twinhter at LeWiDi-2025: Integrating Annotator Perspectives into BERT for Learning with Disagreements

Nguyen Huu Dang Nguyen[1,2] **and Dang Van Thin**[1,2]
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
23521045@gm.uit.edu.vn, thindv@uit.edu.vn

Abstract

Annotator-provided information during labeling can reflect differences in how texts are understood and interpreted, though such variation may also arise from inconsistencies or errors. To make use of this information, we build a BERT-based model that integrates annotator perspectives and evaluate it on four datasets from the third edition of the Learning With Disagreements (LeWiDi) shared task. For each original data point, we create a new (text, annotator) pair, optionally modifying the text to reflect the annotator's perspective when additional information is available. The text and annotator features are embedded separately and concatenated before classification, enabling the model to capture individual interpretations of the same input. Our model achieves first place on both tasks for the Par and VariErrNLI datasets. More broadly, it performs very well on datasets where annotators provide rich information and the number of annotators is relatively small, while still maintaining competitive results on datasets with limited annotator information and a larger annotator pool.

1 Introduction

Human language is often subjective and open to interpretation. In many NLP tasks, it's common for annotators to disagree sometimes for good reasons. But most traditional models ignore this variation and treat all labels as if there's only one correct answer. As a result, they may miss out on useful minority viewpoints and become less adaptable.

The third edition of the Learning With Disagreements (LeWiDi) shared task at EMNLP 2025 (Leonardelli et al., 2025) focuses on a critical challenge: building models that learn from disagreements rather than ignore them. The main objective of the task is to provide a unified evaluation framework for learning from disagreements. It introduces a benchmark including four datasets annotated with both soft labels and perspectivist annotations. Here, soft labels represent probability distributions over possible classes, capturing the degree of annotator disagreement, while perspectivist predictions aim to recover the individual label choices of each annotator. Participating teams are evaluated based on how accurately their models predict both types of outputs.

In our approach, we built a simple but effective BERT-based model (Devlin et al., 2019) that makes use of annotator perspectives during training. Instead of collapsing multiple labels into one, we create a separate training instance for each annotator's view and combine it with their background information. This way, the model learns to understand how different kinds of annotators might interpret the same input differently. Our approach performed well across all four shared task datasets. It was especially effective on tasks that involved a small set of annotators and provided natural language explanations alongside their labels.

2 Task Summary

2.1 Dataset

The LeWiDi 2025 shared task provides four diverse datasets across different NLP tasks. Each dataset is accompanied by annotator metadata, including basic demographic information about the annotators who provided the labels. See Table 1 for dataset statistics and Table 2 for available annotator metadata fields.

- **The Conversational Sarcasm Corpus (CSC)** (Jang and Frassinelli, 2024): A dataset of context–response pairs rated for sarcasm, with ratings from 1 to 6.
- **The MultiPico dataset (MP)** (Casola et al., 2024): A crowdsourced multilingual irony detection dataset. Annotators were tasked to detect whether a reply was ironic in the context of a brief post–reply exchange on social media.

Proceedings of the The 4th Workshop on Perspectivist Approaches to NLP (NLPerspectivists 2025), pages 249–255
November 8, 2025 ©2025 Association for Computational Linguistics

Table 1: Dataset statistics, including task type, instance counts for each split, and annotator information. Unseen annotators refer to annotators whose metadata is not provided.

Dataset	**CSC**	**MP**	**Par**	**VariErrNLI**
Task	Sarcasm Detection	Irony Detection	Paraphrase Detection	Natural Language Inference
No. of Instances				
Train	5628	12017	400	388
Dev	704	3005	50	50
Test	704	3756	50	50
Annotator Details				
Total annotators	840	506	4	4
Annotators / instance	4, 6	2 - 21	4	2, 3, 4
Unseen annotators	12	0	0	0

Field	**Description**	**Datasets**
Annotator ID	Unique identifier	All
Age	Annotator's age at the time of annotation	All
Gender	Self-identified gender	All
Nationality	Annotator's nationality	MP, Par, VariErrNLI
Education	Highest level of education completed	Par, VariErrNLI
Ethnicity (simplified)	The ethnicity of the annotator	MP
Country of birth	Annotator's country of birth	MP
Country of residence	Annotator's current country of residence	MP
Student status	Whether the annotator is a student	MP
Employment status	Annotator's employment status	MP

Table 2: Annotator metadata available across datasets.

Languages include Arabic, German, English, Spanish, French, Hindi, Italian, Dutch, and Portuguese.

- **The Paraphrase Detection dataset (Par)**: [1] A dataset of question pairs for which annotators rated whether the two questions are paraphrases of each other on a Likert scale. In addition to labels, annotators also provided short explanations for their choices.

- **The VariErr NLI dataset (VariErrNLI)** (Weber-Genzel et al., 2024): A dataset originally designed for automatic error detection, distinguishing between annotation errors and legitimate human label variation in Natural Language Inference. Annotators also included short textual explanations for their choices.

2.2 Tasks

The LeWiDi 2025 shared task defines two official evaluation settings. To ensure comparability with the leaderboard, we adopt the same metrics:

Task A (Soft Label Prediction): Given multiple annotator labels per instance, the goal is to predict a probability distribution over possible labels. Models are evaluated on how close the predicted label distribution is to the empirical human label distribution. Manhattan distance is used for binary label datasets (MP, VariErrNLI), and Wasserstein distance is used for ordinal label datasets (Par, CSC).

Task B (Perspectivist Prediction): This task focuses on predicting the individual labels assigned by each annotator. For binary label datasets (MP, VariErrNLI), performance is measured using error rate; for ordinal label datasets (Par, CSC), absolute distance is used.

3 Method

3.1 System Overview

Our model aims to capture how individual annotators see things differently. As shown in Figure 1, we convert each original sample into multiple training instances, each paired with information from a specific annotator. This lets the model pick up on patterns in how different people label the same text.

Dataset Construction: Instead of treating each sample as a single data point, we decompose it into multiple (text, annotator) pairs. Depending on the dataset, adjustments are applied to the input text (e.g., incorporating annotator explanations or source metadata) so that the model can capture how different annotators interpret the same input.

[1] The dataset is maintained by the MaiNLP lab and is not yet published.

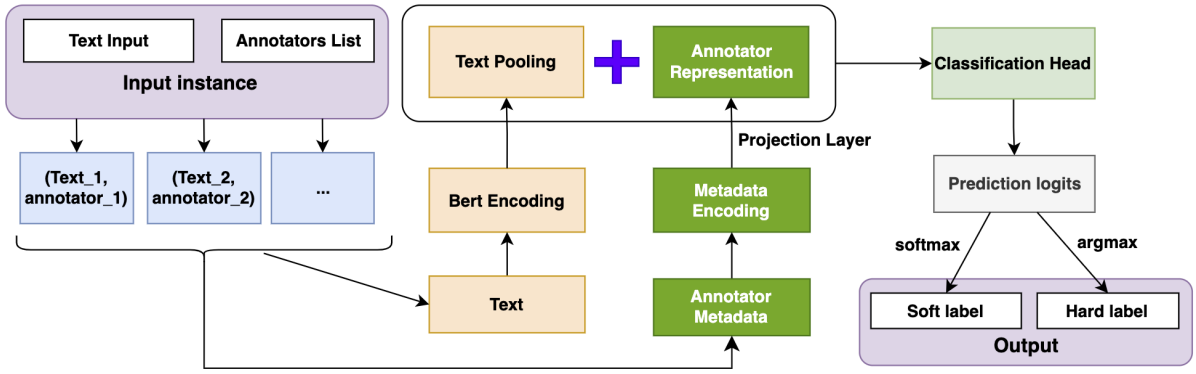

Figure 1: Representation of our BERT with Annotator Information

Detailed processing steps for each dataset are described in Section 3.2.

Input Representation: We encode the input text using a pretrained BERT model to obtain contextualized embeddings. In parallel, the annotator metadata is processed through a projection layer to produce a fixed-size feature vector. These two representations are then concatenated and passed to a classification layer.

Target Construction: Each (text, annotator) pair is treated as a distinct training sample with its corresponding label. This setup enables the model to learn from individual annotator perspectives.

Model Variants: We use MiniLM-L12-H384-uncased (Wang et al., 2020) for the CSC, Par, and VariErrNLI datasets, while DistilBERT-multilingual-cased (Sanh et al., 2019) is employed for MP, which contains multilingual samples.

Training Setup: We train the model using soft label supervision, comparing predictions to full label distributions. Optimization is performed using AdamW (Loshchilov and Hutter, 2019), with dropout, early stopping, and a learning rate scheduler to enhance training stability.

Baselines: We compare against three baselines: the official baseline from the organizers, a TF-IDF + Random Forest (Louppe, 2015) model, and a plain BERT model that doesn't use any annotator information. (furthur described in subsection **??**)

3.2 Text Processing

In all four datasets, each input sample is represented as a pair of textual fields, which we denote as `S1` and `S2`. The concrete meaning of these fields depends on the dataset:

- **CSC**: `S1` = context (the situation preceding the response), `S2` = response.
- **MP**: `S1` = post, `S2` = reply to the post.
- **Par**: `S1` = Question 1, `S2` = Questio n2.
- **VariErrNLI**: `S1` = context (premise), `S2` = statement (hypothesis).

These are concatenated using the `[SEP]` token:

```
S1 [SEP] S2
```

For the Par and VariErrNLI datasets, which include brief natural language explanations written by annotators, we append the explanation (`Exp`) of the corresponding annotator after a second `[SEP]` token:

```
S1 [SEP] S2 [SEP] Exp
```

For the MP dataset, which contains a source metadata field indicating the origin of the input (Reddit or Twitter), we prepend the source before the main text sequence to help the model disambiguate the context. This follows prior work on topic infusion (Sullivan et al., 2023):

```
Source [SEP] S1 [SEP] S2
```

Text Processing in Baseline Models: For the fine-tuned BERT baseline (which does not utilize annotator information), we concatenate all available annotator explanations (if present) and append them to the input sequence. This applies to datasets such as Par and VariErrNLI. For the TF-IDF + Random Forest baseline, we use the same input samples as in our main model, with tokenization performed using TF-IDF vectorization.

3.3 Annotator Metadata Encoding

Annotator metadata is encoded by combining one-hot encoding for categorical features and standard scaling for numerical ones. Missing or invalid values are imputed using the mode. The resulting feature vectors are concatenated into a single metadata representation for each annotator. For the MP dataset, we apply Principal Component Analysis (PCA) (Shlens, 2014), retaining 99.5% of the variance, to reduce the dimensionality from 91 to 31.

4 Experiment Setup

4.1 Comparison Models

We compare the Most Frequent baseline provided by the organizers with three approaches for modeling annotator disagreements and perspectives:

Organizer Baseline (Most Frequent): Two variants are provided by the organizers. (1) For Soft Label Evaluation, the mean label distribution over the training set is used as the prediction for all test items. (2) For Perspectivist Evaluation, each annotator's most frequent label is assigned across all items. Predictions are then evaluated using the respective metrics.

TF-IDF + Random Forest (TF-IDF + RF): For CSC, Par, and VariErrNLI, we extract TF-IDF features from the input text using TfidfVectorize and concatenate them with the annotator vectors. For Par and VariErrNLI, where the number of annotators is relatively small, we train an individual Random Forest regressor for each annotator to better reflect their subjective labeling tendencies. In contrast, for CSC, which includes over 800 annotators, we train a single model using soft labels aggregated across annotators. Due to the multilingual nature of MP, this model is not applicable there.

Fine-tuned BERT (No annotator Information): This baseline ignores annotator identity and treats each instance as a single aggregated sample. We fine-tune a BERT-based encoder using soft labels as targets. Specifically, we use MiniLM-L12-H384-uncased (Wang et al., 2020) for CSC, Par, and VariErrNLI; and DistilBERT-multilingual-cased (Sanh et al., 2019) for MP. This setup serves as a direct comparison point for evaluating the impact of annotator-aware modeling.

Fine-tuned BERT with Annotator Information (Main Model): The model described in subsection 3.1. It takes annotator information into account by treating each (text, annotator) pair as a distinct training sample. We encode the text using a BERT-based model and transform the annotator features via a projection layer. The two representations are then concatenated before classification.

For both **BERT-based models**, we use HuggingFace's `AutoTokenizer` (Wolf et al., 2020) associated with the respective pretrained encoder for text tokenization.

All models are trained using soft label supervision for Task A. For models that incorporate annotator information, we average predictions across annotators to obtain the final output. Predictions for Task B are then derived directly from the outputs of Task A. In contrast, models without annotator information generate a single output distribution, from which Task B labels are obtained via `argmax`.

4.2 Loss Function

For the CSC and Par datasets, which contain ordinal labels, we use Kullback-Leibler (KL) divergence loss for our model and the BERT baseline. The TF-IDF + Random Forest(RF) model is evaluated using the Wasserstein distance as a performance metric. For the MP and VariErrNLI datasets, which involve binary classification tasks, we use L1 loss for training. The TF-IDF + RF model for these datasets is evaluated using the Manhattan distance as a performance metric.

4.3 Evaluation Measures

Evaluation metrics are tailored to each dataset and task, and we follow the official definitions and evaluation scripts provided by the LeWiDi shared task organizers.

Soft Evaluation (Task A):

- **CSC, Par:** Average Wasserstein Distance
- **MP:** Average Manhattan Distance
- **VariErrNLI:** Average Multilabel Average Manhattan Distance

Perspectivist Evaluation (Task B):

- **CSC, Par:** Average Normalized Absolute Distance
- **MP:** Average Error Rate
- **VariErrNLI:** Average Multilabel Error Rate

Metric Summary: The evaluation metrics are designed to capture both aggregate performance (how well predicted distributions align with the overall human label distribution) and perspectivist performance (how well individual annotator perspectives are recovered). For brevity, we omit explicit formulas for commonly used metrics such as Manhattan Distance and Error Rate (and their multilabel variants). For readability, the metrics are presented in summarized form rather than with full mathematical expressions.

Wasserstein Distance (WD): Measures the effort required to transform one distribution into another, assuming ordinal classes:

$$WD(p,t) = \sum_{h=1}^{n}\sum_{k=1}^{n} \min(p_h, t_k) \cdot |h-k|$$

where p and t are discrete distributions over n ordinal categories.

Average Wasserstein Distance (AWD): Let $p^{(i)}$ and $t^{(i)}$ denote the predicted and target distributions for sample i, then:

$$AWD = \frac{1}{N}\sum_{i=1}^{N} WD(p^{(i)}, t^{(i)})$$

The average Wasserstein Distance is 0 in the case of a perfect match.

Normalized Absolute Distance (NAD): For a sample i, let $t_i = [t_{i,1}, \ldots, t_{i,a}]$ be the target labels and $p_i = [p_{i,1}, \ldots, p_{i,a}]$ the predictions for a annotators. The Normalized Absolute Distance is defined as:

$$\text{NAD}(i) = \frac{1}{a}\sum_{k=1}^{a} \frac{|t_{i,k} - p_{i,k}|}{s}$$

where s is the Likert scale range. A value of 0 indicates perfect agreement.

Average Normalized Absolute Distance (ANAD): The final score is obtained by averaging NAD over all N samples:

$$\text{ANAD} = \frac{1}{N}\sum_{i=1}^{N} \text{NAD}(i)$$

5 Result

We report system performance across all datasets and evaluation tasks in Table 3, using the metrics described in Section 4.3. Our model consistently outperforms baseline methods on the VariErrNLI and Par datasets, and shows modest improvements over baselines on CSC and MP. According to the official LeWiDi 2025 leaderboard[2], our system ranks **top-5** on both tasks for the CSC and MP datasets, and achieves **1st place** on both tasks for the Par and VariErrNLI datasets. These rankings are consistent across both Task A (soft evaluation) and Task B (perspectivist evaluation).

6 Further Analysis

We conducted further analysis to understand how incorporating annotator information affects model performance. Overall, models that leverage annotator information tend to outperform those that do not.

For the **Par** and **VariErrNLI** datasets, both of our annotator-aware models (TF-IDF + RF and our proposed BERT-based model) consistently surpassed the organizers' baselines and the BERT-based models without annotator information. With a small and fixed set of annotators, the models can more easily capture individual behavior, helping them understand consistency in how samples are labeled. Additionally, the inclusion of textual explanations allows the models to learn multiple perspectives from each annotator, resulting in richer, more multi-dimensional instance representations and reducing ambiguity compared to using raw labels alone.

In contrast, for the **MP** and **CSC** datasets, using annotator information did not lead to much improvement. These datasets only provide basic metadata (e.g., age, gender), and the number of annotators is much larger, making it harder for the model to learn how each annotator behaves. Moreover, some annotator attributes were missing or not provided for certain examples, so we filled in missing values using the mode for each field. This imputation may have introduced noise and reduced the effectiveness of annotator-aware modeling. Still, in **MP**, the model with annotator features performs slightly better. In the case of **CSC**, our proposed model performed worse than the BERT model that does not use annotator information. A likely explanation is that many annotators in **CSC** lack associated metadata. As a result, we had to fill in missing values with default values, which may

[2]More information about the shared task and leaderboard is available at https://le-wi-di.github.io/.

Table 3: System performance across datasets. **Task A** is evaluated using **WD** (Wasserstein Distance) and **MD** (Manhattan Distance). **Task B** is evaluated using **NAD** (Normalized Absolute Distance) and **ER** (Error Rate). Arrows ↓ indicate lower values represent better performance. The *Baseline* model is provided by the organizers. Best performance for each dataset and metric is highlighted in bold.

Model	**CSC**		**MP**		**Par**		**VariErrNLI**	
	WD(↓)	NAD(↓)	MD(↓)	ER(↓)	WD(↓)	NAD(↓)	MD(↓)	ER(↓)
Baseline	1.169	0.238	0.518	**0.316**	3.23	0.36	0.59	0.34
TF-IDF + Random Forest	0.87	0.247	x	x	1.2	0.34	0.42	0.24
BERT - without annotator information	**0.835**	x	0.48	x	2.04	x	0.4	x
BERT - with annotator information	0.86	**0.228**	**0.45**	0.319	**0.98**	**0.08**	**0.23**	**0.12**

have introduced noise into the input representation and negatively affected the model's performance. These results highlight a general challenge: when the annotator pool is large and metadata is sparse or missing, modeling individual annotator behavior may become difficult.

Model rankings in Task B largely reflect those in Task A, indicating that understanding annotator behavior contributes to overall prediction quality. While annotator-aware modeling benefits datasets with small, information-rich annotator pools, generalizing to larger, sparse pools remains challenging. These results suggest that the approach is most effective when annotator numbers are limited and data is semantically rich, but its effectiveness may decrease as the pool grows and label distributions become sparse, highlighting an open question for future research.

7 Conclusion

In this work, we presented a model that predicts labels for each (text, annotator) pair, aiming to capture individual annotator perspectives rather than just aggregated labels. We evaluated our method on four datasets covering sarcasm detection, irony detection, paraphrase detection, and natural language inference. Our results show that including annotator information often leads to better performance, especially in datasets where annotator perspectives are clearly defined and consistent. However, for datasets with many annotators or missing metadata, the improvement is less clear, and in some cases, using annotator features may introduce noise.

Overall, our findings suggest that modelling individual perspectives is a promising direction for tasks involving subjective annotation. Future work may explore more advanced architectures or evaluate on additional datasets to further understand the benefits and limitations of this approach.

Limitations

Our model has several limitations. First, some datasets (e.g., CSC) lack annotator metadata, requiring us to use dummy or average values, which may negatively affect the model's accuracy. Second, our model does not scale well to datasets with a large number of annotators, since each (text, annotator) pair is treated as a separate input. Third, we use a simple architecture that concatenates text and annotator embeddings, without exploring more advanced approaches like attention or expert mixtures. Lastly, we did not compare our approach against some strong solutions such as multi-task learning (Fornaciari et al., 2021), which could provide useful insights.

References

Silvia Casola, Simona Frenda, Soda Marem Lo, Erhan Sezerer, Antonio Uva, Valerio Basile, Cristina Bosco, Alessandro Pedrani, Chiara Rubagotti, Viviana Patti, and Davide Bernardi. 2024. MultiPICo: Multilingual perspectivist irony corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16008–16021, Bangkok, Thailand. 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.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597, Online. Association for Computational Linguistics.

Hyewon Jang and Diego Frassinelli. 2024. Generalizable sarcasm detection is just around the corner, of course! In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4238–4249, Mexico City, Mexico. Association for Computational Linguistics.

Elisa Leonardelli, Silvia Casola, Siyao Peng, Giulia Rizzi, Valerio Basile, Elisabetta Fersini, Diego Frassinelli, Hyewon Jang, Maja Pavlovic, Barbara Plank, and Massimo Poesio. 2025. Lewidi-2025 at nlperspectives: third edition of the learning with disagreements shared task. In *Proceedings of the 4th Workshop on Perspectivist Approaches to NLP (NLPerspectives)*. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Gilles Louppe. 2015. Understanding random forests: From theory to practice. *Preprint*, arXiv:1407.7502.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Jonathon Shlens. 2014. A tutorial on principal component analysis. *Preprint*, arXiv:1404.1100.

Michael Sullivan, Mohammed Yasin, and Cassandra L. Jacobs. 2023. University at buffalo at SemEval-2023 task 11: MASDA–modelling annotator sensibilities through DisAggregation. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 978–985, Toronto, Canada. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Appendix A: Hyperparameter Details

TF-IDF + Random Forest. For CSC, we use `TfidfVectorizer(max_features=4000)` for tokenization. For Par and VariErrNLI, we use `TfidfVectorizer(max_df=0.7, min_df=2, ngram_range=(1, 3))`. Random Forest hyperparameters (`n_estimators`, `max_depth`) are selected via grid search using the validation set.

BERT-based Models. All transformer-based models are optimized using AdamW (`weight decay=0.01`). Training is done with soft label regression.

CSC and MP: We train for 5 epochs with early stopping based on validation loss. We set dropout rate to 0.4, batch size to 32, and learning rate to 2e-5. We use `ReduceLROnPlateau` (`mode='min'`, `factor=0.5`, `patience=1`) . Texts are tokenized with `max_len=128`.

Par and VariErrNLI: Models are trained for up to 30 epochs with early stopping. Batch size is 16, dropout rate is 0.3 and learning rate is 2e-5. Texts are tokenized with `max_len=128`.

Classification Head: We use a linear layer followed by a dropout layer and another linear projection to the output logits.

Annotator Projection Layer: Annotator metadata is passed through a linear layer followed by a ReLU activation to obtain a fixed-size embedding vector.

Annotator Projection Sizes: 5 (CSC), 32 (MP), 16 (Par), 16 (VariErrNLI). The size of the text representation corresponds to the encoder's hidden size (e.g., 384 for MiniLM).