models and stages. Decreased scores are marked with **bold**

distort distribution-based metrics. We convert both human and model distributions into this format and then compare them.

To measure ranking similarity, we use Kendall's τ coefficient (Kendall, 1938), which captures the ordinal correlation between two ranked lists by counting the number of concordant and discordant pairs. As shown in Table 5.2, the resulting Kendall's τ coefficients are generally low after converting the label counts into rankings. Given that Kendall's τ ranges from -1 (complete disagreement) to 1 (perfect agreement), these values suggest that while some alignment exists, the overall ranking consistency remains limited.

However, we find a consistent improvement in Kendall's τ scores after validation, indicating that validation is still beneficial in leading the rankings more closely to the human annotation variations, further demonstrating the validation process's effectiveness in aligning model outputs with human annotation variations.

5.3 Evaluation of Annotation Error Identification

Following Weber-Genzel et al. (2024), we compute a score per label for each instance by averaging the plausibility scores across all explanations from one LLM for that label. The resulting average scores are used to rank in terms of their likelihood of being erroneous. This ranking is then compared against the self-flagged errors provided by human annotators. We report average precision (AP), as well as precision and recall at the top 100 predictions (P@100 and R@100). This evaluation allows us to quantify the extent to which LLM-detected errors align with human-identified issues, and serves as an additional approach for validating the effectiveness of our LLM-based error detection pipeline.

The results are shown in Table 5.3, with the scores of GPT-3.5 and GPT-4 taken from the original VariErr paper for reference. Among all AED models evaluated in that study, GPT-4 achieved the highest overall performance. In general, larger models demonstrate better performance. Our three tested models perform comparably to GPT-3.5, but none surpass GPT-4. Specifically, Llama-3.3-70B-Instruct and GPT-4.1 show slight improvements over the GPT-3.5 model across all the evaluation settings, but still fall behind GPT-4 and the human heuristics reported by Weber-Genzel et al. (2024).

However, there are also limitations of this evaluation setup. Since the metrics rely on score-based ranking of individual explanations, they may not fully capture the structure of errors. This evaluation may underestimate models that implicitly reject flawed expla-

Model	Reference	Stage	Kendall $\tau \uparrow$
Llama-3.3-70B-Instruct	chaosNLI	before	0.2806
	chaosNLI	after	**0.3301**
	VariErr_round_1	before	0.2472
	VariErr_round_2	after	**0.2788**
Llama-3.1-8B-Instruct	chaosNLI	before	-0.0052
	chaosNLI	after	**0.0535**
	VariErr_round_1	before	0.0207
	VariErr_round_2	after	-0.0111
GPT-4.1	chaosNLI	before	0.2045
	chaosNLI	after	**0.2485**
	VariErr_round_1	before	0.2376
	VariErr_round_2	after	**0.3275**

Table 5.2: Kendall's τ ranking correlation between LLM-generated explanations and human annotations across models and validation stages. Improved scores are marked with **bold**

nations without explicitly ranking them low.

For example, two instances with average scores of 0.75 and 0.05 would both be classified as errors under a fixed threshold. However, when applying ranking-based evaluation, the large difference in their scores introduces an unnecessarily nuanced distinction, while the actual evaluation only considers their relative order, not the semantic meaning of the scores. Moreover, top-k metrics can be sensitive to the long tail of the ranked list. As in our cases, true annotation errors are sparse or unevenly distributed, leading to unstable or uninformative evaluation outcomes.

Model	AP	P@100	R@100
Llama-3.3-70B-Instruct	18.6	24.0	18.6
Llama-3.1-8B-Instruct	15.6	19.0	14.7
GPT-4.1	22.1	24.0	22.0
GPT-3.5	17.6	21.0	16.3
GPT-4	46.5	46.0	35.9

Table 5.3: Performances of different models on error detection using ranking setups. Last two lines are borrowed from Weber-Genzel et al. (2024).

5.4 Error Analysis

5.4.1 LLM-detected Error Distribution

Model	E	N	C	Sum
Llama-3.3-70B-Instruct	9	8	19	36
Llama-3.1-8B-Instruct	4	2	4	10
GPT-4.1	58	0	18	76

Table 5.4: Label-level error counts detected by different LLMs on the VariErr dataset.

To better understand whether LLMs are capable of identifying errors with respect to exact error instances and structure, Table 5.4 presents the number of label-level (aggregated) errors detected by different LLMs on the VariErr dataset.

We observe that the number of errors detected by LLMs is substantially lower than those identified by human annotators in VariErr. This discrepancy arises because a label is considered erroneous only when ALL of its associated explanations are marked as invalid by the LLM. Consequently, errors are more likely to be detected in labels with only one explanation, which is more common in the VariErr dataset. In contrast, labels are associated with more LLM-generated explanations, making complete self-rejection harder under this strict definition.

The models display distinct behaviors. Notably, GPT-4.1 identifies the highest number of errors (76), with a substantial portion attributed to "E" and "C" labels, and none to "N". This scenario suggests that GPT-4.1 model exhibits highly consistent behavior during validation. When it deems a label implausible, it tends to assign low scores to all associated explanations. GPT-4.1 reports no errors under the `N` (neutral) label, which may relate to the intrinsic nature of neutral statements that we discussed before, which are often broader, making them harder to be absolutely invalid.

The human-detected number of errors distribution in VariErr is {"E": 52, "N": 23, "C": 53}. Among the three models, Llama-8B produces an error distribution most similar to this human baseline. This suggests that LLMs are able to reflect the nuanced variation seen in human annotations.

5.4.2 Disagreement in Error Judgements

We examine the detected errors and find that there is only minimal overlap between the errors identified by different LLMs, or between the LLMs and human annotators.

An example is shown in Table 5.5. In this case, both Llama models classified explanation "C" as an error, which aligns with the human judgment ("E", "C"). GPT does not identify any errors for this instance. However, it provides no explanation corresponding to label "C", which suggests that explanations associated with this label might have been filtered out during preprocessing due to being self-assessed as invalid. This raises a critical angle: should explanations that are entirely discarded during preprocessing (e.g., self-invalidated before being seen by the model) also be treated as latent errors and included in the error list?

Moreover, all explanations labeled "E" are self-validated by all three models in this example. However, the human annotator only provides one explanation, which is not self-validated but validated by two out of three peers. Since only one explanation exists for "E", the likelihood of it being rejected increases. This highlights a limitation: relying solely on self-validation may not be sufficient. Future studies could explore incorporating peer feedback or support from LLM decisions, thereby enabling a more robust error detection process.

5.5 Interim Summary

In this section, we compare the LLM-based with the human-based pipeline from multiple perspectives. In terms of distribution, whether we evaluate distributional similarity directly or convert the distributions into rankings, LLM-generated explanations show a closer alignment with the ChaosNLI distribution than those from VariErr, and this alignment improves further after validation.

Using evaluation protocols from prior studies, we find that the LLMs in our experiments perform comparably to GPT-3.5. Furthermore, a more detailed analysis of the detected errors reveals that the overall error distribution is also similar between certain LLM and human annotators, although notable disagreements remain on specific error instances.

id	118460n			
Context	and the other thing is the cost; it's almost prohibitive to bring it to a dealer			
Statement	The cost of fixing it makes it hard to bring it to a dealer.			
Classified Error	Llama-70B	Llama-8B	GPT-4.1	human
	C	C	None	E, C
Human Explanations	✗ 1. Both the context and statement talk about the large cost of fixing it. (E) ✓ 2. It's unclear whether "fixing it" increases the cost. (N) ✓ 3. The cost makes it hard to bring it to a dealer, but it could be the cost of fixing, or the cost of something else, like transport. (N) ✗ 4. It is the cost of bringing it to the dealer that is very expensive. (C)			
Llama-70B	✓ 1. The cost is almost prohibitive, implying that it is too expensive for the individual to afford bringing it to a dealer. (E) ✓ 2. The high cost creates a financial barrier that discourages the individual from seeking repair services at a dealer. (E) ✓ 3. The statement is a paraphrase of the context, so it is not undetermined. (N) ✗ 4. The statement is actually true given the context, as it directly reflects the information provided about the cost being älmost prohibitiveẗo bring it to a dealer. (C)			
Llama-8B	✓ 1. The cost of fixing the item is high, making it difficult to afford bringing it to a dealer. (E) ✓ 2.The expense associated with fixing the item is a significant barrier to bringing it to a dealer. (E) ✓ 3. Bringing the item to a dealer would require a substantial financial investment to fix it first. (E) ✓ 4. The cost of repair is a major obstacle to taking the item to a dealer. (E) ✓ 5. The statement does not specify what ïẗr̈efers to, making the relationship undetermined. (N) ✓ 6. The statement lacks information about the cost, making it unclear if the cost is indeed prohibitive. (N) ✗ 7. The statement implies that the cost is prohibitive but does not specify if the dealer is the only option for fixing the issue. (N) ✗ 8. The statement does not provide a clear causal link between the cost and the difficulty of bringing it to a dealer. (N) ✗ 9. The statement assumes that the cost of fixing is the main reason for not bringing it to a dealer, but other factors may also be involved. (N) ✗ 10. The statement is too vague to determine its relationship to the context. (C) ✗ 11. The cost of fixing is not the same as the cost of bringing it to a dealer. (C) ✗ 12. The statement does not specify that the cost of fixing is the reason it's hard to bring it to a dealer. (C)			
GPT-4.1	✓ 1. The context states that the cost is almost prohibitive, which means it is very high. (E) ✓ 2. The phrase älmost prohibitive to bring it to a dealerd̈irectly supports that the cost is a significant barrier to using the dealer's services. (E) ✓ 3. If the cost is almost prohibitive, then it makes bringing it to the dealer difficult or unlikely. (E) ✓ 4. The context mentions that the cost is älmost prohibitive,ẅhich suggests the cost is very high but does not confirm that it definitively makes it hard to bring it to a dealer. (N) ✓ 5. The statement assumes that the high cost specifically "makes it hard" to bring it to a dealer, but the context does not explicitly state that the difficulty in bringing it to a dealer is due to the cost alone; other factors may be involved. (N)			

Table 5.5: Example of different detected errors for one instance across human and LLMs.✓ denotes that the explanation is self-validated; ✗ denotes it is not notself-validatedd.

6 Downstream Application

To assess the practical impact of removing annotation errors, we fine-tune models on the NLI label distribution prediction task using different dataset variants. Additionally, we introduce synthetic noise to explore its relationship with annotation errors. We begin with the definition of annotation error and random noise used in this study(§6.1). We then present the experimental setups (§6.2), followed by the results of using dataset variant without human-detected errors (§6.3). Finally, we extend the analysis to LLM-detected errors to further support the effectiveness of our proposed LLM-based error detection framework (§6.4).

6.1 Definition of Error and Noise

In this work, we employ both the terms **annotation errors** and **annotation noise**, but use them to denote distinct concepts. Crucially, we introduce "annotation noise" specifically as an artificial, random replacement for "annotation errors" in our controlled experiments.

While some annotation disagreement arises from genuine ambiguity of the task, a portion of disagreement is still due to annotation errors. We refer to **annotation errors** as labels introduced during the annotation process that fail to reflect valid HLV. Such errors are commonly attributed to factors like annotator attention slips, cultural bias or a lack of understanding (Beigman and Beigman Klebanov, 2009; Plank, 2022). Specifically, we adopt the definition of annotation error from Weber-Genzel et al. (2024), who define a label as erroneous if all associated label–explanation pairs fail to be self-validated by the annotators in a two-round error-detection pipeline. This approach offers a structured way to separate plausible label variation from actual annotation mistakes.

In contrast, we define **random noise** in our study as labels generated through a purely stochastic process, uncorrelated with semantics or other textual features. We explicitly use such artificially generated random noise to replace identified annotation errors within our experimental framework.

6.2 Experimental Setups

In this section, we introduce the dataset variants used for the controlled study (§6.2.1), fine-tuning strategies (§6.2.2) and experimental hypothesis (§6.2.3).

6.2.1 Dataset Variants

In our study, we experiment with the VariErr dataset (Weber-Genzel et al., 2024) as the original dataset. To investigate whether annotation errors exert a similar influence as random noise when used as training data in NLI tasks, we conduct a set of controlled experiments using three variants of the dataset:

1. **Original dataset with annotation errors** – contains naturally occurring annotation errors.

2. **Error-free dataset** – constructed by removing identified annotation errors after human validation.

Based on the VariErr dataset, besides using self-validation as criterion, we further investigate an alternative variant in which errors are removed based on peer-validation results.

3. **Dataset with induced random noise** – generated by introducing synthetic noise into the error-free dataset.
   - Replace each label that is not validated (i.e. errors) with a random label selected uniformly from the label set (Zhang et al., 2017).
   - Replacing each detected error with a new label sampled based on the error label distribution reported in the VariErr dataset (see Table 6.1). This probabilistic sampling ensures that labels with higher frequencies in the error distribution are more likely to be selected, thus preserving the dataset's original noise characteristics. For example, given the aggregated distribution {"E":53, "N":23, "C":53}, each replacement label is sampled with approximately 37% probability for "E", 29% for "N", and 34% for "C".

FreqType	E	N	C	Sum
aggregated	53	23	53	129
repeated	87	69	82	238

Table 6.1: Error counts identified through human validation on the VariErr dataset.

6.2.2 Model Fine-Tuning

Following the experiment setups in Chen et al. (2024), we adopt small pre-trained language models, *bert-base-uncased* (Devlin et al., 2019) and *roberta-base* (Liu et al., 2019) as backbone models for fine-tuning on downstream NLI tasks.

Firstly, the models are fine-tuned on the large single-label MNLI training set (392k examples) and validated on the matched development set (9.8k examples) to learn the general structure of the NLI task. We then further fine-tune them on the label distributions derived from the VariErr dataset variants, enabling them to predict label distributions.

VariErr provides two types of label distributions: aggregated (label-level) and repeated (explanation-level). In our main experiments, we use the aggregated distributions as fine-tuning data across all variants, as the definition of annotation errors in the original paper is based on label-level decisions. For each aggregated instance, we construct the aggregated label distribution by assigning equal probability to each of the candidate E/N/C labels if they appear in the set and normalizing the values accordingly. For example, if a premise–hypothesis pair is annotated as "E" and "C" by different annotators, we assign a probability of 0.5 to each of these two labels and 0 to "N", resulting in the soft label vector: {"E": 0.5, "N": 0.0, "C": 0.5}.

In addition, we conduct supplementary experiments using the repeated distributions, where we directly use the original label frequency counts to form soft label distributions. This allows us to assess further the robustness of our findings under more fine-grained supervisions.

To evaluate the training performance, we use the 1,099 ChaonNLI instances that do not overlap with VariErr, and split them into development and test sets, containing 549 and 550 instances, respectively. We use the label distributions of 100 crowdworkers in ChaosNLI (Nie et al., 2020b) as the gold standard and evaluate model performance in predicting soft labels using KL-divergence (Kullback and Leibler, 1951) and weighted F1 scores following previous work (Chen et al., 2024, 2025b). KL-divergence measures the difference between the predicted label distribution and the human-annotated gold

distribution, with lower values indicating closer alignment. The hyperparameters used for fine-tuning are summarized in Table 6.2 and Table 6.3

Hyperparameter	
Learning Rate Decay	Linear
Weight Decay	0.0
Optimizer	AdamW
Max sequence length	128
Learning Rate	2e-5
Batch size	16
Num Epoch	3
Metric for best model	eval_accuracy

Table 6.2: Hyperparameter used for primary fine-tuning BERT and RoBERTa on the MNLI dataset.

Hyperparameter	
Learning Rate Decay	Linear
Weight Decay	0.0
Optimizer	AdamW
Learning Rate	2e-5
Batch size	4
Num Epoch	5
Metric for best model	eval_macro_F1

Table 6.3: Hyperparameter used for further fine-tuning BERT and RoBERTa on the Vari-Err dataset variants.

6.2.3 Experimental Hypothesis

We use the three variants as training data and validated them on ChaosNLI instances. Through this comparison, we want to prove two arguments:

1. **Impact of Annotation Errors on Model Performance** If the model trained on error-free dataset (variant 2) captures label distribution variance more accurately than the model trained on the original dataset with annotation error (variant 1), this indicates that error removal improves alignment with true crowdworker annotation patterns, reducing bias introduced by annotation errors.

2. **Noise vs. Error** If the performance of models trained on dataset with induced random noise (varient 3) closely resembles that of the model trained the original dataset with annotation error (variant 1), this suggests that annotation errors behave similarly to random noise rather than reflecting systematic HLV, and could therefore be removed.

These findings would prove that errors are non-informative and can be safely removed without distorting model performance.

6.3 Results

6.3.1 Impact of Annotation Errors on Alignment with HLV

The impact of annotation errors on model performance is presented in Table 6.4. We report both the weighted F1 score and the KL divergence on the development and test sets.

Error-free (Self) and **Error-free (Peer)** are dataset variants derived from the original VariErr dataset by removing annotations not validated in self- or peer-validation, respectively. In addition, we fine-tune models directly on ChaosNLI instances to form an upper-bound, which consistently yields the best overall results.

	BERT Fine-tuning				**RoBERTa Fine-tuning**			
	Weighted F1 ↑		**KL ↓**		**Weighted F1 ↑**		**KL ↓**	
Dataset	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
VariErr (Original)	**0.5613**	**0.5963**	**0.1736**	**0.1736**	**0.5973**	**0.6314**	**0.1782**	**0.1697**
Error-free (Self)	0.5411	0.5705	0.2559	0.2524	0.5863	0.6187	0.2688	0.2565
Error-free (Peer)	0.5211	0.5677	0.3822	0.3756	0.5802	0.6026	0.4420	0.4233
Upper bound								
ChaosNLI	0.6584	0.6526	0.0715	0.0714	0.6970	0.6990	0.0645	0.0552

Table 6.4: Performance comparison between fine-tuning on original and label-free variants, with the best score marked in **bold**.

Contrary to expectations, removing human-detected annotation errors does not lead to improved alignment with the crowdworker annotation distribution. In fact, all evaluation metrics get worse after training on error-free variants across both validation schemes. This may be attributed to the inherent sparsity in the VariErr dataset: since each instance is annotated by only four annotators, removing a single label can drastically alter the label distribution, e.g., by reducing the label probability to zero. Such removal may cause the resulting distribution to deviate sharply from the original soft label signal, leading to a less smooth distribution.

6.3.2 Thresholding the Evaluation Distributions

To better understand this issue, we attempted to transform the crowdworker annotation distributions in ChaosNLI into low-density annotation distributions as observed in the VariErr dataset. Our hypothesis is that fine-tuning is now conducted using the aggregated labels from the VariErr dataset as training data. Therefore, adjusting the label distributions in the development and test sets to match the sparsity of the training data may lead to more consistent fine-tuning and evaluation alignment.

Specifically, we applied a threshold on the ChaosNLI annotations: for each instance, only the labels that are chosen by more than a certain number of annotators (threshold) among the 100 crowdworker annotators are considered valid and included in the candidate label set. These selected labels are then assigned equal probability mass and normalized to form a uniform soft label distribution over the set, as when we process the original aggregated label sets in the VariErr dataset. For example, an instance from ChaosNLI with the annotation counts {“E”:31, “N”: 59, “C”:10} with a threshold of 20 would yield a label set of {“E”, “N”}, resulting in a transformed distribution of {“E”:0.5, “N”: 0.5, “C”:0.} This process preserves the relative dominance of label choices while enforcing a distributional sparsity consistent with VariErr, where only four annotators annotate each instance.

We experiment with threshold values of 20 and 30, as shown in Table 6.5. Unexpectedly, applying a threshold to the crowdworker label distributions in the ChaosNLI development

and test sets leads to consistently worse evaluation results, compared to using the original soft distributions. This suggests that removing infrequent but plausible labels from the gold distribution may discard meaningful instances and hurt the reliability of performance assessment.

	BERT Fine-tuning				**RoBERTa Fine-tuning**			
	Weighted F1 ↑		**KL ↓**		**Weighted F1 ↑**		**KL ↓**	
Dataset	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Threshold-20								
VariErr (Original)	**0.5296**	**0.5446**	**0.2138**	**0.2225**	**0.5613**	**0.5503**	**0.2091**	**0.2072**
Error-free (Self)	0.4727	0.5012	0.2888	0.2935	0.5387	0.5267	0.2963	0.2961
Error-free (Peer)	0.4820	0.4940	0.4059	0.4087	0.5307	0.5205	0.4612	0.4630
Threshold-30								
VariErr (Original)	**0.5543**	**0.5872**	**0.2434**	**0.2407**	**0.5943**	**0.6012**	**0.2332**	**0.2181**
Error-free (Self)	0.5080	0.5349	0.3174	0.3055	0.5686	0.5838	0.3142	0.2926
Error-free (Peer)	0.5120	0.5279	0.4329	0.4094	0.5621	0.5656	0.4719	0.4452

Table 6.5: Performance comparison under different label threshold settings (20 and 30), with the best score marked in **bold**.

We hypothesize that this is because, although the thresholded distributions appear more similar to the VariErr training data regarding label sparsity and composition, they lack the smoothness and nuance of the original distributions. On one hand, setting a threshold removes low-probability signals that may still carry useful uncertainty information; on the other hand, the distribution now can only be [0.5, 0.5, 0], [1, 0, 0], or [0.33 ,0.33, 0.33], which may cause a bigger discrepancy during evaluation. As a result, the modified distributions lead to worse evaluation performance, particularly in KL divergence, which directly measures soft-label alignment.

This suggests that modifying the development and test distributions to align with the training data artificially may not be effective. Instead, it highlights a limitation of our current setup: the training distributions are overly sparse and could benefit from being more softly informative. Making the training data more distributional, rather than flattening the evaluation sets, may be a more promising direction for future work.

6.3.3 Fine-Tuning with Repeated Annotations

The VariErr dataset also provides *repeated* label distribution, which directly counts how many of the four annotators selected each label. This allows us to use the resulting distribution without applying label set extraction followed by normalization as in *aggregated* label distributions in the previous section.

We assume that this count-based distribution is inherently softer than the normalized aggregated label set variant, as it captures more nuanced opinion differences. This may result in improved performance when used for training.

The results are shown in Table 6.6. Compared to Table 6.4, we observe a slight overall improvement over the results of fine-tuning with aggregated distributions, which indicates a better alignment with the HLV in the ChaosNLI dataset. However, the overall performance gain remains modest, as the error-free variants are still incapable of surpassing the original baseline. Therefore, to maintain consistency with prior work, we continue to use label-level (aggregated) distributions in the subsequent analysis.

	BERT Fine-tuning				RoBERTa Fine-tuning			
	Weighted F1 ↑		KL ↓		Weighted F1 ↑		KL ↓	
Dataset	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
VariErr (Original)	**0.5690**	**0.5916**	**0.1744**	**0.1758**	**0.5976**	0.6159	**0.1705**	**0.1611**
Error-free (Self)	0.5591	0.5631	0.2583	0.2569	0.5927	**0.6313**	0.2559	0.2470
Error-free (Peer)	0.5510	0.5740	0.3633	0.3639	0.5870	0.6094	0.4230	0.4060

Table 6.6: Performance comparison between datase variants (using *repeated* label distributions), with the best score marked in **bold**.

6.3.4 Noise Behaves Similar to Errors

To explore the role of annotation errors in the dataset, we replace the detected errors with either randomly selected noise labels or distributionally selected noise labels and compare their effects on model training performance. All experiments are conducted with three random seeds, and we report the average scores to ensure robustness.

Results are shown in Table 6.7 and Table 6.8. In Table 6.7, which evaluates the impact of replacing non-self-validated errors, all configurations show less than 0.05 difference in both weighted F1 score and KL divergence compared to the VariErr baseline, indicating that these errors influence model performance in a manner comparable to that of the noise. Similarly, Table 6.8 shows the results for replacing non-peer-validated errors. Despite slight fluctuations across seeds, the overall performance remains close to the original.

This further supports the conclusion that unvalidated labels functionally resemble noise in terms of their impact on downstream fine-tuning. Compared to the original VariErr dataset as training data, both random and distributional noise replacements lead to only minimal changes in performance. This suggests that the presence of annotation errors has a similar effect to that of artificially injected noise, despite being artificially injected randomly or according to certain error patterns. In other words, the detected errors behave more like random perturbations than meaningful signals, and have limited impact on the overall label distribution learned by the model.

	BERT Fine-tuning				RoBERTa Fine-tuning			
	Weighted F1 ↑		KL ↓		Weighted F1 ↑		KL ↓	
Dataset	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
VariErr (baseline)	0.5613	0.5963	0.1736	0.1736	0.5973	0.6314	0.1782	0.1697
random-noise-42	0.5450	0.5771	0.2055	0.2046	0.5911	0.6234	0.2054	0.1971
random-noise-43	0.5431	0.5854	0.1831	0.1844	0.5868	0.6021	0.2081	0.1996
random-noise-44	0.5589	0.5870	0.1856	0.1840	0.6056	0.6146	0.2013	0.1886
avg	**0.549**	**0.5832**	**0.1914**	**0.1910**	**0.5945**	**0.6134**	**0.2049**	**0.1951**
dist_noise-42	0.5566	0.5987	0.1912	0.1903	0.5981	0.6106	0.1961	0.1880
dist_noise-43	0.5467	0.5913	0.1936	0.1943	0.6003	0.6087	0.2026	0.1888
dist_noise-44	0.5615	0.5989	0.1780	0.1779	0.6041	0.6273	0.1668	0.1591
avg	**0.5549**	**0.5963**	**0.1882**	**0.1875**	**0.6008**	**0.6155**	**0.1885**	**0.1786**

Table 6.7: Results of replacing **not self-validated** labels with random or distributional noise (multiple seeds), with the average score marked in **bold**.

	BERT Fine-tuning				RoBERTa Fine-tuning			
	F1 ↑		KL ↓		F1 ↑		KL ↓	
Dataset	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
VariErr (baseline)	0.5613	0.5963	0.1736	0.1736	0.5973	0.6314	0.1782	0.1697
random-noise-42	0.5270	0.5844	0.1914	0.1913	0.5847	0.5934	0.2000	0.1905
random-noise-43	0.5248	0.5853	0.2084	0.2167	0.5803	0.6080	0.2016	0.1939
random-noise-44	0.5387	0.5832	0.2042	0.2067	0.5943	0.6108	0.2032	0.1991
avg	**0.5302**	**0.5843**	**0.2013**	**0.2049**	**0.5864**	**0.6041**	**0.2016**	**0.1945**
dist-noise-42	0.5470	0.5899	0.1988	0.2008	0.5706	0.5925	0.1900	0.1898
dist-noise-43	0.5661	0.5811	0.1984	0.1984	0.5784	0.6100	0.1863	0.1757
dist-noise-44	0.5678	0.6024	0.1833	0.1835	0.5936	0.6163	0.1827	0.1805
avg	**0.5603**	**0.5911**	**0.1935**	**0.1947**	**0.5809**	**0.6063**	**0.1863**	**0.1820**

Table 6.8: Results of replacing **not peer-validated** labels with random or distributional noise (multiple seeds), with the average score marked in **bold**.

6.4 Fine-Tuning with LLM-validated Labels

To further demonstrate the effectiveness of LLM-based error detection, we conduct a controlled experiment comparing two training data variants: one using the LLM-generated label distribution before validation, and the other using the distribution after validation. The experimental setups follow the protocol described before and apply the experiment across all three LLM generations, results are shown in Figure 6.9.

Model	**Error**	**BERT Fine-tuning**				**RoBERTa Fine-tuning**			
		Weighted F1 ↑		**KL ↓**		**Weighted F1 ↑**		**KL ↓**	
		Dev	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
Llama-70B	with	0.4046	0.4007	0.1174	0.1183	0.4992	0.4804	0.1157	0.1168
	without	**0.4955**	**0.5373**	**0.1117**	**0.1117**	**0.5743**	**0.6028**	**0.1077**	**0.1070**
Llama-8B	with	0.4023	0.4429	0.1175	0.1183	**0.4929**	0.4786	0.1155	0.1166
	without	**0.4838**	**0.5010**	**0.1150**	**0.1159**	0.4890	**0.4886**	**0.1138**	**0.1147**
GPT-4.1	with	**0.4955**	**0.5005**	**0.1324**	**0.1296**	**0.5020**	**0.5008**	**0.1297**	**0.1300**
	without	0.4369	0.4677	0.1715	0.1633	0.4149	0.4734	0.1725	0.1691

Table 6.9: Comparison of BERT and RoBERTa performance when fine-tuned on the VariErr dataset, with and without errors removed according to different LLMs.

Using the distribution after validation leads to consistently better evaluation performance across Llama models, suggesting that self-validation helps filter out low-quality or inconsistent annotations. However, the same effect is not observed for GPT-4.1, where performance slightly declines after applying validation. This contrast suggests that GPT-4.1's filtering may be more radical, potentially discarding useful diversity in annotations. These findings highlight the importance of model-specific validation strategies when leveraging LLMs for dataset refinement.

To further assess whether LLM-detected errors can be treated as noise that can be removed from the training set, we perform an additional comparison: for each of the three LLMs, we remove the LLM-identified error labels from the original first-round VariErr label set and use the remaining data for training. This is based on the assumption that error labels flagged by the LLM are also more likely to be noisy or inconsistent, and that removing them could lead to improved training quality. This setup is directly comparable

to the "error-free" results in Table 6.4. We aim to assess whether removing LLM-detected errors leads to better fine-tuning results than the original and error-free variants, thereby validating the usefulness of LLMs as error detectors.

	BERT Fine-tuning				**RoBERTa Fine-tuning**			
	Weighted F1 ↑		**KL ↓**		**Weighted F1 ↑**		**KL ↓**	
Dataset	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
VariErr (Original)	*0.5613*	*0.5963*	*0.1736*	*0.1736*	*0.5973*	0.6314	*0.1782*	*0.1697*
Error-free (Self)	0.5411	0.5705	0.2559	0.2524	0.5863	0.6187	0.2688	0.2565
Error-free (Llama-70B)	**0.5668**	0.5907	**0.1735**	**0.1716**	**0.6052**	0.6405	0.1806	0.1720
Error-free (Llama-8B)	0.5605	**0.5945**	0.1745	0.1727	0.5931	**0.6407**	**0.1783**	**0.1692**
Error-free (GPT-4.1)	0.5532	0.5667	0.1950	0.1930	0.5912	0.632	0.2023	0.1903

Table 6.10: Training without error detected by LLMs, with the best score marked in **bold**. The first two lines are borrowed from the previous chapter, which sets a comparison between the influence of LLM-detected and human-detected errors.

Results are shown in Table 6.10. For reference, we also include the original VariErr aggregated baseline and self-detected error-free results in the table. We observe that fine-tuning on the data after removing LLM-detected errors consistently yields better performance compared to the original baseline. This improvement is especially significant for the Llama models, where both BERT and RoBERTa benefit from the cleaned data across Weighted F1 and KLD scores. Notably, in our earlier experiments, filtering out labels that are not validated by either self- or peer annotators does not lead to performance gains over the baseline. This contrast suggests that LLMs may serve as effective assistants for human annotators in identifying instances that deviate from the distribution of multi-annotator agreement.

In such cases, using LLMs to filter out annotation error improves the quality of the dataset and subsequently benefits downstream task performance, which further proves that LLM-based validation can effectively enhance dataset quality, also when applied to human-annotated datasets.

6.5 Interim Summary

In this section, we demonstrate that annotation errors resemble random noise and that downstream models benefit from learning with softer, more nuanced label distributions, even if they are not strictly high-quality. We also show that errors identified by LLMs are indeed meaningful: removing them improves downstream performance. These findings highlight the importance of LLM-involvement in the AED pipeline. LLM can help not only in generating explanations to augment the original label distribution but also in effectively detecting annotation errors.

7 Conclusion

7.1 Experiment Summary

In this study, we explore whether large language models (LLMs) can effectively replicate the annotation and validation process of human experts in the explanation-based annotation error detection (AED) framework. Particularly for challenging tasks like natural language inference (NLI), where multiple annotations can be valid and reflect human label variation (HLV), the inconsistent annotation errors are harder to detect.

To address the scalability limitations of the prior study, we propose a fully LLM-based pipeline that performs both explanation generation and validation using an LLM. During generation, the LLM produces candidate explanations for each NLI label (entailment, neutral, contradiction) given a premise–hypothesis pair. Several preprocessing steps, including removal of invalid explanations and deduplication, are applied to improve explanation quality. In the validation phase, the same LLM assesses how well each explanation supports the corresponding label by assigning a plausibility score. An explanation is marked as invalid if its score falls below a threshold, and a label is considered erroneous if none of its associated explanations pass validation.

We evaluate the effectiveness of this LLM-involved pipeline by comparing it to the human-annotated and -validated VariErr benchmark from two aspects: distributional alignment and error detection. On the distributional side, we observe that LLM-generated explanations closely resemble the ChaosNLI distribution, which is often viewed as showing HLV. Furthermore, this similarity improves after applying validation. Additional analysis using ranking-based comparisons also shows that validation helps the LLM-generated distributions become more consistent with both ChaosNLI and VariErr.

From the perspective of error detection, although the exact error instances found by LLMs differ from those identified by human experts in VariErr, we still prove that LLM-detected errors are meaningful. First, error distributions from some models exhibit similar pattern to those from human annotations, suggesting that models are capable of identifying the types of examples likely to be perceived as an error. Second, when we remove LLM-detected errors from the original VariErr label set, and use it as training data to fine-tune downstream models, performance improves on the ChaosNLI development and test set. This improvement even surpasses what can be achieved by removing human-annotated errors from VariErr.

In general, our results suggest that a fully LLM-driven annotation and validation pipeline can approximate human performance in both simulating HLV and detecting annotation errors. While discrepancies in the final error sets remain, LLMs show potential in helping human annotators distinguish between true annotation errors and HLV. Annotators can consider combining multiple models to lead to more robust and scalable error detection pipelines.

7.2 Future Study

Building on our findings, several promising directions for future work emerge.

1. **Enhancing self-validation with more context** In the current validation process, the LLM is given only a single explanation along with the corresponding label, premise, and hypothesis. In contrast, the original VariErr presents annotators with all explanations provided in the first round during validation.

Due to input token limitations, our current setup restricts such comprehensive context. Future work could explore incorporating all explanations per label or even per instance during validation, enabling the LLM to form a more holistic view and more consistent and accurate behavior when making validation decisions.

2. **Enableing peer-validation** In addition to incorporating multiple explanations into the validation context, future work could explore integrating peer-validation by using LLMs to score not only the explanations they generate themselves, but also those generated by other LLMs. This approach would more closely mirror the original VariErr framework, where peer feedback also plays a crucial role in the validation process.

   One of our findings suggests that the discrepancy between the final error label sets of the human and LLM pipelines may stem from differences in explanation quantity: labels with only one explanation (as in VariErr) are more likely to be classified as error, even when peer opinions diverge. Therefore, leveraging peer information in both human-annotated VariErr and in the LLM-generated pipeline could help lead to more robust validation. However, this approach would place greater demands on the model's capacity to process longer context.

3. **Augmenting original dataset via LLM generation** One drawback of the current setup is the limited number of annotators in the VariErr dataset, which results in sparse annotation distributions. Our attempt to flatten the development/test sets using a threshold is unsuccessful, as the sparsity hinders the softer distributions.

   Given that LLM-based explanation generation and validation have proven effective, future work could consider augmenting the original VariErr dataset with additional explanations generated by LLMs. This hybrid approach could enrich the training data and lead to more robust evaluation and error detection outcomes.

References

Lora Aroyo and Chris Welty. 2015. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24.

Eyal Beigman and Beata Beigman Klebanov. 2009. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 280–287, Suntec, Singapore. Association for Computational Linguistics.

Beata Beigman Klebanov and Eyal Beigman. 2009. Squibs: From annotator agreement to noise models. *Computational Linguistics*, 35(4):495–503.

Beata Beigman Klebanov and Eyal Beigman. 2010. Some empirical evidence for annotation noise in a benchmarked dataset. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 438–446, Los Angeles, California. Association for Computational Linguistics.

Gabriel Bernier-Colborne and Sowmya Vajjala. 2024. Annotation errors and ner: A study with ontonotes 5.0.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Beiduo Chen, Yang Janet Liu, Anna Korhonen, and Barbara Plank. 2025a. Threading the needle: Reweaving chain-of-thought reasoning to explain human label variation. *arXiv preprint arXiv:2505.23368*.

Beiduo Chen, Siyao Peng, Anna Korhonen, and Barbara Plank. 2025b. A rose by any other name: Llm-generated explanations are good proxies for human explanations to collect label distributions on nli.

Beiduo Chen, Xinpeng Wang, Siyao Peng, Robert Litschko, Anna Korhonen, and Barbara Plank. 2024. "seeing the big through the small": Can LLMs approximate human judgment distributions on NLI from a few explanations? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14396–14419, Miami, Florida, USA. 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.

D.M. Endres and J.E. Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860.

Mario Giulianelli, Joris Baan, Wilker Aziz, Raquel Fernández, and Barbara Plank. 2023. What comes next? evaluating uncertainty in neural text generators against human production variability. In *Proceedings of the 2023 Conference on Empirical Methods in*

Natural Language Processing, pages 14349–14371, Singapore. 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, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Pingjun Hong, Beiduo Chen, Siyao Peng, Marie-Catherine de Marneffe, and Barbara Plank. 2025. Litex: A linguistic taxonomy of explanations for understanding within-label variation in natural language inference.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. *Zenodo*. If you use spaCy, please cite it as above.

Chathuri Jayaweera and Bonnie Dorr. 2025. From disagreement to understanding: The case for ambiguity detection in nli. *arXiv preprint arXiv:2507.15114*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Nan-Jiang Jiang and Marie-Catherine de Marneffe. 2022. Investigating reasons for disagreement in natural language inference. *Transactions of the Association for Computational Linguistics*, 10:1357–1374.

Nan-Jiang Jiang, Chenhao Tan, and Marie-Catherine de Marneffe. 2023. Ecologically valid explanations for label variation in NLI. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10622–10633, Singapore. Association for Computational Linguistics.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Jan-Christoph Klie, Bonnie Webber, and Iryna Gurevych. 2023. Annotation error detection: Analyzing the past and present for a more coherent future. *Computational Linguistics*, 49(1):157–198.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Jenny Kunz and Marco Kuhlmann. 2024. Properties and challenges of LLM-generated explanations. In *Proceedings of the Third Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 13–27, Mexico City, Mexico. Association for Computational Linguistics.

Stefan Larson, Adrian Cheung, Anish Mahendran, Kevin Leach, and Jonathan K. Kummerfeld. 2020. Inconsistencies in crowdsourced slot-filling annotations: A typology and identification methods. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5035–5046, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022. Explanations from large language models make small reasoners better.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Sebastian Lubos, Thi Ngoc Trang Tran, Alexander Felfernig, Seda Polat Erdeniz, and Viet-Man Le. 2024. Llm-generated explanations for recommender systems. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 276–285.

Lovish Madaan, David Esiobu, Pontus Stenetorp, Barbara Plank, and Dieuwke Hupkes. 2025. Lost in inference: Rediscovering the role of natural language inference for large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9229–9242, Albuquerque, New Mexico. Association for Computational Linguistics.

Jiafeng Mao, Qing Yu, Yoko Yamakata, and Kiyoharu Aizawa. 2021. Noisy annotation refinement for object detection. *arXiv preprint arXiv:2110.10456*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020a. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020b. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

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.

Ines Rehbein and Josef Ruppenhofer. 2017. Detecting annotation noise in automatically labelled data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1160–1170, Vancouver, Canada. Association for Computational Linguistics.

Frederick Reiss, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger. 2020. Identifying incorrect labels in the CoNLL-2003 corpus. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 215–226, Online. Association for Computational Linguistics.

Susanna Rücker and Alan Akbik. 2023. CleanCoNLL: A nearly noise-free named entity recognition dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8628–8645, Singapore. Association for Computational Linguistics.

Shane Storks, Qiaozi Gao, and Joyce Y. Chai. 2020. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2022. Learning from disagreement: A survey. *J. Artif. Int. Res.*, 72:1385–1470.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. 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 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Leon Weber, Robert Litschko, Ekaterina Artemova, and Barbara Plank. 2024. Donkii: Characterizing and detecting errors in instruction-tuning datasets. In *Proceedings of the 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 197–215, St. Julians, Malta. Association for Computational Linguistics.

Leon Weber-Genzel, Siyao Peng, Marie-Catherine De Marneffe, and Barbara Plank. 2024. VariErr NLI: Separating annotation error from human label variation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2256–2269, Bangkok, Thailand. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization.

List of Figures

List of Tables

Inhalt des beigelegten Software/Datenpackets

The materials accompanying this master thesis are made avaiable under `https://github.com/longfeizzz/Beyond-noise-MA-Zuo.git`. These include:

- `thesis/`: The PDF version of the thesis.
- `src/`: Contains the complete code for reproducing the experiments and results presented in this thesis.
- `scripts/`: Includes Python and shell scripts for data preprocessing, running experiments and evaluation tasks.
- `dataset/`: Contains all relevant datasets used in this study, including processed versions of VariErr and ChaosNLI.
- `results/`: Stores score files and evaluation results for reference and verification.
- `generation/`: LLM-generated explanations with preprocessing results.
- `README.md`: A guideline for setting up the environment and reproducing all experiments.