CICL_DMS at SemEval-2023 Task 11: Learning With Disagreements (Le-Wi-Di)

Dennis Grötzinger and **Simon Heuschkel** and **Matthias Drews**
University of Tübingen, Tübingen, Germany
{firstName.lastName}@student.uni-tuebingen.de

Abstract

In this system paper, we describe our submission for the 11th task of SemEval2023: Learning with Disagreements, or Le-Wi-Di for short. In the task, the assumption that there is a single gold label in NLP tasks such as hate speech or misogyny detection is challenged, and instead the opinions of multiple annotators are considered. The goal is instead to capture the agreements/disagreements of the annotators. For our system, we utilize the capabilities of modern large-language models as our backbone and investigate various techniques built on top, such as ensemble learning, multi-task learning, or Gaussian processes. Our final submission shows promising results and we achieve an upper-half finish.

1 Introduction

This paper is a description of the methods we used for our entry[1] on this year's SemEval Task 11, Learning with Disagreements (Le-Wi-Di) competition (Leonardellli et al., 2023). SemEval is a yearly workshop which comprises various natural language processing shared tasks. Each team that participates in one of these tasks will try to come up with systems that deepen the understanding or improve results on one kind of semantic evaluation challenge. The task we participated in was task 11, Learning with Disagreements (Le-Wi-DI). The goal of the task was to investigate how learning can be best achieved when there is no single gold label available, and instead the opinions of multiple annotators are given. For instance, consider the case of trying to classify if a sentence x contains hate speech or not. Instead of pretending that there is a single ground-truth, binary label to it that every human would agree on, in the task we were provided with a multitude of labels y for x (i.e. y = [0,1,0,0]), where each label reflects the opinion of one annotator. Thus, there were both more uncontroversial instances where most annotators agreed and also more controversial instances where annotators disagreed a lot.

Scoring was done in two ways: First, standard F1 scores were calculated ("hard evaluation"), based upon a majority voting of the annotator's opinion as labels. However, as this was exactly the assumption that the task tries to overcome, hard evaluation was only secondary and mainly used to provide a link to other research. Instead, the main evaluation criterion was "soft evaluation". Here, the cross-entropy between the predictions and the disaggregated crowd-annotations label is measured. The disaggregated crowd-annotations label is describing how many annotators labeled with 1 relative to 0. For instance, the disaggregated crowd-annotations label for y=[0,0,0,0] would be [1,0] and for y=[1,0,0,1] it would be [0.5, 0.5].

Four datasets were included in the task. Each of them had the individual annotator labels given and also the disaggregated crowd-annotations label. In that sense, they could be used as a single harmonized dataset. However, there still were some significant differences between them. Please refer to the system description paper (Leonardellli et al., 2023) or Appendix A for further description.

We have structured our system description in three main sections. First, we describe the backbone and common techniques that we use in all our approaches. This includes the use of large language models, soft loss and the general training scheme. Second, we describe our main approach that we submitted as our final result. This approach is a multi-task learning regime where we use soft and hard labels to fine-tune multiple output layers of our backbone large language model, dependent on the different tasks. Finally, we describe the alternative approaches that we tried which include the use of different loss functions, ensemble techniques, dataset balancing, Gaussian processes, an

[1] Available at https://github.com/cicl-iscl/LeWiDi_SemEval2023

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1030–1036
July 13-14, 2023 ©2023 Association for Computational Linguistics

additional dataset and statistical features.

In addition to the system description, we also report our experimental setup in Section 4 and our results in Section 5. We conclude with a summary of our approach and a discussion about possible future work.

2 Related Work

The realization that it is not sufficient for complex language tasks to only consider one ground truth and instead consider the opinion of a wide range of people with different backgrounds has become apparent at the latest with the advent of ChatGPT and Reinforcement Learning from Human Feedback (Stiennon et al., 2020).

Further approaches to learn with ambiguous data include Bakker et al. (2022) who train a reinforcement learning model to predict the preferences of individuals and thus are able to generate text that is more in line with a broad consensus. Moreover, work from Fornaciari et al. (2021) uses a multi-task learning setting to leverage information between hard labels and soft labels as uncertainty measures on ambiguous NLP tasks which they combine via KL divergence. And for the case that soft labels are not available, Zhou (2008) augment hard labels with disagreement. This approach takes a gold standard but includes the possibility of uncertainty by learning from hard labels as well as uncertainty measures.

3 System Overview

3.1 Common Techniques Across All Approaches

For all of our approaches, we used a large pretrained language model as our backbone. While we tried a few different ones that are available on huggingface transformer library (Wolf et al., 2020) in the end we converged on using the model bert-base-multilingual-cased (Devlin et al., 2018). This enabled us to use the same model for all the given datasets, as one dataset was in Arabic, which bert-base-multilingual-cased was trained on.

For many of our experiments, we did not train the main transformer at all and instead only trained the (or multiple) linear output layer. For example, in the multi-task learning approach each task is identified with a different linear output layer and in the ensemble approach, each model is just one linear layer on top of the transformer.

In addition to that, we used a soft loss for most of our experiments. In comparison to a standard hard loss, where there is only one correct class per training point, soft loss leverages the fact that we have multiple annotators. This enables the calculation of the cross-entropy between the model output and the mean of the annotations. We found the soft loss to be superior to the hard loss in almost all instances.

Furthermore, for all of our approaches except for the second step in the main approach, we joined the available datasets together into one big dataset. This also means that those models work independently of the concrete dataset, and thus might generalize better to new datasets. However, to get the best result for submission, we did do dataset-specific fine-tuning which improved the results a bit.

3.2 Main Approach

Our main approach utilizes a multi-task learning approach and a 2-step fine-tuning regime. Our objective was to try capturing disagreement between different subjective labeling tasks. We used a pretrained bert-base-multilingual-cased (Devlin et al., 2018) as our base model.

For the first step, we added two linear output layers on top of the pooled output of the raw BERT model. The first layer was trained using soft labels and the second layer was trained using hard labels. Each layer had one output neuron transformed by the sigmoid function as model output. This resulted in a combined binary cross entropy loss (BCE) for soft labels (SL) and hard labels (HL), as given by the following equation:

$$Loss = (BCE(HL) + 2 \cdot BCE(SL))/2 \quad (1)$$

Using PyTorch's automatic differentiation capabilities, this loss was used to fine-tune the BERT model on a combination of all the given train datasets.

For the second step, we took advantage of the fact that we had access to the four datasets individually and not just their aggregate. As each dataset is associated with a (slightly) different task (hate speech, misogyny, offensiveness and abusiveness) we therefore can determine for every prediction what the related task is. Thus, we modeled each task with 2 heads (one for soft labels and one for hard labels) each on top of our BERT model, resulting in eight heads in total. For any new input x, we first determined from which dataset it came from and then propagated it to its respective heads. Therefore, the individual heads could "focus" on the relevant information of their associated tasks and datasets.

Each head was a series of three linear layers, combined via tanh non-linear activation functions and also including dropout. During training, we trained the heads on their respective dataset but kept the parameters of the underlying BERT model frozen. As we had to predict both hard labels and soft labels for the evaluation on the leaderboard, we used the respective hard label heads to calculate the hard labels and the respective soft label heads to calculate the soft label. The architecture enables to check disagreement for a given text across different subjective labeling tasks. For example, a text could have high disagreement in sexism, but lower disagreement in offensive language. The results still operate on a common ground of disagreement, in form of the frozen BERT model.

3.3 Alternative Approaches

In this section, we describe the various other approaches that we tried, which include ensemble techniques, different loss functions, dataset balancing, Gaussian processes, additional datasets and statistical features. All of those approaches did not improve our results compared to our main approach. However, as all of our approaches were based on the same large language model backbone, the results were also not far from our main approach, mostly within a 0.1 difference in terms of cross-entropy.

3.3.1 Loss Functions

For most of our experiments, we used a cross-entropy loss function as this loss function is both widely successful and also seems to be the natural go-to for our task, as a low cross-entropy error is exactly what we want to achieve in our task. However, we thought it worthwhile to explore other options for the loss function.

One of the options we tried out was the L1 loss, which measures the mean absolute error between output and target. L1 loss is known to minimize the expected misclassification probability instead of maximizing the fully correct labeling (Janocha and Czarnecki, 2017). This could potentially be useful for our tasks, as the goal is not to get every single label right, but rather to capture the aggregated labels which includes not being sensitive to outliers. However, we found no improvements to our results using L1-loss.

Another approach we considered fruitful was the use of Wasserstein loss. This loss is frequently used in generative adversarial networks, as it prevents mode collapse and in general is a more robust measure of dissimilarity between two distributions compared to cross-entropy (Arjovsky et al., 2017; Frogner et al., 2015). However, we also found no evidence that the Wasserstein loss improved our results.

3.3.2 Ensembles

Ensemble methods are a set of powerful techniques used throughout the field of machine learning and beyond (Sagi and Rokach, 2018). They work by combining the output of multiple models into one single, unified model which has been shown to be invaluable for many tasks (Dietterich, 2000). Ensemble methods also seem to be a natural fit for our problem, as the task involves predicting a combination of individual annotators, which could in theory be conveniently modeled by ensembles.

For our approach, we constructed the ensembles by adding linear layers on top of the transformer language models we were using, typically multilingual-bert. We then trained those linear layers individually and subsequently combined the trained linear layers with one of several approaches.

The training was done in one of two ways. The first approach was a standard training procedure, using the soft loss criteria for gradient calculation. This is the same approach that we used for our submitted system, just that here the underlying transformer model is not trained at all and instead just the linear layers are used. Our second training approach was to leverage the fact that we had access to individual annotators, by training each linear layer on one annotator id. This had two effects: First, it ensured that each of the linear layers received a different loss signal, making the linear layers more diverse. Second, in theory, this could also be used to simulate one annotator with one linear layer and thus recover the aggregated labels when combining the linear layers. However, as we wanted to have one approach that fits all datasets, this was difficult to implement due to the fact that the datasets had varying annotation schemes, some sticking to the same annotators for the whole dataset while others had different annotators for each instance.

The trained linear layers were then combined in one of three ways. Firstly, we simply took the average of the linear layer predictions. Secondly, we used an additional linear layer to combine the outputs of all other linear layers. This additional linear layer was trained as well. Lastly, we also used an additional linear layer, but this time also included the standard deviation of all linear layers as an input feature. The reasoning behind this was that if the linear layers were to truly capture individual annotators, their disagreement would be reflected in the standard deviation between the predictions.

However, none of the six training and combination pairs above improved our overall performance on the datasets compared to our main approach. Still, we think that this approach could be a worthwhile endeavor to explore further, as it seems to be a natural way to model the task.

3.3.3 Dataset Balancing

As we noticed that the datasets were unbalanced, we investigated the impact of balancing techniques. We tried both a combination of under- and oversampling as well as using a weighted cross-entropy loss. However, we found no evidence that balancing helped performance on the test set.

3.3.4 Gaussian Process

Gaussian processes can be a very powerful tool in statistical modeling (Schulz et al., 2018). They work by defining a probability distribution over functions and using tools from probability theory to find the posterior distribution that best fits a given set of data. For our approach, we used the pooled output features of our transformer model as the data. In this sense, every text input was transformed into a vector representation which contains a useful signal for the Gaussian process to work with. We then trained the Gaussian model on this dataset. However, this approach did not improve results beyond our main approach.

3.3.5 Additional Dataset (CoRoSeOf)

Another approach was the addition of a further, large data set of annotated tweets in the domain of offensive language detection. The intention for this was that using this data in the pre-training process would improve our results, even if it would be in a different language. Our choice for this was the CoRoSeOf (Hoefels et al., 2022) dataset, which consists of almost 40,000 Romanian tweets, which were annotated for sexism and offensive language by three annotators.

To use the data, we then had to adapt it to the shared format of the other Le-Wi-Di sets. This included the addition of soft labels, calculated from the annotations, restructuring some columns and simplifying the annotations.

Since the annotations were not in binary format, but rather out of a choice of six answers with more than one domain per answer (Non sexist, Non sexist offensive, Sexist direct, Sexist descriptive, Sexist reporting and Cannot decide), we decided to focus on only sexist annotations. The first two options were converted to a binary 0, the three 'sexist' options to '1', and 'cannot decide' to 'None'. To clarify the results, all entries with 'Cannot decide' were removed, which still left us with around 37000 entries. The distribution of hard labels was quite similar to the other four sets, but the size was larger than the rest of them combined. Using this additional dataset showed improvement for some of our experiments, but in the end we achieved our best result without the dataset.

The different language did not show a problem, which due to the already multilingual datasets was to be expected, although additional care needed to be placed on not introducing too much bias due to the sheer size, which is larger than the other sets combined.

3.3.6 Statistical Features

To further improve results, we also tried to add various statistics as features, including text length, punctuation, word types/frequency and more. This was in part done through CTAP (Chen and Meurers, 2016), an online tool for corpus analysis. The biggest problem, and ultimately the reason for the exclusion in our system was the insufficient length of each text, due to the character limitations imposed by Twitter. The other statistical features did not change the results and were thus also omitted.

4 Experimental Setup

We used the transformer library of Hugging Face (Wolf et al., 2020) to download our pre-trained models. Furthermore, we used PyTorch (Paszke et al., 2017) for all of our code. For experiment tracking, we used Weights & Biases (Biewald, 2020). Finally, for the Gaussian process implementation, we used (Gardner et al., 2018).

For the common ground model of our submitted approach, we calculated the mean cross-entropy score across the whole dev dataset. For the submitted subjective task model, the cross entropy for

Scores	Average	HS-Brexit dataset	ArMis dataset	Conversation Abuse dataset	MD-Agreement dataset
CE	0.43 (9)	0.33 (11)	0.61 (12)	0.23 (8)	0.53 (10)
F1	0.83 (13)	0.89 (8)	0.70 (11)	0.93 (8)	0.80 (15)

Table 1: Our results in both scores on average, and across the subdomains. Bracketed numbers are the relative positions out of 27 participants (including baseline).

a specific dataset was used to evaluate the corresponding heads. The models were trained on the whole train dataset and evaluated on the dev dataset. The training process was stopped if the epoch cross-entropy error on the dev dataset has risen two epochs in a row (early stopping). For submission and further use, the model weights for the epoch with the lowest cross-entropy on the unseen dev dataset were used. Further information on hyperparameter tuning and training setup can be found in Appendix B or our code on Github[2].

5 Results

Our main submission achieved an average cross-entropy score of 0.43 and an average F1 (micro) score of 0.83 overall for four evaluation phase datasets. This is compared to a baseline average cross-entropy score of 5.62 and a baseline average F1 (micro) score of 0.74 and the best result with a cross-entropy score of 0.35 and a baseline average F1 (micro) score of 0.87. With those results, we achieved place 13 out of 27 on the evaluation leaderboard. A detailed description of our results can be seen in Table 1.

For our main approach, the common ground model achieved a mean cross-entropy on the dev datasets of 0.428, whereas the finally submitted model had a score on the dev datasets of 0.424. The second fine-tuning step of the multi-task learning setup did not seem to benefit the model a lot.

All other approaches that we tried either had the same or were very slightly below the score of our main submission, though mostly in the range of 0.1 difference in terms of cross-entropy score.

6 Conclusion

In this paper, we presented our submission for SemEval-2023, Task 11: Learning With Disagreements (Le-Wi-Di). Our main approach, which combines a pre-trained large language model with multi-task learning shows promising results for capturing disagreement as described in the task. Furthermore, extensive investigation into other techniques, such as ensembles, Gaussian processes and different loss functions provides a basis to more accurately determine which approaches are sensible or not to capture disagreement in the future.

One way we think our approach can be improved in the future is to include more data, for instance by developing methods to collect annotations akin to the methods introduced in (Ethayarajh et al., 2023). Furthermore, we think it could be worthwhile to investigate the use of different uncertainty measures, such as Krippendorffs Alpha which measures annotator disagreement for a dataset. Finally, as all of our results are based on the embeddings of large language models, we certainly think that using bigger and more capable models will also improve results.

Dealing with uncertainty in the learning process with soft labels, it would be interesting to integrate an uncertainty measure in the training process.

7 Acknowledgments

We would like to thank the organizers of SemEval and this subtask for their work in setting up and executing this task, Dr. Çagri Çöltekin for his guidance and feedback and Diana Höfels for the usage of the CoRoSeOf dataset.

References

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.

Dina Almanea and Massimo Poesio. 2022. ArMIS - the Arabic misogyny and sexism corpus with annotator subjective disagreements. In *Proceedings of*

[2] Available at https://github.com/cicl-iscl/LeWiDi_SemEval2023

the Thirteenth Language Resources and Evaluation Conference, pages 2282–2291, Marseille, France. European Language Resources Association.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan.

Michiel A. Bakker, Martin J. Chadwick, Hannah R. Sheahan, Michael Henry Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matthew M. Botvinick, and Christopher Summerfield. 2022. Fine-tuning language models to find agreement among humans with diverse preferences.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

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.

Xiaobin Chen and Detmar Meurers. 2016. CTAP: A web-based tool supporting automatic complexity analysis. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 113–119, Osaka, Japan. The COLING 2016 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*, pages 1–15. Springer.

Kawin Ethayarajh, Heidi Zhang, Yizhong Wang, and Dan Jurafsky. 2023. Stanford human preferences dataset.

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.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. 2015. Learning with a wasserstein loss.

Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. 2018. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*.

Diana Constantina Hoefels, Çağrı Çöltekin, and Irina Diana Mădroane. 2022. CoRoSeOf - an annotated corpus of Romanian sexist and offensive tweets. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2269–2281, Marseille, France. European Language Resources Association.

Katarzyna Janocha and Wojciech Marian Czarnecki. 2017. On loss functions for deep neural networks in classification. *CoRR*, abs/1702.05659.

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*, Toronto, Canada. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249.

Eric Schulz, Maarten Speekenbrink, and Andreas Krause. 2018. A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, 85:1–16.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

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, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 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, Online. Association for Computational Linguistics.

Zhi-Hua Zhou. 2008. Semi-supervised learning by disagreement. In *2008 IEEE International Conference on Granular Computing*, pages 93–93.

A Datasets

Le-Wi-Di consists of four separate datasets, each of them dealing with annotation agreement/disagreement in various topics. All of the sets have been standardized by the task's organizers, with additional information to be found in some sets.

1. The first set, HS-Brexit (Akhtar et al., 2021), deals with Hate speech detection in various tweets surrounding the topic of Brexit. Six annotators, three of them British Muslim Immigrants and the rest unspecified other individuals checked whether a tweet consisted of hate speech, with an additional measure of aggressive and offensive language. All of the annotators judged all of the tweets.

2. The second dataset, ArMis (Almanea and Poesio, 2022), judged Arabic tweets on misogyny and sexism. Three annotators, identified as a liberal female, moderate female and conservative male volunteer, judged all of the tweets on the aforementioned topics. This is the only dataset in a language other than English.

3. Multi-Domain Agreement (Leonardelli et al., 2021) contained three different topics (or domains), namely Black Lives Matter, the US Election of 2020 and Covid19. From a large pool of potential annotators, five were chosen at random for each tweet, with no further description of them. The task however was always the same: Detect offensiveness on a binary scale.

4. The last dataset, ConvAbuse (Cercas Curry et al., 2021) is the only one not consisting of tweets, but rather conversations of users with chatbots, namely CarbonBot and E.L.I.Z.A. Three or more annotators try to detect abusiveness directed at the bots by the users, which in contrast to the other sets is not done on a binary but rather a Likert Scale. Other annotations, such as the type and direction of abuse were also noted.

Although the annotations themselves are not uniform, there are some standardized fields. The number of annotators is given, as well as hard and soft labels. Hard labels show the majority of reports on a binary scale of 0 and 1, while soft labels show the distribution of 0 and 1 across all of the entries.

B Hyperparameters

For the submitted approach we have two sets of hyperparameter. One for the common ground base model and one for the submitted subjective task model. We used the random seed 14 for both models. Furthermore, we trained both models with pytorch's AdamW optimizer. The tokenizer for the model was used with a maximal length of 240 including special tokens. The dataset set for training and evaluation was 64.

Differences between the models was mainly due to the different training sizes and corresponded in different number of epochs, learning rates and schedules. For the common ground model training was scheduled on seven epochs with a learning rate of 5e-05 with a cosine schedule including ten percent of all steps as warmup. For the subjective task model we only fine tuned on the different dataset, leading to a great difference in the amount of data per head. Scheduling training for one hundred epochs and restarting the learning rate rate decay multiple times should account for the differences. The learning rate was a bit smaller with 1e-05 and only five percent of all steps as warmup. In the hundred epochs the schedule restarted ten times. As training was interrupted if the cross entropy error is raising, heads with less data trained shorter then with more data.