University at Buffalo at SemEval-2023 Task 11: MASDA–Modelling Annotator Sensibilities through DisAggregation

Michael J. Sullivan[†] **Mohammed Nasheed Yasin**[†] **Cassandra L. Jacobs**
Department of Linguistics
University at Buffalo
{mjs227;m44;cxjacobs}@buffalo.edu
[†] denotes equal contribution

Abstract

Modeling the most likely label when an annotation task is perspective-dependent discards relevant sources of variation that come from the annotators themselves. We present three approaches to modeling the controversiality of a particular text. First, we explicitly represented annotators using annotator embeddings to predict the training signals of each annotator's selections in addition to a majority class label. This method leads to reduction in error relative to models without these features, allowing the overall result to influence the weights of each annotator on the final prediction. In a second set of experiments, annotators were not modeled individually but instead annotator judgments were combined in a pairwise fashion that allowed us to implicitly combine annotators. Overall, we found that aggregating and explicitly comparing annotators' responses to a static document representation produced high-quality predictions in all datasets, though some systems struggle to account for large or variable numbers of annotators.

1 Introduction

The primary challenge of accommodating disagreement in natural language processing is in aggregating (or disaggregating) the decisions of individual annotators. As part of this submission to LeWiDi23 (Leonardellli et al., 2023), we present several possible approaches, deployed to varying degrees of success, for modeling annotators individually or in parallel. Building on the general idea that individual annotators largely process language in similar ways, minus differences of political opinion or life experience, we create features that allow us to compare and contrast annotators to each other directly.

Our first approach creates annotator layers that represent individual annotators using small-dimensionality embeddings. The decisions of the annotators (their personal label) then impact the final (gold) label. These separate networks enable the joint prediction of each annotator while considering the sentence embedding separately. We present two ablation studies showing that integrating the features in this way is critical for performance of the system.

2 Approach 1: Annotator Layers and Document Embeddings

The model consists of a pretrained sentence transformer coupled with a classification head (see Figure 1). We use this template for the HS-Brexit (Akhtar et al., 2021) and ArMIS (Almanea and Poesio, 2022) datasets, each modified to account for the main differences in the number of annotators and the language —we refer interested readers to the Appendix for dataset-specific modifications to this architecture.

The model predicts a soft label for a given input, which is defined as the average of the predicted labels for each individual annotator (i.e. ensemble averaging; Dietterich, 2000). Aach annotator for the dataset is represented by a unique linear layer (*AnnNet*), which produces an embedding from the sentence transformer that reflects that annotator's likely decision on the input text as a proxy of their viewpoint on the topic. The final (sigmoid) classification layer (*DecNet*) produces a predicted label from each annotator embedding, which is shared by all of the annotator layers.

To predict the hard label, an additional layer (*GoldNet*) produces a weighted sum of the annotator embeddings in the form of attention weights. These annotator attention weights are learned over the sentence transformer output. This weighted sum of the *GoldNet* is then passed to the *DecNet* to produce the predicted hard label. The *GoldNet* attention mechanism therefore identifies which annotators' labels align with the gold/hard labels on different kinds of sentence inputs.

Note that the collection of annotator networks is similar to the *crowd layer* introduced by Rodrigues

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 978–985

and Pereira (2018); the unique contribution of our architecture lies in the *GoldNet* aggregation layer and the parameter-sharing implemented by the *DecNet* layer.

2.1 Model Architecture

The model first passes each textual input t through a pre-trained sentence transformer and produces a d_{ST}-dimensional embedding $e(t)$ via mean-pooling over each token embedding. Each annotator a_i is represented by a $d_{ST} \times d_{Ann}$ linear layer ($AnnNet_i$). The annotator latent dimension d_{Ann} varies across datasets and was tuned manually. The *annotator embeddings* $AnnNet_i(e(t))$ are each passed through the $d_{Ann} \times 1$ *decision layer* (*DecNet*; with sigmoid activation) to produce a prediction of the annotator a_i's label for the input t. The single *DecNet* is shared across all annotator latent embeddings to induce parameter sharing. This produces low-dimensional annotator embeddings whose values represent different latent factors that represent the annotator's sensitivity to different aspects of t (i.e., hate speech or textual misogyny). To produce the soft label prediction, the model averages over the annotator label predictions from each annotator (see Equation 1).

$$SoftLabel = \frac{\sum_{i=1}^{n} DecNet(AnnNet_i(e(t)))}{n} \quad (1)$$

To obtain the predicted hard label for a given t and a, the model incorporates the *GoldNet* module (see Figure 1), which takes as input the text embedding $e(t)$ and each of the n annotator latent embeddings $AnnNet_i(e(t))$. The *GoldNet* attention mechanism consists of a $d_{ST} \times n$ layer with softmax activation. Thus, given the text embedding $e(t)$, the attention mechanism produces an n-dimensional vector $\alpha(e(t))$, such that $\alpha(e(t))_i$ encodes the weight that *GoldNet* places on the i^{th} annotator embedding. The output of the *GoldNet* is the *gold embedding* $g(t)$, which is the attention-weighted sum of the annotator embeddings (see Equation 2).

$$g(t) = \sum_{i=1}^{n} AnnNet_i(e(t)) \cdot \alpha(e(t))_i \quad (2)$$

The gold latent embedding $g(t)$ is then passed through the *DecNet*; parameter sharing in this layer ensures that each dimension of the annotator latent embeddings corresponds to the same latent factor of hate speech/misogyny, ensuring that the attention-weighted $g(t)$ is effectively a weighted average of the annotators' opinions about t. The idea here is that the *GoldNet* attention mechanism learns which annotators' labels align with the gold/hard labels on different kinds of inputs.

To produce the final hard label for evaluation, the model rounds the output to 0 if $DecNet(g(t)) < 0.5$, and to 1 otherwise.

2.2 Training and Dataset Considerations

To train the network, we use binary cross-entropy loss with respect to the hard label and the individual annotator labels. We did not directly optimize with respect to the soft label itself as in Uma et al. (2021) and Almanea and Poesio (2022), as we found that our method yielded better results; in particular, Almanea and Poesio (2022) report a cross-entropy (CE) score of 0.586 on ArMIS when directly optimizing with respect to the soft label, while our method (optimizing with respect to individual annotator labels) yields a cross-entropy score of 0.548 for that same dataset (see Table 1). For both datasets, we train with dropout (probability of 0.5) on all layers.

During training, the *GoldNet* is entirely isolated from the rest of the model architecture; the gradient does not flow from the *GoldNet* to the annotator networks or the sentence transformer. This is because, for example, the *GoldNet* attention mechanism may assign the majority of its attention to an annotator a_i that does not agree with the hard label for the input in question. If the gradient were to flow from the *GoldNet* to $AnnNet_i$, then $AnnNet_i$ would receive conflicting error signals from the two sources: its loss with respect to the a_i label, and its loss with respect to the hard label.

For both the HS-Brexit and ArMIS datasets, there is a fixed number of annotators (six and three, respectively); each annotator annotates each example (so each example has six and three annotator labels for HS-Brexit and ArMIS, respectively), and the same annotators annotate the train, test, and evaluation datasets. This is the biggest drawback of this architecture, as it relies on the fixed number of annotators and therefore cannot be implemented for the ConvAbuse or MD-Agreement datasets.

2.3 Ablation Studies

In this subsection, we perform two ablation studies on the models defined in the above section. These two ablation studies are intended to probe the utility of isolating the *GoldNet* from the rest of the

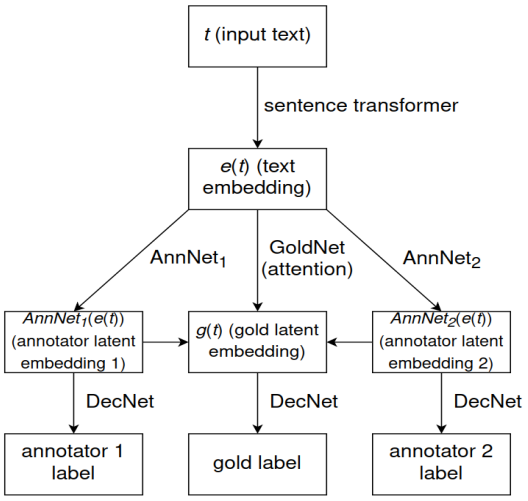

Figure 1: Example of our Approach 1 model architecture (with two annotators for readability).

model with respect to the gradient, and of the the *GoldNet* in general. In the first study (*GoldNet-Grad*), we allow the gradient to flow from the *GoldNet* to the sentence transformer and the annotator networks. In the second (*No-GoldNet*), we remove the *GoldNet* from the architecture entirely, and estimate the predicted hard label by averaging over the predicted labels for each individual annotator. In both ablation studies, the training hyperparameters (number of epochs, learning rate, weight decay value, and optimizer) remained the same as in the full models.

The results of the ablation studies are shown in Table 1. Across both metrics and both datasets, passing the gradient from the *GoldNet* layer to the remainder of the architecture results in a decrease in model performance. Similarly, entirely removing the *GoldNet* from the model results in an increase in cross-entropy and a decrease in F1 on both datasets.

3 Approach 2: Topic Infusion

This method is used primarily in the MD-Agreement (Leonardelli et al., 2021) dataset. We introduce the *polarizability* metric which reflects the ability of a tweet to divide annotator opinion (see Equation 3). Our implementation of the *polarizability* metric is based on the *polarization measure* discussed in (Akhtar et al., 2019). However, the original metric relies on a sophisticated model of group dynamics, which can be challenging to obtain without significant data and expertise.

To mitigate this issue, we have designed our *polarizability* metric to be a plug-and-play feature, which can be easily replaced with other implementations. While we acknowledge the potential benefits of incorporating a more accurate model of group dynamics, we leave this as an area for future exploration.

$$polarizability = 1 - |prop(pos) - prop(neg)| \quad (3)$$

The $prop(pos)$ and $prop(neg)$ indicate the fraction of annotators that assigned a positive and negative label respectively, resulting in a value that ranged from 0 (unanimity) to 1 (an even split). We found that different tweets topics varied in polarizability (see Table 2), leading us to include it as a feature to our model (see Figure 2).

3.1 Design Considerations

In our approach, we represented tweets as plain text blocks. However, we also recognized that the topic of a tweet can be an important factor in determining its sentiment or meaning. Therefore, we chose to incorporate the tweet's topic into the text encoding by separating it from the rest of the tweet using a special token called "[SEP]" (refer Figure 2). By including the tweet's topic in the text encoding, we were able to capture more of the nuances and context of the text, leading to better performance overall. The technique has been adapted from the one used in Xiong et al. (2021) to improve generic text classification.

We used sentence encodings from MiniLM (Wang et al., 2020) as inputs to a linear layer to predict the label distribution. We minimize the KL divergence loss to train our model as this is equivalent to explicitly modeling the cross-entropy (Shlens, 2014). We utilize this approach due to the continuous nature of the ground truth labels, as opposed to the typical one-hot targets used in classification. In this scenario, the minimum cross entropy value achieved at convergence would be influenced by the distribution of the target label `[abuse, non-abuse]`. In this context, the KL-divergence measures the similarity between the distribution of the ground truth labels and the predicted distribution generated by the model. Finally, given the small data setup, we chose to use a fine-tuning approach with a smaller learning rate.

The MD-Agreement had over 700 unique annotators with the average annotation count being less than 15. To accurately represent the annotators, we need a larger sample size than just a few per annotator. Therefore, we decided to use a simpler model

	Full Model	GoldNet-Grad	No-GoldNet	Organizers' Baseline
HS-Brexit (CE):	**0.295**	0.302	0.300	2.710
HS-Brexit (F1):	**0.925**	0.917	0.917	0.890
ArMIS (CE):	**0.548**	0.555	0.571	8.910
ArMIS (F1):	**0.773**	0.759	0.752	0.570

Table 1: Results for the ArMIS/HS-Brexit ablation studies, compared to our full models and the task organizers' baselines.

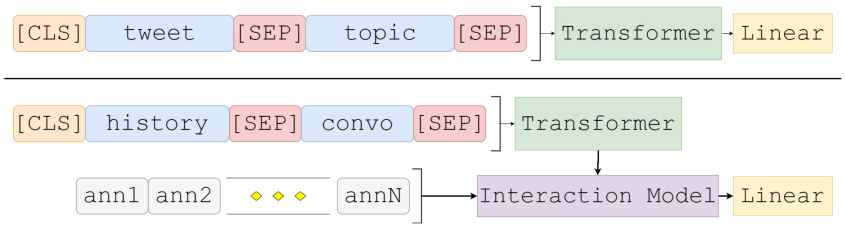

Figure 2: Overview of the data and model set-up for the MD-Agreement (top) and ConvAbuse (bottom) datasets.

Topic	Partition	Samples with Polarizability		
		0 - 0.3	0.31 - 0.5	0.51 - 1.0
Election 2020	Train	760 (0.37)	619 (0.31)	656 (0.32)
	Dev	125 (0.35)	108 (0.30)	123 (0.35)
Black Lives Matter	Train	1260 (0.54)	609 (0.26)	482 (0.20)
	Dev	174 (0.48)	98 (0.28)	87 (0.24)
Covid-19	Train	758 (0.34)	702 (0.32)	746 (0.34)
	Dev	166 (0.42)	111 (0.29)	112 (0.29)

Table 2: Topic-wise tweet polarizability counts. In parentheses are the fractions of samples in each bin.

Aspect	Variant	Evaluation Metric	
		F1-Score	Cross-Entropy
Number of encoder layers	6	0.79	0.52
	12	0.81	0.51
Topic Infusion	With	0.81	0.51
	Without	0.61	0.65

Table 3: Topical Infusion Summary

without any representation of individual annotators per se. Instead, we treated all combinations of annotators as one group and modeled their behavior based on the content of the tweet and its topic. This strategy yielded significant improvements in the F1 and cross-entropy (Table 3).

3.2 Ablation Studies

The ablation study aimed to investigate the effectiveness of incorporating topical information into large language models. Our results demonstrated that the models trained using the *Topic Infusion* technique significantly outperformed those without, as evidenced by a notable gain of 0.2 F1 score and 0.14 for the cross-entropy metric.

We conducted experiments using a transformer base model with both 12 and 6 encoder BERT-based approaches. While the 12-layer model outperformed the 6-layer BERT, the improvement in performance was not substantial enough to justify the additional complexity. These results highlight the potential of the *Topic Infusion* technique as a means of enhancing machine learning models, and suggest that the performance gains may depend on the specific model architecture used.

4 Approach 3: Annotator Interaction Modeling

The conversations in the ConvAbuse (Cercas Curry et al., 2021) dataset consisted of two parts, the history and the current turn. Somewhat mirroring our approach in *Topic Infusion*, we separate these two segments using the "[SEP]" token (refer Figure 2). Notably, we retained the markers for the conversation agent and user as we believe that these markers will enhance the expressiveness of the conversation embeddings.

We had a limited pool of 8 annotators, with each conversation having at least 3 annotators giving us confidence in our ability to reasonably model their sensitivities when it comes to detecting offensive content. We modeled how each annotator would respond to the conversation using annotator embeddings built upon a paraphrase model (Wang et al., 2020). Further, to maintain continuity we optimize the KL-divergence objective in this approach as

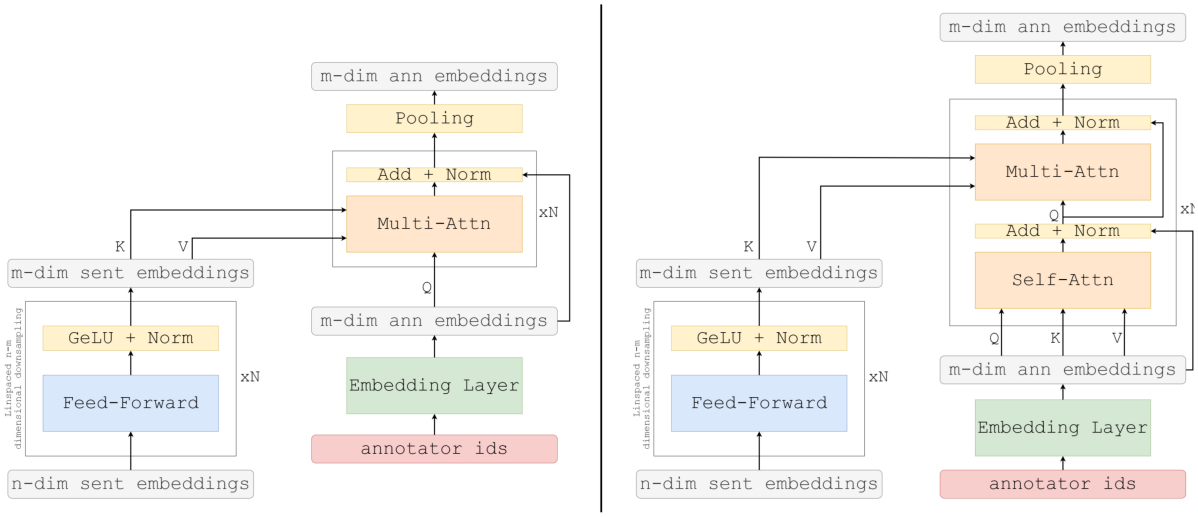

Figure 3: The interaction model with (right) and without (left) cross-consultation.

well. We detail the modeling below.

4.1 Interaction Model

The interaction model we use in this approach consists of four primary components which we summarize below and outline in Figure 3. The detailed implementations of all these components can be found in our code repository [1].

Linspaced downsampling. The conversation embedding is down sampled to the match the annotator embedding dimensions using linspaced downsampling. This N-step approach selects the intermediate dimensionalities that are evenly spaced between the annotator and conversation embedding sizes. We use $N = 2$ for this process. Although not loss-free, this incremental step down proved to be beneficial in retaining information.

Annotator embedding. During the preprocessing stage, a feature vector is created to facilitate annotator encoding. Each dimension in the vector corresponds to an annotator, enabling the representation of multiple annotators for a given document. To ensure equal length for all annotator lists and allow for attention calculations, missing annotators are assigned a value of 0.

Subsequently, the annotator IDs are passed through an embedding layer. It is helpful to conceptualize the interaction model as a conventional transformer with a vocabulary size of `num_annotators`. One significant difference is the absence of positional embeddings. In this scenario, different permutations of annotators are deemed equivalent, and thus positional information is not required.

Annotator–Conversation interactions. Here we model two types of annotator interaction with the conversation sample which we outline in Figure 3. In the first kind of interaction we imagine a jury-like system where the annotators, consult and discuss with each other before individually annotating the sample. In the second, we have the annotators independently annotating each sample (as in ConvAbuse). We label these cases as with and without *cross-consultation* respectively.

The number of attention heads in our case was decided based on the number abuse categories encountered in the dataset (i.e., N=8). The number of annotator–conversation layers was left at 1, though we can potentially increase this number in future work.

Cross-consultation. We model the annotator-sample interaction through consecutive self and cross multi-head attention layer blocks as described in (Vaswani et al., 2017). Intuitively, think of this layer as the annotators paying attention to each other's viewpoints. Subsequently we have the cross attention layer where we model how each annotator interacts with the conversation.

No cross-consultation. As with cross-consultation, we model the annotator-sample interaction through consecutive cross multi-head attention layers. The cross attention layer models how each annotator interprets a given conversation. Crucially we omit the self-attention layer as there was no inter-annotation interaction to model.

Pooling. We use mean pooling as used

[1] https://github.com/calicolab/MASDA-semeval

in Reimers and Gurevych (2019) to obtain a single Ann_{dim} dimensional embedding for the annotator group of the sample. The output of the Interaction model is of shape (`batch_size, num_annotators, h_dim`). Here, the `h_dim` is the same as that of the sentence embedding model. This embedding is then used to predict the abuse label distribution for the sample.

4.2 Ablation Studies

In this study, compared how well large language models worked when annotators were incorporated as features and in what way. Specifically, we explored the effect of varying the dimensionality of the annotator embeddings, with values of 64, 128, and 256 being tested. Our results demonstrate a substantial improvement in performance from 64 to 128 dimensions, with further increases to 256 dimensions resulting in slightly worse performance (see Table 4). These findings suggest that a higher dimensionality does not necessarily lead to improved performance, and that finding an optimal dimensionality is nontrivial.

Additionally, we evaluated the performance of different transformer models trained on various datasets. Our results indicate that the paraphrase language model exhibited the best performance due to its high percentage of conversational data, replicating its success in short text domains (Vahtola et al., 2022). These findings highlight the importance of carefully selecting the base model when developing downstream systems.

Aspect	Variant	Evaluation Metrics	
		F1-Score	Cross-Entropy
Encoder Layer Count	6	0.81	0.28
	12	0.92	0.24
MiniLM-L12-v2 Dataset	all	0.72	0.33
	paraphrase	0.92	0.24
Interaction Model	With	0.92	0.24
	Without	0.63	0.38
Annotator Embedding Dimensions	64	0.84	0.25
	128	0.92	0.24
	256	0.91	0.24

Table 4: Annotator Interaction Modeling Summary

Consistent with the results reported in section 3.2, our study provides significant evidence supporting the use of the annotator interaction model. In contrast to our previous findings, however, our current analysis reveals that increasing the encoder layer does indeed result in improved performance. Specifically, our experiments demonstrate that increasing the encoder layer from 6 to 12 in the transformer base model yields notable gains in performance, as reflected by improvements in various metrics (refer to Table 4). These findings suggest that the optimal design of the models may depend on multiple factors, including the specific techniques and architectures employed.

5 Discussion

As part of this shared task, we leveraged the ability for Transformer-based neural networks to encode annotators' perspectives somewhat independently from the "gold" label associated with a controversial text. We find that different methods of annotator representation show significant variability in their effectiveness. Ultimately, finding an approach that is as universal as possible should be the main focus of future work.

Ethics Statement

The work presented in this paper presents the same ethical challenges as the datasets over which models are built. Pre-trained language models are known to encode social biases and thus may not be sufficiently capable of capturing some differences in perspective. Because our model can be used to identify annotators who may not necessarily align with majority opinions, the outputs can be used to exclude some annotators from the final judgment, which may impact the social impact of any model trained on annotator beliefs that is released. Given the sensitive nature of the topics that were annotated (e.g., abusive language), care should be taken to avoid annotation tasks that do not protect the mental health of annotators; our method could potentially be used to flag annotators for whom this is a risk.

References

Muhammad Abdul-Mageed, Christopher Brown, and Dua'a Abu-Elhij'a. 2013. Twitter in the context of the arab spring.

Sohail Akhtar, Valerio Basile, and Viviana Patti. 2019. A new measure of polarization in the annotation of hate speech. In *AI* IA 2019–Advances in Artificial Intelligence: XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19–22, 2019, Proceedings 18*, pages 588–603. Springer.

Sohail Akhtar, Valerio Basile, and Viviana Patti. 2021. Whose opinions matter? perspective-aware models to identify opinions of hate speech victims in abusive language detection. *CoRR*, abs/2106.15896.

Dina Almanea and Massimo Poesio. 2022. ArMIS - the Arabic misogyny and sexism corpus with annotator subjective disagreements. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2282–2291, Marseille, France. European Language Resources Association.

Amanda Cercas Curry, Gavin Abercrombie, and Verena Rieser. 2021. ConvAbuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational AI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7388–7403, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Elisa Leonardelli, Stefano Menini, Alessio Palmero Aprosio, Marco Guerini, and Sara Tonelli. 2021. Agreeing to disagree: Annotating offensive language datasets with annotators' disagreement. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10528–10539, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elisa Leonardellli, Gavin Abercrombie, Dina Almanea, Valerio Basile, Tommaso Fornaciari, Barbara Plank, Massimo Poesio, Verena Rieser, and Alexandra Uma. 2023. SemEval-2023 Task 11: Learning With Disagreements (LeWiDi). In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

Filipe Rodrigues and Francisco Pereira. 2018. Deep learning from crowds. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jonathon Shlens. 2014. Notes on kullback-leibler divergence and likelihood. *arXiv preprint arXiv:1404.2000*.

Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Teemu Vahtola, Mathias Creutz, and Jörg Tiedemann. 2022. It is not easy to detect paraphrases: Analysing semantic similarity with antonyms and negation using the new SemAntoNeg benchmark. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 249–262, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.

Yijin Xiong, Yukun Feng, Hao Wu, Hidetaka Kamigaito, and Manabu Okumura. 2021. Fusing label embedding into bert: An efficient improvement for text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1743–1750.

A Appendix

In this section, we discuss the training hyperparameters and other minor details of our models.

A.1 Approach 1 - Annotator Layers

HS-Brexit

There are six annotators for HS-Brexit, and therefore six annotator layers in the model, with $d_{Ann} = 5$. For this dataset, we use the *MiniLM-L6-v2*[2] sentence transformer, which produces $d_{ST} = 384$-dimensional sentence embeddings.

This model was trained for 160 epochs with the Adam optimizer, using a learning rate of $1e-3$ for the $AnnNet_i$, *GoldNet*, and *DecNet* modules, a learning rate of $1e-5$ for the sentence transformer (to prevent catastrophic forgetting), and a weight decay of $1e-4$ across all layers.

ArMIS

We use the *CAMeLBERT-base MSA*[3] (Modern Standard Arabic) transformer, as 66.1% of Arabic tweets are in MSA (Abdul-Mageed et al., 2013). This model has several drawbacks when compared to *MiniLM-L6-v2*—in particular, *CAMeLBERT-base MSA* produces $d_{ST} = 768$-dimensional sentence embeddings, almost twice as large as *MiniLM-L6-v2*—but there are significantly fewer Arabic language models than there are for English,

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[3] https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-msa

and we were unable to find a better alternative. To compensate for the larger sentence embeddings, we use a smaller annotator latent embedding dimension ($d_{Ann} = 3$) for each of the three ArMIS annotators.

This model was trained for 25 epochs with the Adam optimizer, using a learning rate of $1e-3$ for the $AnnNet_i$, *GoldNet*, and *DecNet* modules, a learning rate of $5e-6$ for the sentence transformer, and a weight decay of $1e-3$ across all layers. As *CAMeLBERT-base MSA* is a larger model than *MiniLM-L6-v2* (and therefore requires more GPU memory), this model was trained using mini-batch stochastic gradient descent with a batch size of 64.

A.2 Approach 2 - Topic Infusion

All sentences from the MD-Agreement and ConvAbuse datasets were encoded as 384-dimensional vectors derived from the huggingface model *MiniLM-L12-v2*[4].

As part of the finetuning approach, we kept the learning rate of the transformer's parameters at $1/10^{th}$ the learning rate of the other parameters.

A.3 Approach 3 - Interaction Model

The embedding layer that processes the Annotator ID vector produces 128-dimensional annotator embeddings with dimension 12 to generate the `ann_dim` dimensional annotator embeddings.

For this model, we used the `paraphrase-MiniLM-L12-v2` [5] sentence transformer used to generate the `384 dimensional` embeddings. This approach is is identical in architecture to the one used in *Topic Infusion*. We chose this model as it was trained on more conversational data than *MiniLM-L12-v2*.

Cross-consultation model attentions are accomplished by assigning the Sentence embeddings as the Key (K) and Value (V) and the Annotator embedding as the Query (Q).

Models without cross-consultation are fed the Sentence embeddings as the Key (K) and Value (V) and the Annotator Emebdding as the Query (Q).

[4] https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

[5] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L12-v2