Deep-change at CoMeDi: the Cross-Entropy Loss is not All You Need

Mikhail Kuklin
Moscow State University, Russia
Yandex, Russia
kuklin.mike@yandex.ru

Nikolay Arefyev
University of Oslo, Norway
nikolare@uio.no

Abstract

Manual annotation of edges in Diachronic Word Usage Graphs is a critical step in creation of datasets for Lexical Semantic Change Detection tasks, but a very labour-intensive one. Annotators estimate if two senses of an ambiguous word expressed in two usages of this word are related and how. This is a variation of the Word-in-Context (WiC) task with some peculiarities, including diachronic data, an ordinal scale for annotations consisting of 4 values with pre-defined meanings (e.g. homonymy, polysemy), and special attention to the degree of disagreement between annotators which affects the further processing of the graph. CoMeDi is a shared task aiming at automating this annotation process. Participants are asked to predict the median annotation for a pair of usages in the first subtask, and estimate the disagreement between annotators in the second subtask. Together this gives some idea about the distribution of annotations we can get from humans for a given pair of usages.

For the first subtask we tried several ways of adapting a binary WiC model to this 4 class problem. We discovered that further fine-tuning the model as a 4 class classifier on the training data of the shared task works significantly worse than thresholding the original binary model. For the second subtask our best results were achieved by building a model that predicts the whole multinomial distribution of annotations and calculating the disagreement from this distribution. Our solutions for both subtasks have outperformed all other participants of the shared task.

1 Introduction

Diachronic Word Usage Graphs (DWUGs) (Schlechtweg et al., 2021) have recently become a de-facto standard data structure when working on Lexical Semantic Change Detection (LSCD) tasks (Schlechtweg, 2023). A graph is built for a particular ambiguous lemma. Graph nodes correspond to usages of this lemma from an older or a newer corpus. Edges are annotated with human judgements about relatedness of senses of the target lemma in the two corresponding usages. The annotations are integer values from 1 to 4, where 1 means completely unrelated senses and 4 the same sense. Based on these annotations a number of automated procedures can be applied to the graph, including filtering noisy and ambiguous usages based on disagreement between annotators, inferring senses of the target lemma, discovering novel or lost senses of the lemma. However, to get reasonable results from these procedures an abundant amount of high quality annotations is required. Given that the number of edges grows quadratically with the number of usages this annotation task is especially resource-consuming.

CoMeDi (Schlechtweg et al., 2025) is a shared task calling for automating this manual annotation process. It relies on DWUG datasets that had been previously created for Russian (Rodina and Kutuzov, 2020; Kutuzov and Pivovarova, 2021; Aksenova et al., 2022), Chinese (Chen et al., 2023), Spanish (Zamora-Reina et al., 2022), Norwegian (Kutuzov et al., 2022), German, Swedish and English (Schlechtweg et al., 2024; Kurtyigit et al., 2021; Hätty et al., 2019; Schlechtweg et al., 2018). It consists of two subtasks, the first requires predicting the median of human annotations for a pair of usages and the second aims at estimating disagreement between annotators on this pair. We propose several solutions for each subtask.

Our solutions for the first subtask are based on an existing binary WiC model. One approach to adapting it to the subtask is further fine-tuning for the 4 class classification problem on the training data of the shared task. Another approach is taking the predicted probability of the positive class (i.e. that the sense is the same in two usages) from the original binary model and converting it to the 4 point

Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation, pages 48–64
January 19, 2025.

scale by thresholding. These thresholds can be selected to directly maximize the evaluation metric of the subtask. Surprisingly, the second approach gives much better results. Binarising CoMeDi training data and further fine-tuning of the binary WiC model on it gives additional performance gains. For the second subtask we trained models to predict the measure of disagreement directly or predict the whole distribution of annotations from which the measure of disagreement can be calculated. The second approach has shown better results.

Our best solutions demonstrated the highest performance among all participants of the shared task during the evaluation period. In the post-evaluation period we improved the results and systematically studied various design options.

2 Related work

Predicting if two occurrences of the same ambiguous word have similar or different senses is known as the Word-in-Context (WiC) task. Most often it is framed as a binary classification task (Pilehvar and Camacho-Collados, 2019; Martelli et al., 2021). A graded version of this task was also considered before, e.g. in SemEval-2020 Task 3 (Armendariz et al., 2020), with the Spearman's and Pearson's correlations between model and human judgements serving as evaluation metrics. In the CoMeDi shared task Krippendorff's α is used as a metric and models are required to return exactly the same annotations as humans, not just some correlated predictions.

Many WiC models exist, but in the recent shared tasks on LSCD the SOTA / near-SOTA results were obtained by systems relying on XL-LEXEME (Cassotti et al., 2023) and DeepMistake (Arefyev et al., 2021). Since our solutions of the CoMeDi shared task employ the DeepMistake model, we will describe it focusing on those details that are important for understanding our solutions. DeepMistake was originally developed as a solution for the Multilingual and Cross-Lingual WiC (MCL-WiC) task (Davletov et al., 2021), and then further improved and adapted for two LSCD shared tasks in Russian (Arefyev et al., 2021) and Spanish (Homskiy and Arefyev, 2022). It consists of an XLM-R (Conneau et al., 2019) based backbone, which encodes two input usages concatenated together. For each occurrence of the target word an embedding is calculated by mean-pooling XLM-R outputs for subwords of this occurrence. Then a target aggregation function combines the embeddings of two occurrences of the target word into a single representation, which is fed to a classification head. Extensive experiments with various target aggregation functions were carried out. Among 10 aggregation functions explored in Davletov et al. (2021) the best function was *comb_dmn*, which is the concatenation of the component-wise difference of unnormalized and the component-wise product of normalized embeddings: $comb\_dmn(x, y) = (x - y, \overline{x} \odot \overline{y})$. In Arefyev et al. (2021) a function *l1ndotn* concatenating the Manhattan distance and the dot product of normalized embeddings was proposed, which proved to work better at least for LSCD: $l1ndotn(x, y) = (||\overline{x} - \overline{y}||_1, \overline{x} \cdot \overline{y})$. DeepMistake was originally initialized with XLM-R weights and fine-tuned on training, development and trial data from MCL-WiC. The combined train set consists of usages in English, Russian, French, Arabic and Chinese, and also a few cross-lingual pairs. For the shared tasks on LSCD it was further fine-tuned on the data in Russian and Spanish from these tasks.

3 Subtask 1: Median Judgment Classification

3.1 Task description

In this subtask participants are provided with pairs of word usages. Each pair has several human judgments on an ordinal scale from 1 to 4. The task is to predict the median of these judgments for each usage pair. The evaluation is performed using the ordinal version of Krippendorff's α (Krippendorff, 2018), which accounts for the degree of deviation between the predicted and true median values.

3.2 Models

In this section we introduce our solutions for the median judgment classification subtask. All of them employ the WiC model DeepMistake (Davletov et al., 2021; Arefyev et al., 2021). The original DeepMistake model is a binary classifier predicting if two usages of the same word have the same sense. This model can be used directly and predict 2 out of 4 classes, or the predicted probability of the positive class can be quantized into 4 intervals to get a 4-class classifier. To better adapt DeepMistake to the shared task we further fine-tune it as a binary classifier on the CoMeDi training data. Additionally, we experiment with replacing the classification head and fine-tuning the model as a 4-class classifier.

Model/Participant	Krippendorff's α							
	ZH	**DE**	**EN**	**NO**	**RU**	**ES**	**SV**	**AVG**
2class@CoMeDi-ZH	**0.424**	0.723	**0.732**	0.633	**0.633**	**0.748**	**0.675**	0.652
Mixed	**0.424**	0.723	**0.732**	**0.668**	0.623	**0.748**	**0.675**	**0.656**
comedy_baseline_2	0.379	**0.728**	0.654	0.515	0.550	0.656	0.601	0.583
daalft	0.317	0.656	0.555	0.589	0.487	0.636	0.648	0.555
NBTailee	0.362	0.672	0.574	0.438	0.420	0.595	0.608	0.524
JuniperLiu	0.140	0.492	0.507	0.080	0.128	0.330	0.224	0.271
comedi_baseline	0.059	0.274	0.102	0.124	0.112	0.175	0.018	0.123

Table 1: Evaluation results on the subtask 1. Best results for each language are in **bold font**.

3.2.1 DeepMistake-based models

Most of our experiments employ the **MCL→DWUG$_{es}$+XLWSD$_{es}$** version of DeepMistake, which is the best model from (Homskiy and Arefyev, 2022). It was initialized from XLM-R (Conneau et al., 2019) and underwent the two-stage fine-tuning process. Initially, it was fine-tuned on the multilingual MCL-WiC dataset (Martelli et al., 2021), followed by a combination of the Spanish DWUG (Zamora-Reina et al., 2022) and the Spanish subset of XLWSD (Pasini et al., 2021). This model employs the l1ndotn aggregation function and we further adapt it to the shared task by fine-tuning it on the CoMeDi train sets.

All models were fine-tuned for 50 epochs using AdamW with a linear learning rate scheduler, lr=1e-05 and early stopping by the average Krippendorff's α across all languages (except for 2class@CoMeDi-ZH+byNO, see below).

2class@CoMeDi. This model variant was fine-tuned for binary classification on the concatenation of all CoMeDi train sets employing the binary cross-entropy (BCE) loss. Here, examples with the median annotations of 1 and 2 were employed as examples of the negative class, while examples with the median of 3 and 4 as examples of the positive class.

2class@CoMeDi-2,3. This model is identical to the previous one, but examples with the median annotations of 2 and 3 were excluded from its train set. We hypothesized that training only on the clear-cut examples having most annotations of 1 or 4 will improve model performance.

2class@CoMeDi-ZH. Identical to 2class@CoMeDi, but examples in Chinese were removed from the train set. This was based on our preliminary experiments where we observed that fine-tuning on the Chinese train set only results in the worst performance on all dev sets, including the Chinese one (see Appendix A).

2class@CoMeDi-ZH-DE. For this model, both Chinese and German[1] examples were removed from the train set.

2class@CoMeDi-ZH+byNO. We observed that Norwegian is the only language for which the results on the dev set can be significantly improved if early stopping is done by the Krippendorff's α on this specific language as opposed to the average across all languages. This model was fine-tuned similarly to 2class@CoMeDi-ZH, but the checkpoint with the best dev performance on Norwegian was selected.

4class@CoMeDi. This version was fine-tuned for 4-class classification on the CoMeDi dataset. It utilized the cross-entropy (CE) loss where the target was the median annotation for a pair of usages.

4class@CoMeDi-ZH. Identical to the previous model, but examples in Chinese were removed from the train set.

3.2.2 NMthres

To adapt a model trained for binary classification to predict four classes, thresholding can be applied to the predicted probability of the positive class. This method is taken from the baseline of the shared task, we will refer to it as **NMthres**. For each language separately, NMthres learns 3 thresholds that discretize a continuous input variable into 4 classes by optimizing the target metric using the Nelder-Mead method (Nelder and Mead, 1965). NMthres can be applied to the probability of the positive class predicted by any binary DeepMistake model.

3.2.3 Inference methods

For inference different strategies are applied. For all 4-class DeepMistake models the class with the highest probability is selected directly. In contrast, for the 2-class models without NMthres either class

[1]We selected German as the second candidate for exclusion because of the poor accuracy of a model trained on German on other dev sets, see Appendix A.

1 or 4 is chosen based on the threshold of 0.5. Otherwise, the predicted class is selected by NMthres.

3.3 Evaluation results

During the evaluation phase we submitted two sets of predictions. The first submission consists of predictions from the 2class@CoMeDi-ZH model as it achieved the highest Krippendorff's α on the development set. The second submission titled **Mixed** was constructed using predictions from multiple models: for Norwegian we used predictions from the 2class@CoMeDi-ZH+byNO model, for Russian we employed 2class@CoMeDi, and for other languages we utilized 2class@CoMeDi-ZH. For both submissions NMthres was optimized on the CoMeDi dev set and applied to the predicted probabilities of the positive class from the DeepMistake models. The results on the test set are presented in Table 1.

During the evaluation phase, both submissions proved to outperform all other participants on average across languages. We also have achieved the best results on all individual languages except for German where comedy_baseline_2 secured the top position. Our second submission was a bit better than the first one on Norwegian, but worse on Russian.

3.4 Post-evaluation experiments

3.4.1 Train-test overlap

After the evaluation phase it was revealed that the Spanish portion of the CoMeDi test set is derived from the Spanish DWUG dataset (Zamora-Reina et al., 2022) which was partially used to fine-tune the MCL→DWUG$_{es}$+XLWSD$_{es}$ model. Table 2 shows the overlap between the training data for this model and the Spanish test set.

Due to the significant overlap, we aimed to assess its impact on the final Krippendorff's α on the test set. We compared four DeepMistake models from Homskiy and Arefyev (2022): (1) *MCL→DWUG$_{es}$+XLWSD$_{es}$*, (2) *MCL* trained solely on the MCL dataset with no overlap with the CoMeDi test set, (3) *MCL→RSS* fine-tuned on the RuSemShift (RSS) (Rodina and Kutuzov, 2020), and (4) *MCL+RSS+DWUG$_{es}$+XLWSD$_{es}$* fine-tuned on all datasets simultaneously. Although the models trained on RuSemShift also show some overlap with the training set, the MCL model exhibits no overlap at the usage level, as indicated in Table 2.

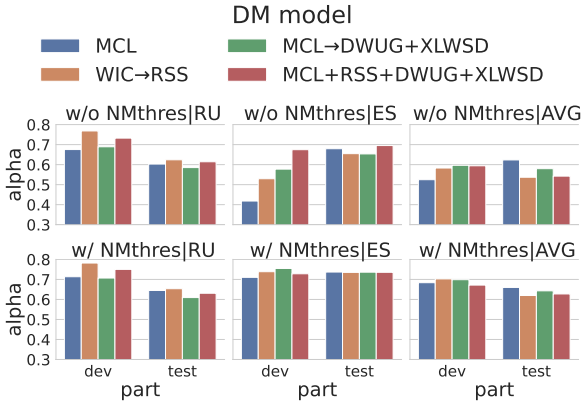

Figure 1: Krippendorff's α of DeepMistake models w/ and w/o NMthres. Results are on Russian and Spanish sets, and on average across all languages. See extended plot in Figure 7.

The results of this comparison are shown in Figure 1. DeepMistake models fine-tuned on RSS are clearly better for Russian. For Spanish the results are mixed, on the test set all models are on par when using NMThres. On average across languages, among models that are not fine-tuned on CoMeDi the best test result of 0.660 are achieved by the MCL model (no overlap), this model outperforms both of our submissions and other participants as well.

Inspired by improvements of the 2class@CoMeDi-ZH model over the non-fine-tuned version, we similarity fine-tuned the MCL model on CoMeDi data. We found that with fine-tuning on CoMeDi the results of the MCL model are worse, but still better than other participants, see table 3.

3.4.2 Optimizing CoMeDi Training Data for Fine-Tuning DeepMistake Models

In this subsection, we explore which subsets of CoMeDi training data should be utilized for fine-tuning the DeepMistake models to enhance Krippendorff's α. We conducted experiments by removing examples with the median annotation equal to 2 or 3, excluding Chinese examples, and excluding both Chinese and German language examples. The outcomes of these experiments are depicted in Figure 2.

Our findings indicate that for improved performance on the Chinese development and test sets of CoMeDi, it is beneficial to exclude Chinese data during training (see appendix C for a more in-depth analysis). Furthermore, removing the German examples from the training data does not significantly

Language	Part	MCL	DWUG$_{es}$ + XLWSD$_{es}$	RSS
Spanish	dev	-	**3/112/175 (30/31/28 %)**	-
	test	-	**4/112/155 (20/15/10 %)**	-
Russian	dev	3/0/0 (10/0/0 %)	-	8/363/180 (29/16/16 %)
	test	3/0/0 (5/0/0 %)	-	11/487/244 (20/11/11 %)

Table 2: Overlap between the CoMeDi evaluation data and training data of DeepMistake models. Both the absolute counts of lemmas / usages / usage pairs in common and the proportions of test items present in the training set (in brackets) are reported. During the evaluation phase we employed the model trained on MCL and DWUD$_{es}$+XLWSD$_{es}$. Its training data overlaps with the CoMeDi test data for Spanish (in **bold**). It also has 3 common lemmas but no common usages with the test data for Russian. In the post-evaluation experiments we additionally experimented with a model trained on MCL only to avoid overlaps on the level of usages and usage pairs, as well as models trained on RSS which overlaps with the test data for Russian. MCL also contains examples in English and Chinese, but we found no overlaps with the corresponding evaluation sets.

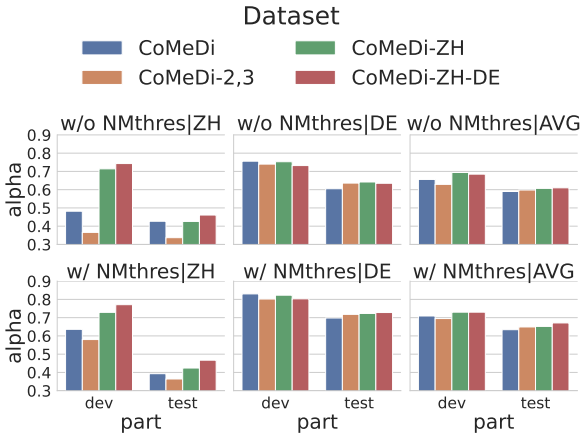

Figure 2: Krippendorff's α of 2-class DeepMistake models fine-tuned on different subsets of CoMeDi train data. The results on the Chinese and the German dev sets, and on average across all dev sets are shown. See extended plot in Figure 8.

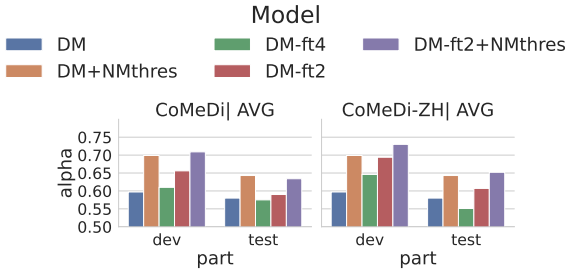

Figure 3: Average Krippendorff's α across all languages for models fine-tuned on CoMeDi and CoMeDi-ZH. DM stands for DeepMistake with the MCL→DWUG$_{es}$+XLWSD$_{es}$ weights. ft-2 and ft-4 stand for 2-class and 4-class fine-tuning respectively. See extended plot in Figure 9.

affect performance on the German test set or the overall Krippendorff's α, but it does lead to better results on the Chinese subset. Conversely, removing examples with median annotations of 2 and 3 results in poorer performance on the Chinese set and reduces the average performance on the development set, although there is a slight improvement on the test set.

3.4.3 Evaluating Fine-Tuning Strategies on CoMeDi Training Sets

In this analysis, we evaluated various training strategies for fine-tuning DeepMistake models, as depicted in Figure 3.

Our investigation suggests that training with a 4-class cross-entropy (CE) approach (DM-ft4) is suboptimal. While the Krippendorff's α on the development set shows a slight improvement, the performance on the test set declines compared to the over original DeepMistake which was not fine-tuned on CoMeDi data (DM). In contrast, fine-tuning the 2-class DeepMistake model (DM-ft2) yields noticeable improvements over the DM on the development set, albeit with only modest gains on the test set. These test set improvements do not surpass the results obtained by simply applying the initial DeepMistake model with NMthres (DM+NMthres).

Overall, fine-tuning DeepMistake as a binary classifier on CoMeDi training data and then applying NMthres to obtain a 4-class classifier delivers the best results. Comprehensive post-evaluation results are provided in Table 3, and a more detailed comparison is provided in Appendix B.

4 Subtask 2: Mean Disagreement Ranking

4.1 Task description

Similarly to subtask 1, participants are given pairs of word usages. The objective is to predict the mean absolute difference between judgments of

Model	Krippendorff's α ZH	DE	EN	NO	RU	ES	SV	AVG
DM w/o NMthres								
MCL	*0.453*	*0.675*	*0.624*	*0.660*	*0.603*	*0.679*	*0.669*	*0.623*
MCL→DWUG$_{es}$+XLWSD$_{es}$	0.408	0.689	0.520	0.575	0.585	0.653	0.628	0.580
DM w/ NMthres$_{dev}$								
MCL	*0.465*	***0.741***	*0.727*	***0.688***	***0.645***	*0.737*	*0.617*	*0.660*
MCL→DWUG$_{es}$+XLWSD$_{es}$	0.423	0.738	0.710	0.680	0.609	0.736	0.604	0.643
2class w/o NMthres								
2class@CoMeDi	0.427	0.605	0.640	0.642	0.555	0.658	0.604	0.590
2class@CoMeDi-2,3	0.337	0.636	0.637	0.631	0.578	0.683	0.686	0.598
2class@CoMeDi-ZH	0.426	0.642	0.626	0.609	0.599	0.669	0.680	0.607
2class@CoMeDi-ZH+byNO	0.340	0.524	0.589	0.649	0.442	0.536	0.548	0.527
2class@CoMeDi-ZH-DE	0.461	0.635	0.644	0.631	0.554	0.677	0.671	0.610
MCL,2class@CoMeDi-ZH	*0.417*	*0.592*	*0.626*	*0.639*	*0.543*	*0.605*	*0.544*	*0.567*
2class w/ NMthres$_{dev}$								
2class@CoMeDi	0.393	0.698	0.712	0.649	0.623[2]	0.735	0.633	0.634
2class@CoMeDi-2,3	0.364	0.718	0.748	0.664	0.630	0.719	**0.699**	0.649
2class@CoMeDi-ZH	0.424[1,2]	0.723[1,2]	0.732[1,2]	0.633[1]	0.633[1]	0.748[1,2]	0.675[1,2]	0.652
2class@CoMeDi-ZH+byNO	0.436	0.620	0.637	0.668[2]	0.547	0.591	0.597	0.585
2class@CoMeDi-ZH-DE	0.467	0.728	**0.758**	0.662	0.629	**0.773**	0.679	**0.671**
MCL,2class@CoMeDi-ZH	*0.392*	*0.692*	*0.733*	*0.642*	*0.619*	*0.728*	*0.637*	*0.635*
4class								
4class@CoMeDi	**0.517**	0.665	0.514	0.609	0.532	0.583	0.602	0.575
4class@CoMeDi-ZH	0.393	0.643	0.516	0.559	0.526	0.627	0.592	0.551

Table 3: Post-evaluation results on the test set of subtask 1. Best results for each language are in **bold font**. Superscripts refer to our two submissions during the evaluation phase. By default, fine-tuned models are based on MCL→DWUG$_{es}$+XLWSD$_{es}$, unless specified otherwise. Models based on MCL (no overlap with CoMeDi test data) and their results are in *italic*.

different annotators for a given pair of usages:

$$D(J) = \frac{1}{|J|} \sum_{(j_1,j_2)\in J} |j_1 - j_2| \qquad (1)$$

J in Equation 1 is the set of pairs of judgments for the same usage pair.

The evaluation metric is Spearman's ρ (Spearman, 1904) between the predicted and the real mean disagreements between annotators for a set of usage pairs.

4.2 Models

In this section, we describe various approaches for modelling annotator disagreement using the DeepMistake model. Our initial strategy focused on a regression model designed to directly predict the mean disagreement between annotators, leveraging the mean squared error (MSE) loss function. To address difficulties in learning from noisy regression target values, we introduced a binary classification variant, which aims to identify usage pairs where all annotators provided the same answer, applying the binary cross-entropy (BCE) loss for learning. Furthermore, we experimented with a model that predicts the entire distribution of annotations and calculates disagreement from this distribution. It is trained on individual annotations as separate training examples. To improve the predicted distribution we also implemented a **Power selector** method. This method transforms the predicted probabilities by raising them to the language-specific powers that are selected to maximize the target metric.

4.2.1 DeepMistake-based models

Similarly to subtask 1, our models for subtask 2 are based on **MCL→DWUG$_{es}$+XLWSD$_{es}$**. To predict the level of disagreement between annotators, we employed the comb_dmn aggregation function during the fine-tuning process. In contrast to l1ndotn returning a two-dimensional vector of distances which should represent sense proximity but not ambiguity or difficulty leading to disagreements, comb_dmn returns a high-dimensional representation potentially preserving more information relevant to the subtask. To test this hypothesis, we also trained a model using the l1ndotn function for comparison.

All models were fine-tuned using the same optimizer hyperparameters as in Subtask 1. Early stopping was performed based on the average Spearman's ρ across all languages.

comb_dmn,mse@CoMeDi-#less4: Our initial approach was a regression model that directly pre-

Model/Participant	Spearman's ρ ZH	DE	EN	NO	RU	ES	SV	AVG
comb_dmn,ce@CoMeDi-#less4	0.301	**0.204**	**0.078**	**0.286**	**0.175**	**0.187**	**0.350**	**0.226**
daalft	**0.539**	0.108	0.042	0.272	0.167	0.115	0.296	0.220
comedy_baseline_2	0.485	0.085	0.060	0.235	0.116	0.078	0.079	0.163
chuphuocvip123	0.362	0.099	0.018	0.156	0.050	0.012	0.172	0.124
comedi_baseline	0.387	0.093	0.064	0.076	0.049	0.077	0.081	0.118
JuniperLiu	0.358	0.022	0.038	-0.042	0.067	0.040	0.090	0.082
sunfz1	0.302	-0.001	0.045	-0.071	0.069	0.038	0.089	0.067

Table 4: Evaluation results on the subtask 2. Best results for each language are in **bold font**.

dicts the quantity of interest. This model was fine-tuned to predict the mean of pairwise absolute judgment differences between annotators, employing the mean squared error (MSE) loss function. Examples containing fewer than four annotations were excluded to ensure robustness of the training data, as such examples might not provide sufficient information about the distribution[2]. In particular, this filtration removes all examples from the Chinese and Norwegian train sets.

comb_dmn,bce@CoMeDi-#less4: Since the mean disagreement is estimated from as few as 4-5 annotations for most usage pairs, learning a good regression model from such noisy targets may be impossible. Thus, we experimented with less noisy targets even though they are indirectly related to the mean disagreement we are interested in. This model was trained with the binary cross-entropy (BCE) loss to determine if all annotators provided the same answer for a pair of usages. All examples with less than 4 annotations were excluded from the train set.

comb_dmn,ce@CoMeDi-#less4: Instead of directly predicting the mean disagreement, we can try training a model that predicts the whole distribution of annotations for a given pair of usages, and then estimate the mean disagreement from that distribution. Technically, 4 class models trained for subtask 1 return the probability distribution over possible annotations, but since they are trained to predict the median annotation only they have no chance to learn anything about disagreements between annotators. Thus, for subtask 2 we do not aggregate annotations of each usage pair but instead fine-tune the model on each individual annotation as a separate training example.

comb_dmn,ce@CoMeDi: To verify if removing usage pairs with less than four annotations is really helpful when training on individual annotations, we trained this model on annotations of all pairs. This increased the number of training examples by 5x.

l1ndotn,ce@CoMeDi-#less4: In order to check if our initial decision to use DeepMistake with the comb_dmn aggregation function for subtask 2 was optimal, we trained this model which is similar to the previous one but employs the l1dotn aggregation function instead of comb_dmn.

4.2.2 Power selector

For models trained on individual annotations and schemed to predict the probability distribution across annotators, we designed an approach to optimize their predictions for the target metric. Specifically, for each language, we fit four powers α_i to which the class probabilities p_i are raised:

$$\hat{p}_i = \frac{p_i^{\alpha_i}}{\sum_{j=1}^{4} p_j^{\alpha_j}} \quad (2)$$

This method is inspired by the temperature softmax[3] often used to undersample / oversample frequent / rare classes, e.g. in word2vec (Mikolov et al., 2013). This method is also related to common calibrating techniques (Guo et al., 2017). The selection of these powers is performed similarly to the NMthres process, utilizing the Nelder-Mead optimization method to maximize Spearman's ρ.

4.2.3 Inference methods

In case of the model fine-tuned with the MSE loss between the predicted and gold mean disagreements, we directly return its predictions. For the model trained with the BCE loss function we return the predicted probability that there are some

[2]In the preliminary experiments we tried fine-tuning models directly predicting mean disagreement with both mse and bce losses on all examples, but they achieved near zero performance. This is probably due to very noisy estimates of the mean disagreement when less than 4 annotations are available. Thus, for the second subtask we mostly experimented with models trained on examples with 4 or more annotations.

[3]The Power Selector can be viewed as a more generalized approach compared to temperature scaling in softmax function. While the temperature softmax technique uniformly raises all probabilities to the same power, our approach assigns a distinct power to each probability individually.

Model	Spearman's ρ ZH	DE	EN	NO	RU	ES	SV	AVG
aggregated annotations								
comb_dmn,mse@CoMeDi-#less4	0.462	**0.241**	0.110	0.215	0.192	0.136	0.238	0.228
comb_dmn,bce@CoMeDi-#less4	0.497	0.237	0.089	0.300	0.212	0.120	0.245	0.243
separate annotations w/o pows								
comb_dmn,ce@CoMeDi	0.484	0.206	0.130	0.276	0.237	0.232	0.262	0.261
comb_dmn,ce@CoMeDi-#less4	0.426	0.197	**0.148**	0.298	0.183	0.123	0.297	0.239
l1ndotn,ce@CoMeDi-#less4	0.605	0.148	0.084	0.448	0.162	0.108	0.282	0.262
separate annotations w/ pows$_{dev}$								
comb_dmn,ce@CoMeDi	0.571	0.218	0.128	0.421	**0.256**	0.159	0.302	0.293
comb_dmn,ce@CoMeDi-#less4	0.301[1]	0.204[1]	0.078[1]	0.286[1]	0.175[1]	0.187[1]	0.350[1]	0.226
l1ndotn,ce@CoMeDi-#less4	**0.616**	0.148	0.084	0.454	0.162	0.108	0.282	0.265
separate annotations w/ pows$_{train}$								
comb_dmn,ce@CoMeDi	**0.616**	0.236	0.129	0.424	0.253	**0.236**	0.297	**0.313**
comb_dmn,ce@CoMeDi-#less4	0.574	**0.241**	0.143	0.294	0.194	0.161	**0.360**	0.281
l1ndotn,ce@CoMeDi-#less4	**0.616**	0.227	0.080	**0.456**	0.234	0.109	0.266	0.284

Table 5: Post-evaluation results on the test set of subtask 2. Best results for each language are in **bold font**. Superscripts refer to our submission during the evaluation phase.

disagreements between annotators assuming that higher probability corresponds to larger disagreements. For models trained on individual annotations we take the whole predicted probability distribution over 4 classes and calculate its standard deviation. Additionally, the power selector can be applied.

4.3 Evaluation results

During the evaluation phase, our sole submission was from the comb_dmn,ce@CoMeDi-#less4 model, which incorporated a power selection model optimized on the development set. This model achieved the best performance across all languages except for Chinese, where it recorded the poorest results among all participants. Comprehensive results of the evaluation phase are presented in Table 4.

4.4 Post-evaluation experiments

Upon completion of the evaluation phase, we proceeded to evaluate all models using the test set. For models that were trained on individual annotations, we explored several strategies: not employing the power selection model, fitting it on the CoMeDi train sets, and fitting it on the CoMeDi development sets. The results of these evaluations are detailed in Table 5. It is clear that the power selector helps significantly, and it is better to fit it on the train sets. Removing examples with less than 4 annotations hurts the performance on average across languages, at least when training on individual annotations, though the results vary from language to language. Comparing l1ndotn with comb_dmn, the results are

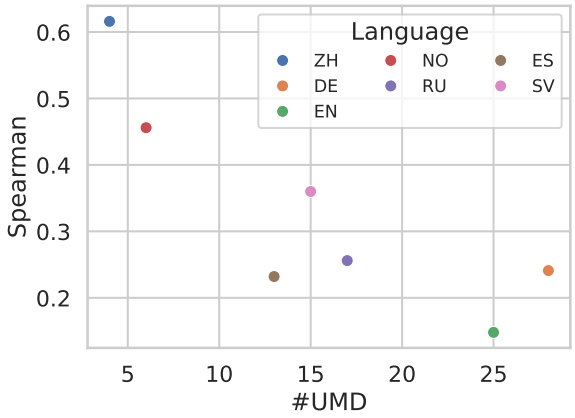

Figure 4: Best achieved Spearman's ρ and #UMD per language on the test set.

not consistent across languages as well requiring more experiments to draw reliable conclusions.

Comparing the results for different languages, Chinese and Norwegian have higher metrics while there are only two annotations per example for this languages which should result in quite noisy ground truth mean disagreement. We hypothesised that the good results may be related to fewer unique values of the mean disagreement when there are fewer annotators. We investigated the impact of the number of unique values of the mean pairwise absolute judgment (#UMD) on the Spearman's correlation across different languages. For each language, we selected the best result achieved during the post-evaluation phase and #UMD, as depicted in Figure 4.

Our analysis indicates that languages with the best results – Chinese and Norwegian, exhibit rel-

atively low #UMD, characterized by less than 7 unique values for mean disagreement. Conversely, English and German, which have some of the lowest ρ, are associated with the highest #UMD.

5 Conclusion

We have proposed the winning solutions for the CoMeDi shared task and experimented with different design choices. To our surprise fine-tuning a 4 class WiC model on the training data from the shared task has shown worse results than thresholding the original binary WiC model. Whether it is due to the insufficient or noisy training data, or bad correlation between the cross-entropy loss and the target metric Krippendorff's alpha remains to be investigated. A promising direction for the future experiments is designing surrogate losses that are better correlated with Krippendorff's alpha. We also observed that removing CoMeDi training data in Chinese significantly improves results, including the results for Chinese. A reasonable next step may be selecting an optimal combination of training sets for each test language separately.

For the second subtask our best solution was learning to predict the whole distribution of annotations for a given usage pair. In the future work it is reasonable to try alternative loss functions as well, e.g. minimizing the KL-divergence between the predicted and the real probability distributions.

References

Anna Aksenova, Ekaterina Gavrishina, Elisey Rykov, and Andrey Kutuzov. 2022. Rudsi: graph-based word sense induction dataset for russian. *arXiv preprint*.

Nikolay Arefyev, Maksim Fedoseev, Vitaly Protasov, Daniil Homskiy, Adis Davletov, and Alexander Panchenko. 2021. Deepmistake: Which senses are hard to distinguish for a word-in-context model. volume 2021-June, pages 16–30.

Carlos Santos Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Ivan Vulić, and Mohammad Taher Pilehvar. 2020. SemEval-2020 task 3: Graded word similarity in context. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 36–49, Barcelona (online). International Committee for Computational Linguistics.

Pierluigi Cassotti, Lucia Siciliani, Marco de Gemmis, Giovanni Semeraro, and Pierpaolo Basile. 2023. Xl-lexeme: Wic pretrained model for cross-lingual lexical semantic change. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Jing Chen, Emmanuele Chersoni, Dominik Schlechtweg, Jelena Prokic, and Chu-Ren Huang. 2023. ChiWUG: A graph-based evaluation dataset for Chinese lexical semantic change detection. In *Proceedings of the 4th International Workshop on Computational Approaches to Historical Language Change*, Singapore. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Adis Davletov, Nikolay Arefyev, Denis Gordeev, and Alexey Rey. 2021. LIORI at SemEval-2021 task 2: Span prediction and binary classification approaches to word-in-context disambiguation. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 780–786, Online. Association for Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. *ArXiv*, abs/1706.04599.

Anna Hätty, Dominik Schlechtweg, and Sabine Schulte im Walde. 2019. SURel: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics*, pages 1–8, Minneapolis, MN, USA.

Daniil Homskiy and Nikolay Arefyev. 2022. DeepMistake at LSCDiscovery: Can a multilingual word-in-context model replace human annotators? In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 173–179, Dublin, Ireland. Association for Computational Linguistics.

Klaus Krippendorff. 2018. *Content Analysis: An Introduction to Its Methodology*. SAGE Publications.

Sinan Kurtyigit, Maike Park, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Lexical Semantic Change Discovery. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online. Association for Computational Linguistics.

Andrey Kutuzov and Lidia Pivovarova. 2021. Rushifteval: a shared task on semantic shift detection for russian. *Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Dialog conference*.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Enstad, and Alexandra Wittemann. 2022. NorDiaChange: Diachronic semantic change dataset for

Norwegian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2563–2572, Marseille, France. European Language Resources Association.

Federico Martelli, Najla Kalach, Gabriele Tola, and Roberto Navigli. 2021. SemEval-2021 task 2: Multilingual and cross-lingual word-in-context disambiguation (MCL-WiC). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 24–36, Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

John A. Nelder and R. Mead. 1965. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13648–13656.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Julia Rodina and Andrey Kutuzov. 2020. RuSemShift: a dataset of historical lexical semantic change in Russian. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*. Association for Computational Linguistics.

Dominik Schlechtweg. 2023. *Human and Computational Measurement of Lexical Semantic Change*. Ph.D. thesis, University of Stuttgart, Stuttgart, Germany.

Dominik Schlechtweg, Pierluigi Cassotti, Bill Noble, David Alfter, Sabine Schulte im Walde, and Nina Tahmasebi. 2024. More DWUGs: Extending and evaluating word usage graph datasets in multiple languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida. Association for Computational Linguistics.

Dominik Schlechtweg, Tejaswi Choppa, Wei Zhao, and Michael Roth. 2025. The CoMeDi shared task: Median judgment classification & mean disagreement ranking with ordinal word-in-context judgments. In *Proceedings of the 1st Workshop on Context and Meaning–Navigating Disagreements in NLP Annotations*, Abu Dhabi, UAE.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic Usage Relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, New Orleans, Louisiana.

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. 2021. DWUG: A large resource of diachronic word usage graphs in four languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7079–7091, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Charles Spearman. 1904. The proof and measurement of association between two things. *American Journal of Psychology*, 15:88–103.

Frank D. Zamora-Reina, Felipe Bravo-Marquez, and Dominik Schlechtweg. 2022. LSCDiscovery: A shared task on semantic change discovery and detection in Spanish. In *Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change*, pages 149–164, Dublin, Ireland. Association for Computational Linguistics.

A Cross-lingual transfer

In our preliminary experiments we fine-tuned seven DeepMistake models as 4-class classifiers on each train set from CoMeDi separately and evaluated each of them on each dev set separately. While fine-tuning on individual train sets and using 4-class fine-tuning objective proved to be suboptimal in the end, this preliminary experiment gives some ideas about usefulness of each train set for the performance on each language. Figure 5 shows how Krippendorff's alpha change while training these seven models. While we didn't have enough resources to optimize them all to the point of full convergence, some trends can clearly be observed from these curves. The Chinese train set is always one of the worst train sets for the Krippendorff's alpha on all dev sets, including the Chinese one. The Norwegian train set is the best when evaluating on Norwegian and one of the best for Chinese, but among the worst for all other languages. The German train set is among the best for all languages except for Chinese and Spanish where it is in the middle. Fine-tuning DeepMistake on a train set for the same language the evaluation is made on works best for German, Norwegian, Russian and Spanish, but not Chinese, English or Swedish. Based on bad model performance for all languages when fine-tuning on the train set in Chinese, when fine-tuning

the binary DeepMistake model used in our submissions on all CoMeDi training data we excluded training examples in Chinese. It seems potentially beneficial to construct an optimal subset of training data for each language separately, e.g. excluding Norwegian from the training data of a model that is not targeted at Norwegian, but we leave systematic experiments in this direction for the future work.

For comparison, figure 6 shows the learning curves for the same models but taking accuracy as an evaluation metric instead of Krippendorff's alpha. Surprisingly, the observations drastically differ when changing the evaluation metric. The best accuracy on the Chinese dev set is achieved when training on English or Chinese train sets. The model trained on Norwegian is now among the best models for all dev sets. And the one trained on German is among the worst models for all dev sets except German and English. This shows that a model achieving the best accuracy may be among the worst for Krippendorff's alpha and vice versa. See appendix C for a more in-depth analysis of this discrepancy.

B Detailed model comparison

In Figure 7, we compare different DeepMistake models. Training data of these models has no overlap with following CoMeDi dev/test sets: German, English, Norwegian, Chinese, and Swedish. For German and English, all models with NMthres exhibit similar performance across both sets. In contrast, for Norwegian and Chinese, the models MCL and MCL→DWUG$_{es}$+XLWSD$_{es}$ perform better than others. Meanwhile, in Swedish, the models MCL→RSS and MCL+RSS+DWUG$_{es}$+XLWSD$_{es}$ emerge as superior.

Our analysis, depicted in Figure 8, which compares different training sets, reveals that fine-tuning models with NMthres on CoMeDi-2,3 significantly improves performance for Swedish subsets. For other languages, using the complete CoMeDi training data is equally effective, and sometimes even more beneficial. While CoMeDi-ZH and CoMeDi-ZH-DE do not show much difference from CoMeDi in most cases, with the exception of Chinese, they generally perform better overall.

As shown in Figure 9, the DeepMistake model with NMthres consistently outperforms variant without it across all languages, except for Swedish. This trend is also observed in the 2-class fine-tuned models. Additionally, when comparing the 4-class fine-tuned model trained on CoMeDi-ZH with the DeepMistake model without NMthres, the fine-tuned model shows better performance on all development sets, except Russian. However, on the test set, the situation reverses, with the non-fine-tuned model performing better.

C Chinese mystery

Removing the Chinese train set when fine-tuning DeepMistake as a binary classifier on the CoMeDi training data strikingly improves the Krippendorff's alpha on the Chinese development set in subtask 1 (from 0.48 to 0.71) while giving only a small improvement in accuracy (from 0.88 to 0.90). Here we investigate why this happens. Krippendorff's α is defined as:

$$\alpha = 1 - \frac{D_o}{D_e}, \tag{3}$$

where D_o is the observed disagreement and D_e the disagreement expected by chance. The observed disagreement in general case is defined as:

$$D_o = \frac{1}{n} \sum_{i \in R} \sum_{j \in R} \delta_{ij} \sum_{u} \frac{m_u * n_{iju}}{P(m_u, 2)}, \tag{4}$$

where n is the total number of labels (in our case both predicted labels and ground truth labels), R is the set of possible labels, u is a usage pair, m_u is the number of labels assigned to the usage pair u. Finally, n_{iju} is the number of pairs (i, j) consisting of labels assigned to u.

For the ordinal version of Krippendorff's α:

$$\delta_{ij} = (\sum_{k=i}^{j} n_k - \frac{n_i + n_j}{2})^2, \tag{5}$$

where n_x is the number of labels equal to x among both the predicted and the ground truth labels of all usage pairs.

In our case $m_u = 2$ because for each example there is a ground truth label and a predicted label. After substituting this into formula 4 we get $D_o = \frac{1}{n} \sum_{i=1}^{4} \sum_{j=1}^{4} \delta_{ij} * 2 \sum_u I[y_u = i, \hat{y}_u = j] = \frac{1}{n} \sum_{i=1}^{4} \sum_{j=1}^{4} 2\delta_{ij} c_{ij}$, where c_{ij} is the number of usage pairs with the ground truth label of i and the predicted label of j. Thus, the final formula for Krippendorff's α in our case can be written as:

$$\alpha = 1 - \sum_{i=1}^{4} \sum_{j=1}^{4} \frac{2\delta_{ij} c_{ij}}{n * D_e} \tag{6}$$

Figure 10 plots confusion matrices where a cell (i, j) shows the contribution $\frac{2\delta_{ij}c_{ij}}{n*D_e}$ of the corresponding type of errors (when class i is misclassified as class j) to Krippendorff's α, and also standard confusion matrices showing proportions of examples with different predicted and ground truth labels. We can observe proportions of different types of errors (i, j) and how they contribute to the final value of Krippendorff's α in 6.

While the error rates of two models on the Chinese dev set are comparable, the proportions of different types of errors differ drastically. For the model trained on all training sets including the Chinese one all errors are related to predicting 4 instead of some other class. Such types of errors strongly reduce Krippendorff's alpha because of the dominating frequency of label 4 resulting in large values of δ_{i4} (see formula 5) and thus large contribution of c_{i4} in formula 6. On the other hand, the model trained without the Chinese train set produces fewer errors of such types and more errors related to predicting 1 instead of some other class. However, the latter types of errors make much smaller contribution to Krippendorff's alpha (unless the correct label is 4).

For the development sets in languages other than Chinese such a large difference in error types and thus Krippendorff's alpha is not observed, as shown in Figures 11, 12. We believe that this is related to the proportions of negative examples (classes 1 and 2) in the training sets for different languages, see Figure 13. In the Chinese train set this proportion is negligible, thus, the model learns to predict the positive class for inputs in Chinese unless there are very strong evidences in favour of negative class. In the Chinese dev set the proportion of classes 1 and 2 is significantly larger, so this learnt strategy leads to some errors for examples of these classes which are fatal for Krippendorff's alpha. When the Chinese train set is excluded the model cannot learn any specific strategy for inputs in Chinese. For other languages the proportions of negative examples in the corresponding train sets are reasonable and for them we don't observe significant changes in the proportions of errors of different types between two models.

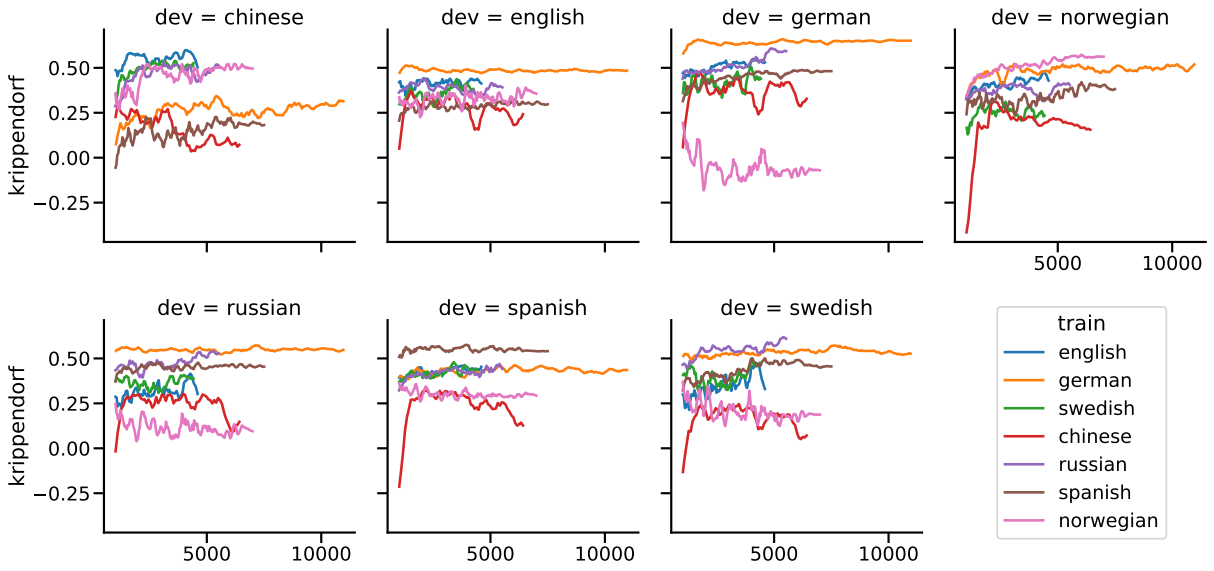

Figure 5: Cross-lingual transfer evaluation. Krippendorff's alpha on the dev sets for the DeepMistake models being fine-tuned as a 4-class classifiers on the train sets for each language separately.

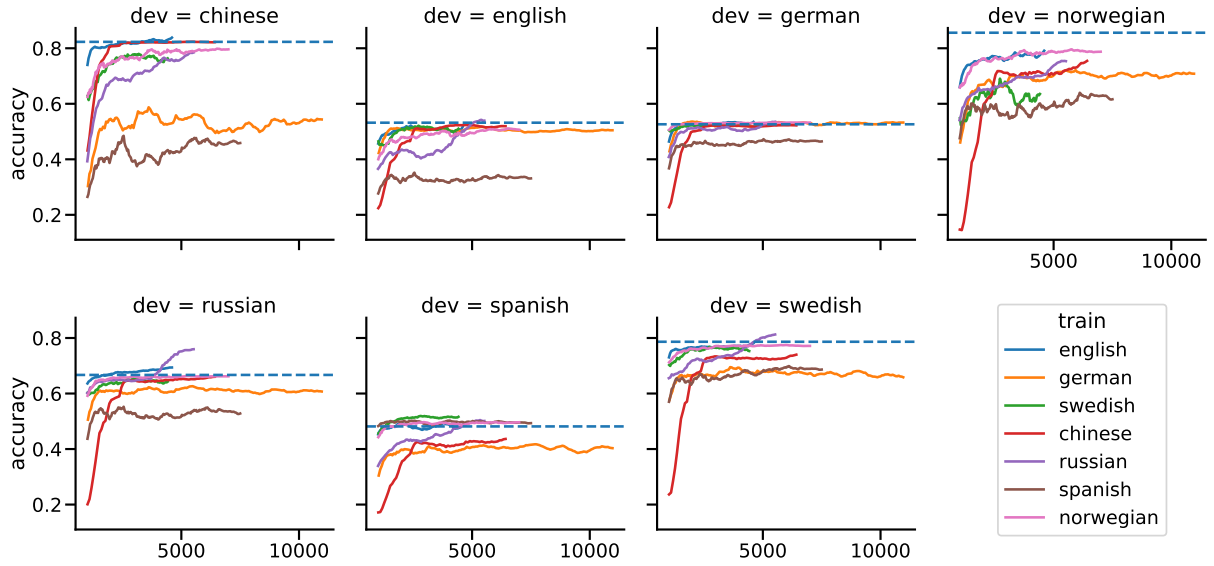

Figure 6: Cross-lingual transfer evaluation. Accuracy on the dev sets for the DeepMistake models being fine-tuned as a 4-class classifiers on the train sets for each language separately. The horizontal dashed lines show the proportion of the most frequent class.

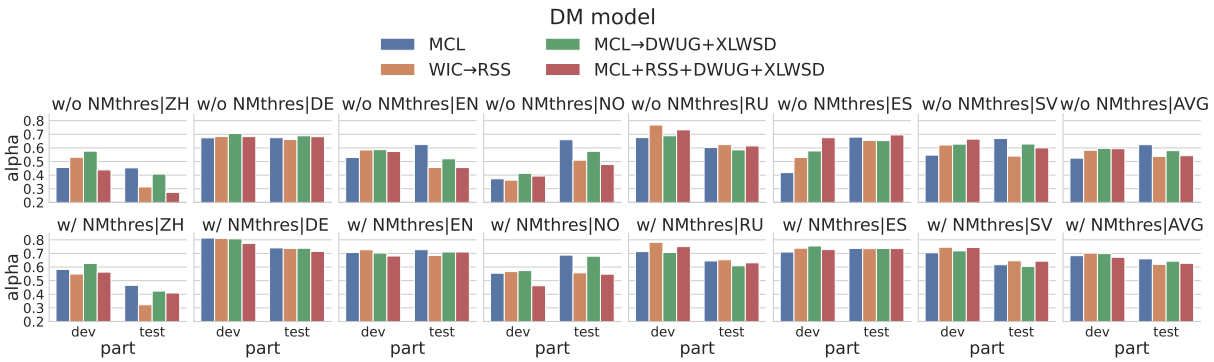

Figure 7: Krippendorff's α of DeepMistake models w/ and w/o NMthres. The simplified version of the plot is shown in Figure 1.

Figure 8: Krippendorff's α of 2-class DeepMistake models fine-tuned on different subsets of CoMeDi train data. The simplified version of the plot is shown in Figure 2.

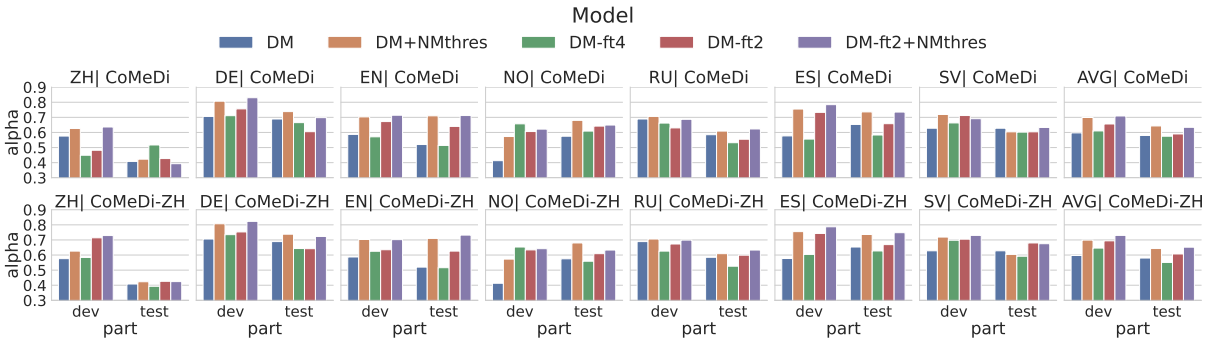

Figure 9: Krippendorff's α of models fine-tuned on CoMeDi and CoMeDi-ZH models. The simplified version of the plot is shown in Figure 3.

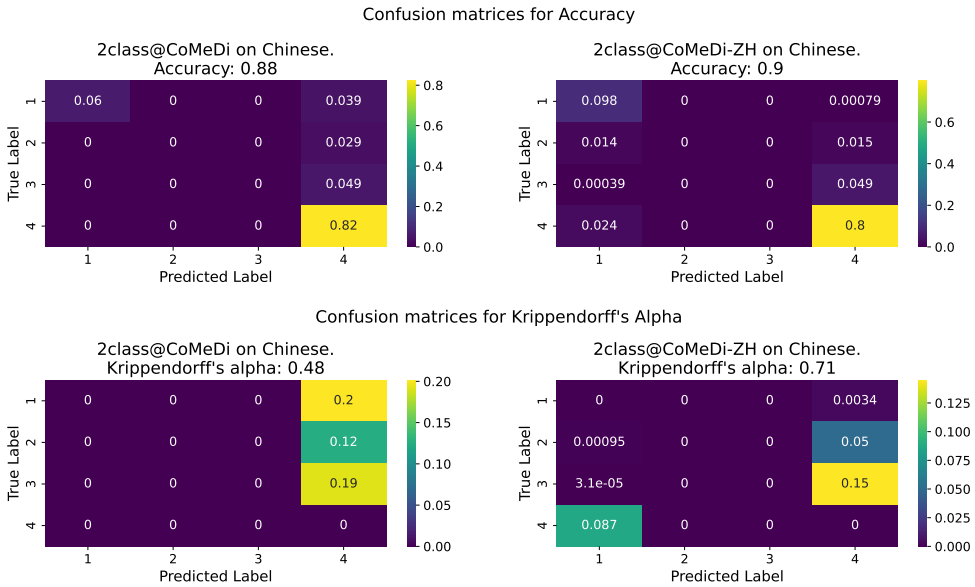

Figure 10: Confusion matrices built on the Chinese CoMeDi dev set for the 2class@CoMeDi and 2class@CoMeDi-ZH models. In confusion matrices for Krippendorff's α below (i, j)-th cell quantifies the contribution $\frac{2\delta_{ij}c_{ij}}{n*D_e}$ of the corresponding type of errors to Krippendorff's α. These contributions sum up to 1 - α. In confusion matrices for accuracy above it quantifies the proportion of examples of the corresponding type, accuracy is the sum of diagonal cells.

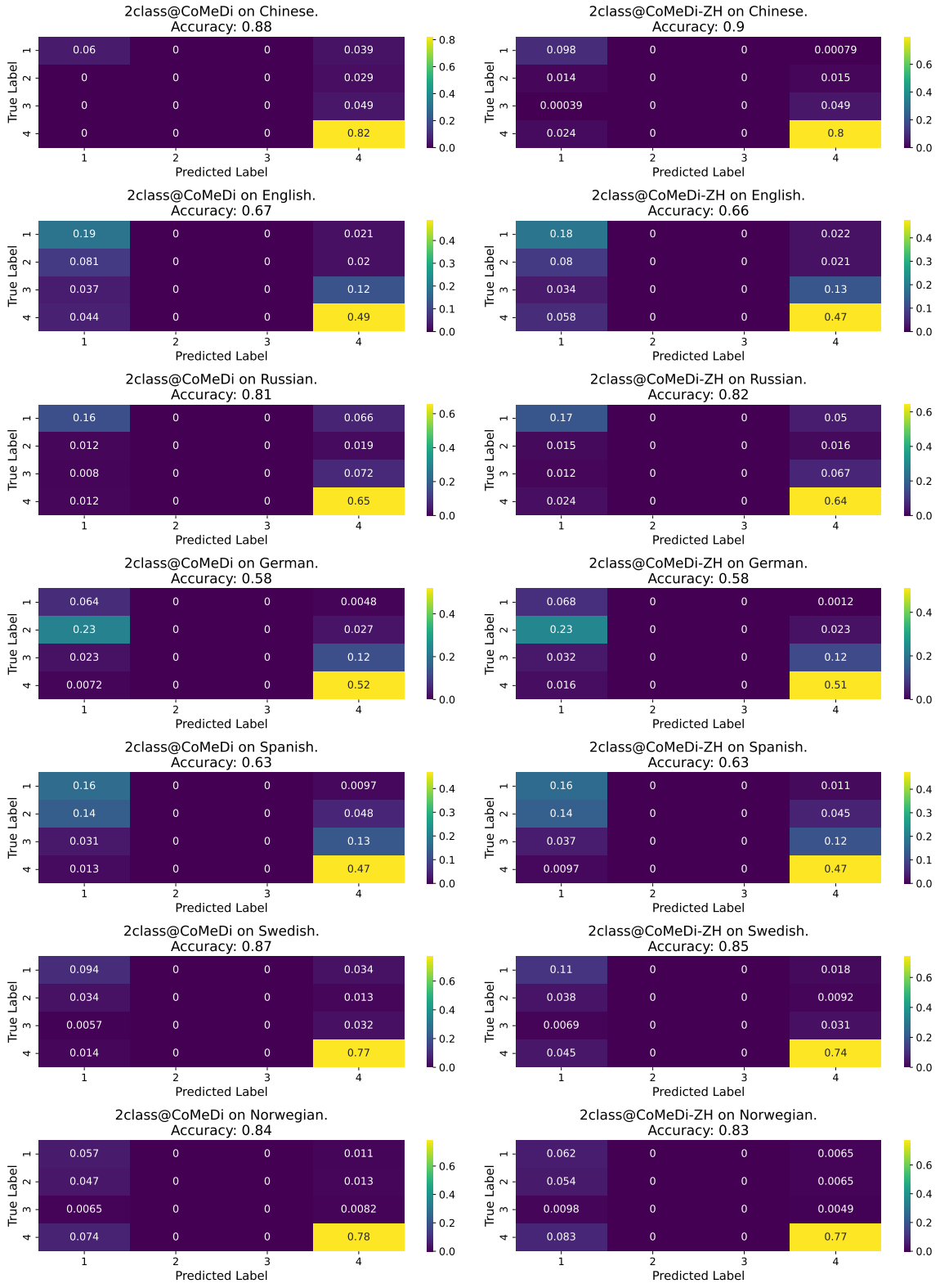

Figure 11: Confusion matrixes on CoMeDi dev sets of 2class@CoMeDi and 2class@CoMeDi-ZH.

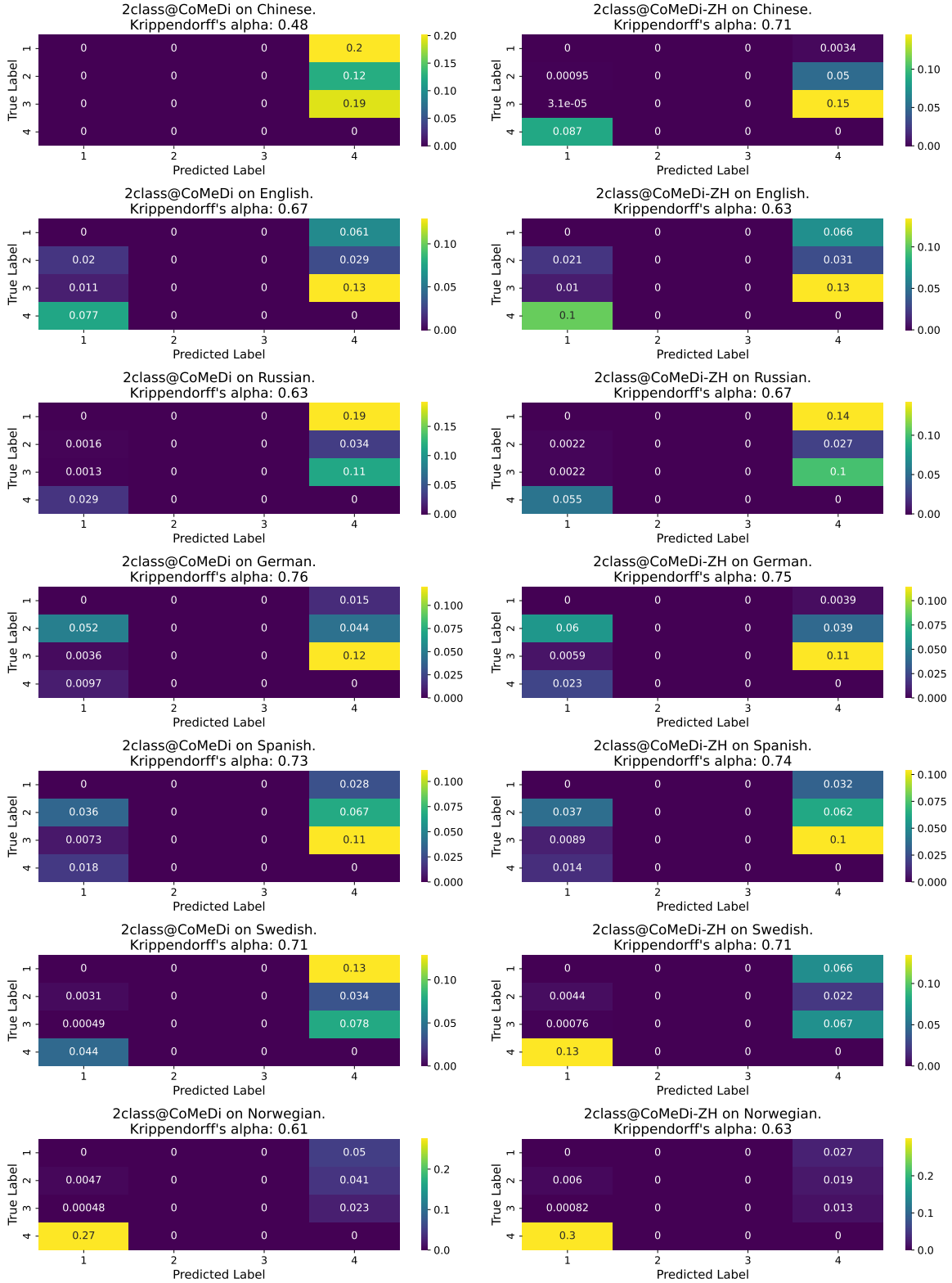

Figure 12: Confusion matrices built on the CoMeDi dev sets for the 2class@CoMeDi and 2class@CoMeDi-ZH models, (i, j)-th cell quantifies the contribution $\frac{2\delta_{ij}c_{ij}}{n*D_e}$ of the corresponding type of errors to Krippendorff's α. These contributions sum up to 1 - α.

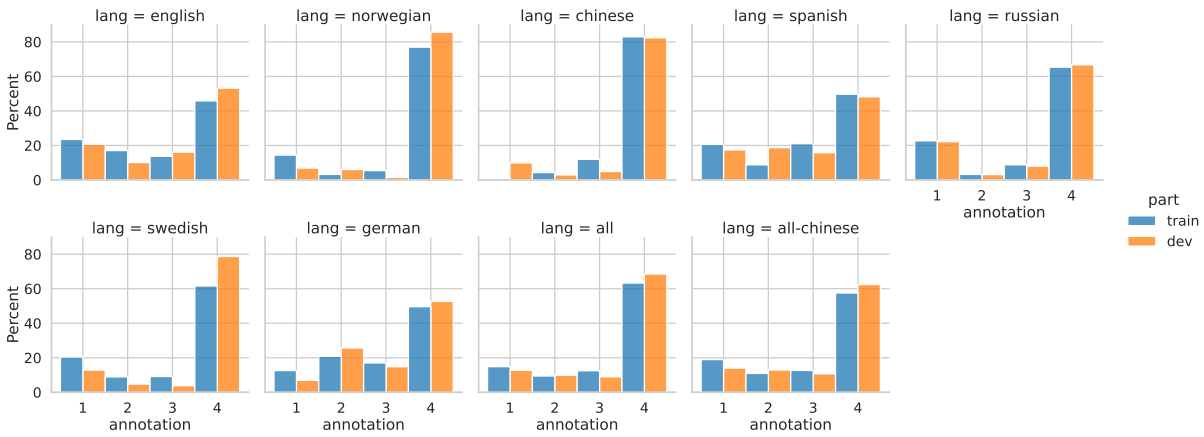

Figure 13: Class proportions in the train and dev sets for different languages and in combined train and dev sets.