Composing Structure-Aware Batches for Pairwise Sentence Classification

Andreas Waldis[1,2], **Tilman Beck**[2], **Iryna Gurevych**[2]
[1]Information Systems Research Lab
Department of Computer Science, Lucerne University of Applied Sciences and Arts
[2]Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science, Technical University of Darmstadt
www.hslu.ch
www.ukp.tu-darmstadt.de

Abstract

Identifying the relation between two sentences requires datasets with pairwise annotations. In many cases, these datasets contain instances that are annotated multiple times as part of different pairs. They constitute a structure that contains additional helpful information about the inter-relatedness of the text instances based on the annotations. This paper investigates how this kind of structural dataset information can be exploited during training. We propose three batch composition strategies to incorporate such information and measure their performance over 14 heterogeneous pairwise sentence classification tasks. Our results show statistically significant improvements (up to 3.9%) - independent of the pre-trained language model - for most tasks compared to baselines that follow a standard training procedure. Further, we see that even this baseline procedure can profit from having such structural information in a low-resource setting.[1]

1 Introduction

Datasets that define pairwise relations between sentence-level text instances are widely used in Natural Language Processing (NLP). They describe the relation of sentence pairs with an annotated label. Common examples of such pairwise classification tasks are Paraphrase Identification (Wang et al., 2017; Dolan et al., 2004), Natural Language Inference (Williams et al., 2018a; Bowman et al., 2015), Semantic Textual Similarity (Cer et al., 2017; Reimers et al., 2019), or Argument Convincingness (Habernal and Gurevych, 2016).

In many such datasets (eq. six out of 11 GLUE tasks), single sentences can occur in multiple pairwise annotations. Figure 1 shows such an example where three annotated pairs share a common question. Besides the annotations themselves, such structural properties of datasets carry additional helpful information about the inter-relatedness of the text instances. We argue that the defined discriminative attributes of a text instance are learned most readily when the instance is encountered in multiple contexts. Therefore, we hypothesize that a (neural) learner can utilize such additional information when provided appropriately.

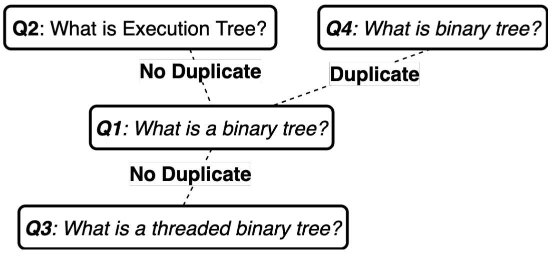
Q2: What is Execution Tree?
Q4: What is binary tree?
No Duplicate
Duplicate
Q1: What is a binary tree?
No Duplicate
Q3: What is a threaded binary tree?

Figure 1: Example of three pairwise annotations (edges) using four unique questions (nodes), taken from the QQP dataset (Wang et al., 2018). Q1 is the common element of all annotations.

There are several ways to control the training process considering such external information. Contrastive learning (Chen et al., 2020; Giorgi et al., 2021; Gao et al., 2021) aims at learning text representations in a self-supervised fashion where similar instances are aligned and dissimilar pairs are separated using an external measure - i.e. semantic similarities. In Curriculum Learning (Bengio et al., 2009) the training data order is determined by the estimated difficulty of the instances using a additional heuristic. Inspired by such work, we want to examine whether considering the dataset-annotation structures affects the models' performance. But neither we create new pairs nor open a dependency to an external heuristic. More specifically, we present three different strategies to compile training batches that consider that text instances occur in multiple pairwise configurations. This approach is also motivated by recent work (Dodge et al., 2020; Zhou et al., 2020) investi-

[1] We provide the code and hyperparameter optimisation details at https://github.com/UKPLab/acl2022-structure-batches

Findings of the Association for Computational Linguistics: ACL 2022, pages 3031 - 3045
May 22-27, 2022 ©2022 Association for Computational Linguistics

gating the effect of training order and inter-instance correlations on model performance.

We evaluate the strategies on 14 heterogeneous tasks from different domains and in two different scenarios to measure the generalizability of our approach. Our experimental results show significant performance improvements on a wide range of tasks for both scenarios compared to a standard training setup. Our contributions can be summarized as follows:

1. we propose three different batching strategies for pairwise text classification tasks to integrate inter-instance relations into the training procedure
2. we show statistically significant performance improvements on a wide range of heterogeneous tasks in our experimental results
3. we discuss the role of dataset characteristics, additional computational complexity and the stability of our approach

To foster the reproducibility of our work, we publish all experimental code and hyperparameters.

2 Approach

By analysing different pairwise annotated datasets, we found that various ones contain single sentences that occur in multiple annotation instances (see *Degree* in Table 1). For example, every sentence of the QNLI dataset is annotated with 1.9 other sentences on average. With our approach, we want to exploit this untouched information to improve the tasks' performance. Thus, we show how we capture this information in a graph structure and strategies to implicitly present it to the neural network.

2.1 Annotation Graph (AG)

We use a graph structure to represent all annotations of a dataset - as in Figure 1. In this graph, nodes are unique sentences, edges represent a label for a pair of them, and the degree k indicates the number of connected edges of a node. Based on the typed of annotations, this graph can be directed or undirected.

In Figure 2 we show an example of such a graph structure and its construction. It includes six sentences $V = \{V_1, ..., V_6\}$ and seven pairwise annotations $E = \{(V_1, V_2), ..., (V_5, V_6)\}$. Within the graph, we define a nodes' neighbourhood as all directly connected nodes - like $\{V_1, V_2\}$ for node V_3. In the case of an edge, we consider edges connect to one of its starting points as the neighbours - for example, (V_4, V_5) and (V_5, V_6) are neighbours.

Using this structure, we define different operations: $f_e(n)$ returns all edges of a given node n, and $f_s(c, x)$ randomly samples x elements from a collection of edges c. Further, we use the average degree μ_k, its standard deviation σ_k, and coefficient of variation ($CV_k = \frac{\sigma_k}{\mu_k}$) to characterise an AG. Using these measurements, we can group the selected tasks into three groups (see Table 1). The first one (G_1) includes tasks (all in-domain tasks, UKP-A, BWS) that do not show extreme patterns in the graph ($CV \approx 1$). The second group, G_2 (Arg-Conv, and Evi-Conv), has a high average degree but a lower std. dev. ($CV < 1$). The third group G_3 fits tasks (Evi-St, ArgQ-St, Arg-KP) where a few nodes with a high *degree* are connected to many others with a small *degree* ($CV > 1$).

	Dataset	Label	Degree	Group	Metric
In-Domain	SICK-NLI	3-Class	3.2 ± 2.1	G_1	*acc*
	SICK-REL*	1-5	3.2 ± 2.1	G_1	ρ
	RTE	3-Class	1.1 ± 0.6	G_1	*acc*
	QNLI	Binary	1.9 ± 0.8	G_1	*acc*
	MNLI-m	3-Class	1.5 ± 0.9	G_1	*acc*
	MNLI-mm	3-Class	1.5 ± 0.9	G_1	*acc*
	QQP	Binary	1.6 ± 2.2	G_1	F_1
Cross-Topic	UKP-A	Binary	3.5 ± 3.0	G_1	F_1 *macro*
	BWS*	0-1	1.6 ± 1.5	G_1	ρ
	Arg-Conv	Binary	22.2 ± 4.6	G_2	*acc*
	Evi-Conv	Binary	6.2 ± 4.4	G_2	*acc*
	Evi-St	Binary	1.9 ± 5.8	G_3	F_1 *macro*
	Arg-KP	Binary	7.1 ± 18.1	G_3	F_1 *macro*
	ArgQ-St	3-Class	2.0 ± 20.7	G_3	*acc*

Table 1: Overview of the 14 used datasets for the In-Domain and Cross-Topic Scenario. In the latter we train on different topics then we evaluate. *Degree* denotes average number of edges of a node and datasets marked with (*) are regression tasks.

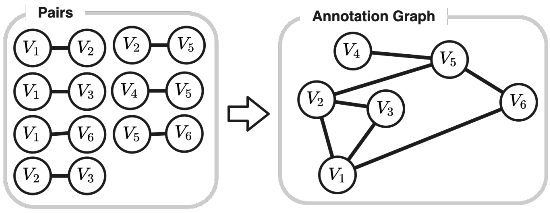

Figure 2: Construction of an annotation graph (AG) with a degree of 2.5 ± 0.84 and $CV_k = 0.34$.

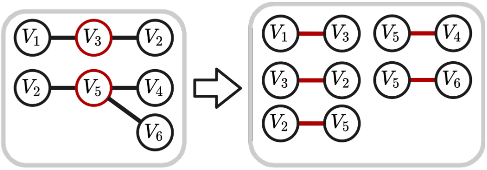

Figure 3: Example batch for the strategy NODE

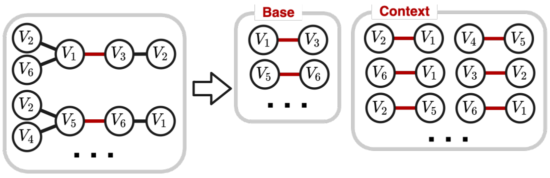

Figure 4: Example batch for the strategy EDGE-I

2.2 Batching Strategies

With the following strategies, we randomly traverse through the graph, either with the focus on the neighbourhood of nodes (NODE) or edges (EDGE-I, EDGE-II). Since NODE will incorporate all the neighbours of a node, it could overfit towards them - given a high μ_k. Thus, EDGE-I and EDGE-II focus on just a limited neighbourhood to reduce this potential dominance.

NODE This strategy composes a batch by focusing on common nodes within the AG. Figure 3 shows this process with a set of example nodes ($N = \{V_3, V_5\}$). For every node n, we select all connected edges ($\{(V_1, V_3), (V_3, V_2)\}$ for node V_3). The loss L is equal to the average error (using the cross-entropy objective function $\mathcal{J}$) for each edge e, as defined in Equation 1.

$$L = -\frac{1}{\sum_n^N |f_e(n)|} \sum_n^N \sum_e^{f_e(n)} \mathcal{J}(\hat{y_e}, y_e) \tag{1}$$

EDGE-I The second strategy starts from a set of randomly selected edges E to construct a single batch. For each base edge $e \in E$, a set of context edges E' is sampled from the neighbourhood of e. To select these neighbours, we consider the two nodes (i, j) that are the starting points of e - as in Equation 2. For both of them, we randomly select two[2] directly connected edges using f_s (Equation 3). Figure 4 shows an example batch that considers the two base edges $B = \{(V_1, V_3), (V_5, V_6)\}$. For base edge (V_1, V_3), the set of context edges is $E' = \{(V_1, V_2), (V_1, V_6), (V_3, V_2)\}$. To calculate the loss, we sum the average error of base and the context edges - as in Equation 4.

$$E' = \bigcup_{(i,j)}^{E} f'_e(i) \cup f'_e(j) \setminus \{(i,j)\} \tag{2}$$

$$f'_e = \begin{cases} f_e(n) & \text{if } |f_e(n)| \leq 2 \\ f_s(f_e(n), 2) & \end{cases} \tag{3}$$

$$L = \frac{-1}{|E|} \sum_e^E \mathcal{J}(\hat{y_e}, y_e) + \frac{-1}{|E'|} \sum_{e'}^{E'} \mathcal{J}(\hat{y_{e'}}, y_{e'}) \tag{4}$$

EDGE-II Within EDGE-I all context edges are treated equally and independently of their base edge. In EDGE-II we adapt the calculation of the loss to focus on the fact that the neighbourhood size of base edges can vary. First we sum the error of an base edge e with the average error of its neighbours e' (as in Equation 5). Afterwards, we average this sum over all e in E (Equation 6).

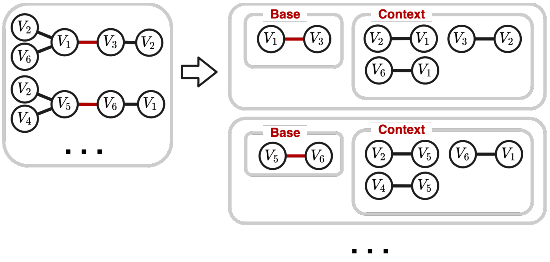

Figure 5: Example batch for the strategy EDGE-II

$$\mathcal{J}' = \mathcal{J}(\hat{y_e}, y_e) + \frac{1}{|E'^{(e)}|} \sum_{e'}^{E'^{(e)}} \mathcal{J}(\hat{y_{e'}}, y_{e'}) \tag{5}$$

$$L = \frac{-1}{|E|} \sum_e^E \mathcal{J}'(e) \tag{6}$$

Batch Composition Due to the nature of the described strategies, a single training instance can be contained in multiple batches. For NODE, every edge (i.e. training instance) is used twice as both contained nodes are sampled individually. In the case of EDGE-I and EDGE-II, the occurrence of one instance depends on how many times it is sampled as context edge and is affected by the AGs

[2] We check different node numbers during early preliminary experiments and found two works the best.

density. The chances to sample an edge as a context edge are higher in a more dense area. Thus, the effective number of instances processed within a batch can vary. Note, when we speak of batch size, we refer instead to the number of initially sampled nodes or edges, not the effective one.

3 Data and Training Setup

3.1 Datasets

We evaluate our approach on 14 heterogeneous pairwise classification tasks in two scenarios.[3] Table 1 shows an overview of the tasks including the label type, the degree (μ_k, σ_k), and evaluation metrics.

The first scenario aims at evaluating our general idea using standard natural language understanding tasks, e.g. GLUE (Wang et al., 2018). In the second scenario, we use tasks for the challenging cross-topic evaluation, where train, development, and test set covers different topics to measure the generalizability. For this scenario, we rely on Argument Mining tasks, which include sentence-level arguments assigned to a specific topic (Stab and Gurevych, 2017; Reimers and Gurevych, 2019a).

In-Domain Scenario The first scenario consists of five tasks (RTE, MNLI, QNLI, QQP) from the GLUE benchmark (Wang et al., 2018) and the SICK dataset (Marelli et al., 2014) that provides annotations for relatedness (SICK-REL) and natural language inference (SICK-NLI). As in Devlin et al. (2019), we exclude the WNLI dataset because of the problematic data structure.[4] The average degree of all tasks of these scenarios ranges from 1.1 to 3.2 (as in Table 1).

Cross-Topic Scenario We use two argument similarity datasets, UKP-A (Reimers et al., 2019) and BWS (Thakur et al., 2021). For UKP-A, we binarize the labels into *similar* and *not-similar* as suggested by the authors. Next, we use the evidence dataset from Gleize et al. (2019) that annotates topic, stance, and convincingness for a set of evidence pairs. Apart from the evidence convincingness task (Evi-Conv), we compose a stance prediction task (Evi-St) given evidence and a topic. Further, we use the stance annotations in Gretz et al. (2020) for a second stance prediction task (ArgQ-St) and the dataset provided by Bar-Haim et al. (2020) matching arguments with keypoints (Arg-KP). Finally, we use the argument convincingness dataset (Arg-Conv) by Habernal and Gurevych (2016). All cross-topic tasks are evaluated using multiple folds. We sample these folds on our own except for UKP-A and ArgQ-St - where the authors provide the folds. For all these tasks, we see a more diverse average degree (1.6 to 22.2).

3.2 Training Setup

We fine-tune BERT (Devlin et al., 2019) for the proposed batching strategies and the baseline BASE with random batch sampling. As we earlier described, single training instances can occur in several batches, depending on the batching strategy, Considering NODE every instance occur twice as well as approximately twice for EDGE-I and EDGE-II. In the case of BASE, we saw no sustainable difference of showing training instances once or twice per epoch in preliminary experiments. Eventhough, we want to ensure a fair and comparable setting and thereby include every instance twice for BASE. This is equal as for the NODE strategy and an approximation for EDGE-I and EDGE-II.

Due to computational expenses, we fine-tune the language models for large tasks (QNLI, MNLI-m, MNLI-mm, QQP) over three epochs and the remaining ones for five epochs. For all experiments we use four NVIDIA A4000 GPUs using *PyTorch* v1.8.1, *Huggingface* v4.9.1 (Wolf et al., 2019), and *Sentence-Transformer* v2.0.0 (Reimers and Gurevych, 2019a).

Model Architecture We use for our experiments both bi- and cross-encoder model architecture. *Bi-encoders* showed their computational efficiency for pairwise tasks (Reimers and Gurevych, 2019a) because they encode every distinct sentence separately and use efficient operations (like cosine similarity) to find a prediction. In comparison, *cross-encoders* increase the complexity by encoding every sentence pairs together. To select the pre-trained language model, we distinguish between NLI tasks (SICK-NLI, RTE, QNLI, MNLI) and others. For NLI tasks, we use the standard pretrained weights (i.e. *bert-base-uncased*) since SBERT (Reimers and Gurevych, 2019a) models were trained on NLI data.

The detailed architecture of the models looks as follows. For cross-encoders, we use the standard text pair separators following Devlin et al. (2019). In the case of bi-encoders, we use the cosine similarity of the text pair embeddings fol-

[3] We provide additional details and examples for each task in the Appendix § A.1.

[4] See https://gluebenchmark.com/faq

lowed by the sigmoid function for regression tasks (BWS, SICK-REL). For binary classification (UKP-A, Evi-St, Arg-KP, QQP) tasks, we determine an optimal threshold towards the development set as done by Reimers et al. (2019). For all multi-class tasks (RTE, QNLI, MNLI, SICK-NLI), we use softmax to aggregate both sentence embeddings and their difference as done by Reimers and Gurevych (2019a) and for capturing the annotation direction for tasks with a directed AG (Arg-Conv, Evi-Conv).

Hyperparameters We optimise the batch size for all experiments, strategies, and tasks with zero as random seed and keep the rest of the hyperparameters fixed following previous work (Mosbach et al., 2021; Dodge et al., 2020) (see Appendix § A.2 for details). To compare the different batch sizes, we take the best performing epoch considering the development set. For MNLI, we average the performance of the two development sets (MNLI-m & MNLI-mm). When having multiple folds, we select the optimal batch size according to the average performance overall folds, rather than optimising it separately for each fold.

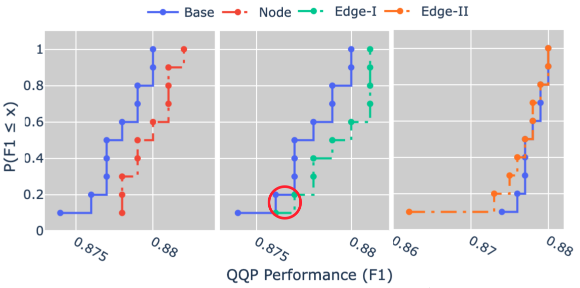

Figure 6: Comparison of the cumulative distribution functions (CDF) of Base with Node, Edge-I, and Edge-II for the QQP task. It shows for a given observation x the probability of observing x or a smaller value in the CDF.

Evaluation Setting We fine-tune every language model with the optimised batch size using ten random seeds, find the final results from the epoch with the highest development score, and report average and std. dev. on the test set. These metrics approximate the underlying results due to the non-gaussian distributed results and an expected performance variance (Dodge et al., 2020). Thus, we test whether an approach outperforms a baseline and vice-versa - i.e. in Figure 6. One option is using the Mann-Whitney U-test (Mann and Whitney, 1947) - also known as Wilcoxon Rank-Sum test (Wilcoxon, 1945; McKnight and Najab, 2010) - which checks whether our approach (i.e. Node) is stochastic larger than the baseline Base (Lehmann, 1955). To match this criterion, the cumulative distribution function (CDF) of the superior approach needs to be consistently below the other one - shown on the left of Figure 6. In Dror et al. (2019), the authors show the sensitivity of the U-test towards minor violations of this requirement. Thus, they proposed *Almost Stochastic Order* (*ASO*) Dror et al. (2019) that better adapts to results of neural networks by slightly allowing some violation ϵ. Such a situation is shown in the middle of Figure 6, where we observe Edge-I outperforming Node but its CDF is not constantly under the other one. Here, the U-test fails ($p < 0.05$) to make a decision due to the minor marked violation while *ASO* can confirm our observation. In contrast, on the right, we see our approach is underperforming the baseline (consistently above the blue line). Using *ASO* we can confirm this observation while the U-test can not gives us a decision ($p < 0.05$).

Since this desired softening of *ASO* increases the risk of type-I errors (i.e., we observe a significant improvement when there is none), we apply a strict test setting compared to other work (Dodge et al., 2020; Zhang et al., 2021). We use a p-value of 0.01 and adapt it with the Bonferroni correction (Bonferroni, 1936) (we provide additional details in the Appendix § B.1). For reference, we also apply the U-test and bootstrap-test (Efron and Tibshirani, 1994) - both with $p < 0.05$ - to test significant improvements and deteriorates apart of *ASO*.

4 Experiments

4.1 In-Domain and Cross-Topic Evaluation

In the first experiment, we evaluate the general effect of using our approach by fine-tuning a BERT bi-encoder. We report the task's mean, standard deviation, and significance with a publicly available test set (Table 2). As the test sets for datasets from the GLUE benchmark are not publicly accessible, we report results based on the development set and the ensemble performance (majority vote) on the test set (set Appendix § B.2 for the test results).[5]

Overall, the results show that Node significantly improves the baseline Base on nine tasks and is never outperformed statistically significant. For Edge-I and Edge-II, we see a statistically significant improvement in eleven and five tasks while being outperformed in zero and

[5] Evaluated with (`https://gluebenchmark.com/`)

four cases, respectively. Considering cross-topic results, we see that EDGE-I performs with a improvement/deterioration ratio of 5/0 better than NODE (4/0) and EDGE-II (2/1). Similar, EDGE-1 performs slightly better than NODE on in-domain tasks (6/0 vs. 5/0) and outperforms EDGE-II (2/3).

4.2 Low-Resource Scenario

In the second experiment, we examine the effect of our approach for the low-resource scenario. Therefore, we iterative select 25, 50, 75, and 100 instances from SICK-NLI, SICK-REL, BWS, and Evi-Conv in a way that these subsets match the average and std. dev. *degree* of the full dataset. In addition, we randomly sample for each subset a control subset (RANDOM) to verify the added value of having structural information within the training samples. These control subsets are trained in the same setting as BASE. We sample four folds for every subset and control subset of the four tasks to get more robust results.

Figure 7 shows the results for all the subsets on the four selected datasets where we see the proposed strategies underperforming BASE on the most subsets. Exceptions are EDGE-II for SICK-REL and NODE for Arg-KP with a ratio of 4/0, as well as the subsets with 25 instances. For them, we observe nine significant improvements and no deteriorations of our strategies in 12 cases, where NODE brings a significant improvement in overall tasks. Considering the control subsets RANDOM, we see that they perform significantly worse than BASE in 12 out of 16 cases.

4.3 Model Agnostic

Next, we want to check whether the success of batching strategy depends on the model type in use. We choose UKP-A, BWS, Evi-St, SICK-NLI, and SICK-REL to cover both scenarios and the overall performance spectrum reported in § 4.1. We compare the bi-encoder (BERT-bi) and cross-encoder (BERT-cross) architecture using BERT. Further, we examine the effect of having more parameters by evaluating BERT-Large in the bi-encoder setting. Finally, we investigate the influence of the model family by comparing BERT with ALBERT (Lan et al., 2020), and RoBERTa (Liu et al., 2019).

Table 3 shows the aggregated results of this experiment. It lists the improvements/deteriorations ratio of all language models and strategies. These results show that EDGE-I outperforms NODE for BERT (1/0 vs. 4/0) while performing on par for BERT-cross (2/0 both). EDGE-II achieves a ratio of 2/1. Looking at BERT-large, we see that NODE and EDGE-I have the same performance (1/2), while EDGE-II underperforms both (0/2). Considering the model family (BERT, ALBERT, RoBERTa), we see for EDGE-I (4/0, 3/2, 3/1) slightly better ratios that for NODE (1/0, 3/1, 3/1), while EDGE-II (2/1, 2/2, 3/2) perform worse. In general, we observe a better improvement/deterioration ratio on BERT (7/1) than on BERT-cross (6/1), RoBERTa (9/4), ALBERT (8/5), and BERT-large (2/6). Considering the strategies, we see an overall ratio of 10/4 for EDGE-I, 13/5 for NODE and 8/8 for EDGE-II.

4.4 Summary

Summarising the experiments shows our approach's significant effect on the performance of different tasks and language models. In detail, we see EDGE-I and NODE significantly improve the performance for a majority of the tasks while EDGE-II causes fewer improvements and all the significant deteriorations. Further, we see slightly better performance on the in-domain tasks than the cross-topics ones. One reason is that finding an optimal batch size is challenging for cross-topics due to the additional regularization coming from having multiple diverse folds. This fact could have a bigger effect on the batching strategies because the batch size has more influence than for BASE.

Considering the low-resource setting, we see the success of our strategies for the extreme case of 25 instances but can not find a clear trend for all the subsets. We see one reason in the face that BASE works similar to our proposed strategies when having a small instance number. In this case, the probability of selecting two instances with a common sentence is higher even with the standard batch sampling procedure. Further, we note that including structural information can provide an added value for the low-resource setting since we see RANDOM constantly underperforming BASE.

Overall, the EDGE-I strategy seems to be slightly superior over NODE, for bi-encoders. We note its significant performance gain on datasets with different task types and its' model agnostic capabilities. For cross-encoder, we see EDGE-I performing similar to NODE but on general with a larger margin. We can imagine that cross-encoders are more sensible when a distinct sentence appears multiple times.

	SICK-NLI	SICK-REL	RTE	QNLI	MNLI-m	MNLI-mm	QQP
BASE	80.3±1.1	88.9±0.2	62.8±2.0	79.8±0.2	73.6±0.2	74.0±0.3	87.8±0.2
NODE	80.6±1.1	**89.1±0.1**$^{(1,3,5)}$	62.6±1.5	80.4±0.1$^{(1,3,5)}$	**74.2±0.4**$^{(1,3,5)}$	**74.4±0.2**$^{(1,3,5)}$	**88.0±0.1**$^{(1,3,5)}$
EDGE-I	**81.2±0.7**$^{(1,5)}$	**89.1±0.1**$^{(1,3,5)}$	62.4±1.6	**80.5±0.2**$^{(1,3,5)}$	74.0±0.2$^{(1,3,5)}$	74.3±0.3$^{(1,3,5)}$	87.9±0.2$^{(1,3,5)}$
EDGE-II	79.2±1.5$^{(2,6)}$	89.0±0.1$^{(1,5)}$	**63.4±0.8**	80.2±0.3$^{(1,3,5)}$	73.8±0.5	73.8±0.4$^{(2)}$	87.6±0.5$^{(2)}$

	UKP-A	BWS	Arg-Conv	Evi-Conv	Evi-St	Arg-KP	ArgQ-St
BASE	**71.4±1.3**	59.5±0.9	81.8±0.3	71.9±0.7	83.8±1.0	72.2±0.9	89.1±0.4
NODE	71.3±0.9	59.8±1.0	**82.1±0.3**$^{(1,3,5)}$	**72.7±0.6**$^{(1,3,5)}$	83.7±1.2	**72.8±0.8**$^{(1)}$	89.2±0.3$^{(1)}$
EDGE-I	**71.4±1.0**	**60.0±0.6**$^{(1)}$	82.0±0.2$^{(1)}$	72.5±0.6$^{(1,3,5)}$	**85.2±1.1**$^{(1,3,5)}$	72.4±0.5	**89.6±0.4**$^{(1,3,5)}$
EDGE-II	71.1±0.7$^{(4)}$	59.7±0.4	81.8±0.4	72.0±0.6	84.9±1.5$^{(1,5)}$	70.1±0.6$^{(2,4,6)}$	**89.6±0.4**$^{(1,3,5)}$

Table 2: Results of BERT bi-encoder using different batching strategies on 14 heterogeneous tasks. Tasks in the upper table are evaluated in-domain, results in the lower part in a cross-topic scenario. We report Pearson Correlation (SICK-REL, RTE), micro-F1 (QQP), and macro-F1 (UKP-A, Evi-St, Arg-KP) as evaluation measures, for all others we report accuracy scores. The best performance for each task is marked in **bold** and statistically significant improvements (*ASO*$^{(1)}$, U-test$^{(3)}$, bootstrap$^{(5)}$) and deteriorations (*ASO*$^{(2)}$, U-test$^{(4)}$, bootstrap$^{(6)}$) are indicated. We find in 17 cases a significant improvement and one deterioration based on all tests. Further, in four and in two cases an improvement or deterioration only based on *ASO*.

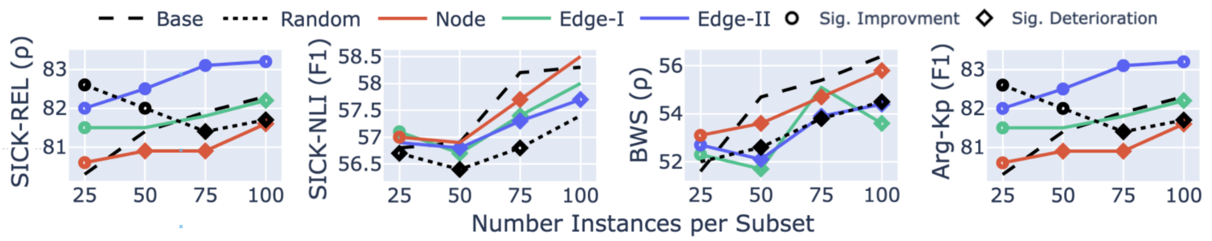

Figure 7: Performance of all strategies, the baseline, and the random sampled control sets SICK-REL, SICK-NLI, BWS, and Arg-KP for 25, 50, 75, and 100 instances (see § B.3 for raw results and details of the subsets). Circles indicates significant improvements and squares deteriorations (using *ASO* with $p < 0.01$).

	Improvement / Deterioration		
Model	NODE	EDGE-I	EDGE-II
BERT-bi	1/0	4/0	2/1
BERT-cross	2/0	2/0	2/1
BERT-large	1/2	1/2	0/2
ALBERT	3/1	3/2	2/2
RoBERTa	3/1	3/1	3/2

Table 3: Overview of statistically significant improvements and deteriorations - using *ASO* - based on § B.4.

5 Further Analysis

Based on the previously shown experiments, we further analyse the influence of the graph structure and the stability & computational complexity of our approach.

5.1 Influence of Graph Structure

We observe for NODE a moderate correlation (0.5) of the selected batch size with the *CV*. We see one reason for this coherence in the fact that having a high *CV* means that there are rare nodes with a high *degree*. Thus, when one of them are sampled in one batch, they can dominate it. Therefore, increasing the batch size can reduce this dominance. For EDGE-I, and EDGE-II we can not observe a notable correlation.

When considering previously showed patterns (Table 1) G_1 and G_2, we observe a slightly better ratio of EDGE-I (7/0 and 2/0) than for NODE (5/0 and 2/0). In case of the third group G_3, we observe similar performance of both (2/0) while EDGE-I outperforms NODE in absolute terms. Compared to NODE, we see EDGE-I better gaining from situations where the *degree* of a few nodes grows extremely ($k > 400$) like in ArgQ-St or Evi-St.

To summarise, we see that the structural patterns influence the training and performance of the different strategies. Thus, we can derive that the batch size of NODE should grow with the *CV*, or that EDGE-I is better suited for tasks where a few nodes have a large degree.

5.2 Stability

Previously work (Dodge et al., 2020; Zhou et al., 2020) identify the training instances' order as a reason for instabilities. Since we adopt this order for our approach, we verify whether the proposed batching strategies lead to additional instabilities. For this purpose, we verify the results of all experiments for a significant difference in the performance variance for every batching strategy compared to the baseline. Using the Brown-Forsyth test (Brown and Forsythe, 1974), we find in 15 out of 165 cases of all experiments a significant ($p < 0.01$) difference in the performance variance, where ten reduced and five raised the variance compared to the baseline. Thus, we conclude that our approach does not introduce new instabilities.

5.3 Computation Complexity

We keep the model size and structure unchanged and thereby do not add any new complexities during inference. For training, the complexity for NODE and BASE is $\mathcal{O}(2n)$ since both process every training instance twice. For EDGE-I and EDGE-II, the complexity depends on the AGs' density. In extreme cases without structure, it is equal to $\mathcal{O}(n)$ because no context edges are sampled, and only base edges are processed. For the other extreme, when every node has at least three connected edges, the complexity is $\mathcal{O}(n + 4n)$ since we sample for every training instance at most four context edges - two for both starting points. In the average case the complexity is approx. $\mathcal{O}(n + 2n(\mu_k - 1))$, since we sample for both starting point of every base edge on average (μ_k - 1) context edges - where μ_k is the average *degree* of all nodes. Note that we subtract one because we already consider, with the based edge, one of the connected edges for both starting points. Thus, NODE and the baseline has a higher complexity than EDGE-I and EDGE-II until μ_k exceeds 1.5.

6 Related Work

While not directly comparable, our work is related to (supervised) contrastive learning in natural language processing (Rethmeier and Augenstein, 2021). Most approaches in this domain (Pagliardini et al., 2018; Logeswaran and Lee, 2018; Giorgi et al., 2021; Gao et al., 2021) aim to learn text representations where related samples (positive pairs) are aligned while unrelated samples (negative pairs) are separated. This self-supervised learning uses training objectives like text reconstruction (Logeswaran and Lee, 2018) or using supervision signals (Conneau et al., 2017; Cer et al., 2018; Reimers and Gurevych, 2019b) from labelled data like Natural Language Inference (NLI) (Bowman et al., 2015; Williams et al., 2018b). In their setup with NLI data, Gao et al. (2021) adapt training batches such that entailment relations are treated as positive examples but contradiction relations and all other in-batch instances as negative examples. Compared to these approaches, our setup focus on the supervised learning setting for these downstream tasks, rather than learning text representations which can be used latter on for these tasks.

In general, our approach adapts the training instance order that a model processes. This idea is also at the core of Curriculum Learning (Bengio et al., 2009) where training instances are reordered according to their estimated difficulty. This has been shown to be beneficial for model performance (Tay et al., 2019; Xu et al., 2020) and faster convergence (Platanios et al., 2019). While Curriculum Learning approaches make use of heuristics to adapt the sample order in one epoch, our approach only relies on the dataset structure to control the composition of training batches.

Dodge et al. (2020) identified that the order of the training samples is a random factor that influences the non-deterministic learning process of neural networks. Further, Zhou et al. (2020) found that inter-instance correlations lead to instabilities during training. We acknowledge these effects and investigate if inter-instance relations can be leveraged in the batch composition to improve the task performance for pairwise text classification.

7 Conclusions

We presented three strategies that adapt the composition of batches to encode structural dataset information. We evaluated these batching strategies on 14 heterogeneous tasks from different domains. Our results confirm the usefulness of this structural information during model training. EDGE-I show the best overall results, including different model types (e.g. ALBERT or RoBERTa) and model architectures (bi- or cross-encoder). Further, we see its success on tasks with extreme characteristics (high degree) and in situations where annotated data is extremely scarce (25 instances). We interpret our results as a promising step to integrate structural dataset information besides instance-level annotations. Further, we encourage future annotation studies to consciously consider including pairs that share common text instances for two reasons. First, to exhaust all possibilities later and second, we showed that even baseline approaches can gain from such structures.

This work covered a broad set of pairwise classification datasets that provide a structure of annotation pairs that share text instances. We plan to employ our method on datasets that do not meet this requirement by inducing inter-instance relations using similarity metrics for future work.

Acknowledgments

The work has received financial support from two sources: (1) the effort of Andreas Waldis is supported by the Hasler Foundation Grant No. 21024, (2) Tilman Beck received support for his work from the German Research Foundation (DFG) as part of the Research Training Group KRITIS No. GRK 2222/2. Further, we thank Luca Mazzola, Leonardo Ribeiro, Kexin Wang, and the anonymous reviewers for their valuable feedback.

References

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020. From arguments to key points: Towards automatic argument summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039, Online. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Carlo Bonferroni. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

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

Morton B Brown and Alan B Forsythe. 1974. Robust tests for the equality of variances. *Journal of the American Statistical Association*, 69(346):364–367.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. 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)*, pages 879–895, Online. Association for Computational Linguistics.

Martin Gleize, Eyal Shnarch, Leshem Choshen, Lena Dankin, Guy Moshkowich, Ranit Aharonov, and Noam Slonim. 2019. Are you convinced? choosing the more convincing evidence with a Siamese network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 967–976, Florence, Italy. Association for Computational Linguistics.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2020. A large-scale dataset for argument quality ranking: Construction and analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7805–7813.

Ivan Habernal and Iryna Gurevych. 2016. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599, Berlin, Germany. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Erich Leo Lehmann. 1955. Ordered families of distributions. *Annals of Mathematical Statistics*, 26:757–777.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Henry B. Mann and Douglas R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Patrick E. McKnight and Julius Najab. 2010. Mann-whitney u test. *Corsini Encyclopedia of Psychology, Vol 1*.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. 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 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019b. 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, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.

Nils Rethmeier and Isabelle Augenstein. 2021. A primer on contrastive pretraining in language processing: Methods, lessons learned and perspectives. *CoRR*, abs/2102.12982.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*, pages 296–310.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 4144–4150. AAAI Press.

Frank. Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics*, 1:196–202.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018a. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018b. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav. Artzi. 2021. Revisiting few-sample bert fine-tuning. *8th International Conference on Learning Representations, ICLR 2021*.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.

A Training Setup

We present additional information on the training setup, including details of the used datasets and hyperparameters.

A.1 Used Datasets

We outline additional detail of the used dataset complementary to § 3.1. Table 9 introduce examples for all used datasets, and Table 10 show additional details for all of them, like the average degree or the number of topics.

A.2 Hyperparameters

The Table 4 and Table 5 shows the evaluated hyperparameters for the different strategies and the used pre-trained language model for the different experiments. This information complements § 3.2

Parameter	**Values**
Batch Size	$\{8, 16, 32\}$ (Base) $\{8, 10, 12, 14\}$ (Node) $\{8, 16, 24, 32\}$ (Edge-I & Edge-II)
Learning Rate	$2e^{-5}$
Optimizer	*AdamW*
Optimizer Function	*Cross-Entropy*
Warmup	10% (linear)

Table 4: Overview of the different used hyperparameters.

	Tasks	bert-base-uncased	bert-large-uncased	albert-base-v2	roberta-base	stsb-bert-base	stsb-bert-large	stsb-roberta-v2	pharaphrase-albert-small-v2
Bi	NLI	1,2,3	3	3	3	-	-	-	-
	Other Tasks	-	-	-	-	1,2,3	3	3	3
Cross	NLI	3	-	-	-	-	-	-	-
	Other Tasks	3	-	-	-	-	-	-	-

Table 5: Overview of the used Huggingface model tags for fine-tuning during the different experiments. (1) refer to the first experiment *In-Domain and Cross-Topic Evaluation*, (2) to *Dataset Size*, and (3) to *Model Agnostic* .

B Additional Results of the Experiments

In this section, we show the additional details of the three Experiments (§ 4.1, § 4.2,§ 4.3).

B.1 Significance Testing Correction

Following the defined significance testing setting (see § 3.2) we use a corrected p-value ($p = 0.01$) for the different experiments. Thus, we divide it by 14 for the first experiment, 8 for the second one, and 6 and 4 for the cross- and in-domain tasks in the third one.

B.2 Experiment: In-Domain and Cross-Topic evaluation

The Table 6 shows the results for a significant improvement ϵ or deterioration ϵ', complementary to § 4.1.

Looking at the GLUE test results (Table 7), we see improvements in absolute numbers for RTE (all strategies), MNLI-m (EDGE-I & II), MNLI-mm (EDGE-II), and QQP (NODE).

B.3 Experiment: Low-Resource Scenario

We show in Table 11 the raw results for all subsets and control subsets (RANDOM) of the SICK-REL, SICK-NLI, BWS, and Evi-St task that we use to compose Figure 7 in § 4.2. The last four columns include results of testing for a significant improvement ϵ or deterioration ϵ'.

B.4 Experiment: Model Agnostic

Table 12 shows the result of the model agnostic experiments in detail. In addition, it lists the re-

		Significance (ϵ / ϵ')		
		NODE	EDGE-I	EDGE-II
SICK-NLI	*in-domain*	1.0/1.0	0.0/1.0	1.0/0.01
SICK-REL		0.0/1.0	0.0/1.0	0.04/1.0
RTE		1.0/1.0	1.0/1.0	1.0/1.0
QNLI		0.0/1.0	0.0/1.0	0.0/1.0
MNLI-m		0.0/1.0	0.0/1.0	1.0/1.0
MNLI-mm		0.0/1.0	0.0/1.0	1.0/0.09
QQP		0.0/1.0	0.0/1.0	1.0/0.04
UKP-A	*cross-topic*	1.0/1.0	1.0/1.0	1.0/1.0
BWS		0.82/1	0.09/1.0	1.0/1.0
Arg-Conv		0.0/1.0	0.23/1.0	1.0/1.0
Evi-Conv		0.0/1.0	0.0/1.0	1.0/1.0
Evi-St		1.0/1.0	0.0/1.0	0.05/1.0
Arg-KP		0.38/1.0	1.0/1.0	1.0/0.0
ArgQ-St		0.19/1.0	0.0/1.0	0.0/1.0

Table 6: Results the significance testing of in- and cross-topic tasks computed with *ASO* with $p < 0.01$.

	BASE	NODE	EDGE-I	EDGE-II
RTE	59.2	59.2	**60.7**	60.2
QNLI	80.7	80.2	80.6	80.1
MNLI-m	33.6	33.7	34.0	33.8
MNLI-mm	75.9	**76.2**	75.3	76.0
QQP	67.0	**67.7**	67.0	66.8

Table 7: Test results on the GLUE tasks. Best results per dataset are marked in **bold**.

	Size	Degree	Random
BWS	25	1.49 ± 0.09	1.03 ± 0.04
	50	1.54 ± 0.03	1.05 ± 0.02
	75	1.56 ± 0.04	1.10 ± 0.04
	100	1.54 ± 0.02	1.12 ± 0.04
SICK-NLI	25	1.88 ± 0.08	1 ± 0
	50	1.77 ± 0.04	1 ± 0
	75	1.78 ± 0.04	1 ± 0
	100	1.78 ± 0.06	1 ± 0
SICK-REL	25	1.88 ± 0.08	1 ± 0
	50	1.77 ± 0.04	1 ± 0
	75	1.78 ± 0.04	1 ± 0
	100	1.78 ± 0.06	1 ± 0
Arg-KP	25	4.01 ± 0.20	1.65 ± 0.11
	50	4.34 ± 0.11	1.93 ± 0.06
	75	4.79 ± 0.11	2.06 ± 0.09
	100	5.03 ± 0.09	2.15 ± 0.08

Table 8: Overview of the average and std. dev. of the degree for all subsets with 25, 50, 75, and 75 samples. Column *Degree* lists the details for the specific sampled subsets, and *Random* the ones for the random sample for the control subsets.

Dataset	Sentence A	Sentence B	Label
BWS	*We shouldn't penalize someone for life.*	*Abortions cause psychological damage.*	0.41
UKP-A	*Cleaner, Greener, Safer, Smarter.*	*The efficiency advantage of electric motors means excellent on-road "fuel" economy.*	Similar
Arg-Conv	*Spam and adware seems to be so much more compatible with IE.*	*If the Firefox is the best then why everybody tries to have IE compatible sites?*	1
Evi-Conv	*The recently independent country of Southern Sudan also recognizes polygamy.*	*A 2011 opinion poll showed that most Malaysians and Indonesians youth opposed polygamy.*	2
Evi-St	*A 2011 opinion poll showed that most Malaysians and Indonesians youth opposed polygamy.*	*We should legalize polygamy*	CON
Arg-KP	*anyone who contributes to ending a life should be punished*	*Assisted suicide is akin to killing someone*	Matching
ArgQ-St	*A majority of americans identify with a religion.*	*We should adopt atheism.*	CON
RTE	*Edward VIII became King in January of 1936 and abdicated in December.*	K*King Edward VIII abdicated in December 1936.*	Entailment
QNLI	*What portion of Berlin's population spoke French by 1700?*	*By 1700, one-fifth of the city's population was French speaking.*	Entailment
MNLI	*Sorry but that's how it is.*	*This is how things are and there are no apologies about it.*	contradiction
QQP	*What was the deadliest battle in history?*	*What was the bloodiest battle in history?*	Duplicated
SICK-REL	*Three kids are sitting in the leaves*	*Three kids are jumping in the leaves*	3.8
SICK-NLI	*Three kids are sitting in the leaves*	*Three kids are jumping in the leaves*	Neutral

Table 9: Examples of the different tasks annotated with the corresponding labels.

sults for the five selected datasets on five language models. The results of testing for a significant improvement ϵ or deterioration ϵ' are shown in the last three columns. These insights complements the aggregated results of Table 3 in the third experiment (§ 4.3).

	Dataset	Label	Pairs	Topics	Degree	Folds	Split	Metric
In-Domain	SICK-NLI*	3-Class	9.954	-	3.2±2.1 (1)	1	4553-495-4906	*acc*
	SICK-REL	Score (1-5)	9.954	-	3.2±2.1 (1)	1	4553-495-4906	ρ
	RTE*	3-Class	4.866	-	1.1±0.6 (1)	1	2.490-277-2.099	*acc*
	QNLI*	Binary	115k	-	1.9±0.8 (1)	1	104k-5463-5463	*acc*
	MNLI-m*	3-Class	413k	-	1.5±0.9 (1)	1	391k-9.714-9796	*acc*
	MNLI-mm*	3-Class	413k	-	1.5±0.9 (1)	1	391k-9832-9847	*acc*
	QQP	Binary	751k	-	1.6±2.2 (1)	1	363k-40k-390k	F_1
Cross-Topic	UKP-A	Binary	3.595	28	3.5±3 (1)	4	17-4-7	F_1 *macro*
	BWS	Score (0-1)	3.400	8	1.6±1.5 (1)	4	5-1-2	ρ
	Arg-Conv	Binary	11.650	32	22.2±4.6 (2)	4	19-5-8	*acc*
	Evi-Conv	Binary	5.697	69	6.2±4.4 (2)	4	46-7-16	*acc*
	Evi-St	Binary	11.394	69	1.9±5.8 (3)	4	46-7-16	F_1 *macro*
	Arg-KP	Binary	24.093	28	7.1±18.1 (3)	4	17-4-7	F_1 *macro*
	ArgQ-St	3-Class	30.497	71	2±20.7 (3)	1	49-7-15	*acc*

Table 10: Summary of the number of folds and the used splits for all used tasks. NLI task are marked with *. The degree is grouped into three pattern-groups: (1) the coefficient of variation (CV) is around one, (2) the CV is below one, and (3) the CV is clearly above one.

		Scores (Bi)					**Significance (ϵ / ϵ')**			
	Size	Base	Random	Node	Edge-I	Edge-II	Random	Node	Edge-I	Edge-II
BWS	25	51.6±1.1	52.0±0.8	**53.1±0.8**$^{(1,3)}$	52.3±0.3$^{(1)}$	52.7±0.2$^{(1,3)}$	0.65/1.0	0.0/1.0	0.27/1.0	0.02/1.0
	50	**54.7±1.1**	52.6±0.8$^{(2,4)}$	53.6±1.2$^{(2,4)}$	51.7±0.7$^{(2,4)}$	52.1±0.7$^{(2,4)}$	1.0/0.0	1.0/0.04	1.0/0.0	1.0/0.0
	75	**55.4±1.0**	53.8±1.3$^{(2,4)}$	54.7±1.1$^{(2)}$	55.1±0.8	53.9±0.8$^{(2,4)}$	1.0/0.0	1.0/0.12	1.0/0.72	1.0/0.0
	100	**56.4±0.8**	54.5±0.8$^{(2,4)}$	55.8±0.8$^{(2)}$	53.6±1.1$^{(2,4)}$	54.4±0.8$^{(2,4)}$	1.0/0.0	1.0/0.23	1.0/0.0	1.0/0.0
SICK-NLI	25	56.8±0.2	56.7±0.5$^{(2)}$	57.0±0.2$^{(1)}$	**57.1±0.1**$^{(1,3)}$	56.9±0.3	1.0/0.43	0.04/1.0	0.01/1.0	1.0/1.0
	50	**56.9±0.1**	56.4±0.8$^{(2,4)}$	**56.9±0.2**	56.7±0.4$^{(2)}$	56.8±0.2$^{(2)}$	1.0/0.03	1.0/1.0	1.0/0.06	1.0/0.05
	75	**58.2**±0.4	56.8±0.4$^{(2,4)}$	57.7±0.4$^{(2,4)}$	57.4±0.5$^{(2,4)}$	57.3±0.4$^{(2,4)}$	1.0/0.0	1.0/0.0	1.0/0.0	1.0/0.0
	100	58.3±0.8	57.4±0.4	**58.5±0.6**	58.0±0.5	57.7±0.5$^{(2,4)}$	1.0/1.0	1.0/1.0	1.0/0.79	1.0/0.0
SICK-REL	25	80.3±0.7	**82.6±0.1**	80.6±0.2$^{(1,3)}$	81.5±0.1$^{(1,3)}$	82.0±0.2$^{(1)}$	1.0/1.0	0.38/1.0	0.0/1.0	0.0/1.0
	50	81.4±0.4	82.0±0.1$^{(1,3)}$	80.9±0.5$^{(2,4)}$	81.5±0.3	**82.5±0.1**$^{(1,3)}$	0.0/1.0	1.0/0.0	0.9/1	0.0/1.0
	75	81.9±0.3	81.4±0.3$^{(2,4)}$	80.9±0.6$^{(2,4)}$	81.8±0.3	**83.1±0.2**$^{(1,3)}$	1.0/0.0	1.0/0.0	1.0/1.0	0.0/1.0
	100	82.3±0.3	81.7±0.4$^{(2,4)}$	81.6±0.4$^{(2,4)}$	82.2±0.3$^{(2)}$	**83.2±0.2**$^{(1,3)}$	1.0/0.0	1.0/0.0	1.0/0.32	0.0/1.0
Arg-KP	25	63.3±0.5	**63.5±0.9**	**63.5±0.4**$^{(1)}$	**63.5±0.7**	63.2±0.4	1.0/1.0	0.29/1.0	0.78/1	1.0/0.86
	50	64.6±0.4	64.0±0.7$^{(2,4)}$	**64.8±0.4**$^{(1)}$	64.5±0.6	64.8±0.5$^{(1)}$	1.0/0.0	0.24/1.0	1.0/1.0	0.13/1.0
	75	64.8±0.5	62.4±0.5$^{(2,4)}$	**65.7±0.4**$^{(1,3)}$	65.4±0.6	65.4±0.5	1.0/0.0	0.0/1.0	1.0/1.0	1.0/1.0
	100	65.6±1.1	62.7±0.4$^{(2,4)}$	**66.1±0.5**$^{(1)}$	**66.1±0.5**$^{(1)}$	66.0±0.4	1.0/0.0	0.25/0.0	0.21/1.0	0.5/1.0

Table 11: Results of the evaluation concerning different dataset sizes for Arg-KP, SICK-REL, and SICK-NLI. The column size indicates for SICK-REL, and SICK-NLI how many training instances are used and for Arg-KP how many topics. For the first four rows pick just a portion of one topic. Statistically significant improvements ($ASO^{(1)}$, U-test$^{(3)}$) and deteriorations ($ASO^{(2)}$, U-test$^{(4)}$) are indicated. The best performance for each task is **bold** marked.

		Strategy				Significance (ϵ / ϵ')		
	Task	BASE	NODE	EDGE-I	EDGE-II	NODE	EDGE-I	EDGE-II
BERT-bi	UKP-A	**71.4±1.3**	71.3±0.9	**71.4±1.0**	71.1±0.7$^{(4)}$	1.0/1.0	1.0/1.0	1.0/1.0
	BWS	59.5±0.9	59.8±1.0	**60.0±0.6**$^{(1)}$	59.7±0.4	0.77/1.0	0.09/1.0	1.0/1.0
	Evi-St	83.8±1.0	83.7±1.2	**85.2±1.1**$^{(1,3)}$	84.9±1.5$^{(1)}$	1.0/1.0	0.0/1.0	0.05/1.0
	SICK-NLI	80.3±1.1	80.6±1.1	**81.2±0.7**$^{(1)}$	79.2±1.5$^{(2)}$	1.0/1.0	0.0/1.0	1.0/0.01
	SICK-REL	88.9±0.2	89.1±0.1$^{(1,3)}$	89.1±0.1$^{(1,3)}$	89.0±0.1$^{(1)}$	0.0/1.0	0.0/1.0	0.04/1.0
BERT-cross	UKP-A	76.0±0.5	76.1±0.8	**76.5±0.7**$^{(1)}$	**76.5±0.5**$^{(1)}$	1.0/1.0	0.02/1.0	0.0/1.0
	BWS	63.6±1.1	64.5±0.5$^{(1,3)}$	63.4±1.8	63.9±1.7$^{(4)}$	0.0/1.0	1.0/1.0	1.0/1.0
	Evi-St	72.5±6.3	72.1±8.1	**76.4±6.9**$^{(1)}$	65.9±7.8$^{(2)}$	1.0/1.0	0.0/1.0	1.0/0.0
	SICK-NLI	86.0±0.7	**86.5±0.9**$^{(1)}$	86.0±0.6	86.3±0.5	0.2/1.0	1.0/1.0	1.0/1.0
	SICK-REL	89.6±0.5	89.5±0.4	**89.8±0.5**	**89.8±0.5**$^{(1)}$	1.0/1.0	0.57/1.0	0.1/1.0
BERT-Large	UKP-A	72.4±0.6	**72.6±0.8**	72.1±1.0$^{(2)}$	71.6±1.3$^{(2)}$	0.79/1.0	1.0/0.39	1.0/0.07
	BWS	**58.6±0.7**	57.3±4.9$^{(2)}$	58.0±4.9$^{(2)}$	56.0±5.9$^{(2)}$	1.0/0.03	1.0/0.19	1.0/0.01
	Evi-St	**87.6±1.2**	85.7±3.3	86.1±2.5	86.9±1.6	1.0/1.0	1.0/1.0	1.0/1.0
	SICK-NLI	79.3±0.7	**80.7±1.3**$^{(1,3)}$	80.5±1.5$^{(1,3)}$	79.6±1.4	0.0/1.0	0.03/1.0	1.0/1.0
	SICK-REL	**89.0±0.3**	88.8±0.3$^{(2)}$	**89.0±0.1**	88.9±0.2	1.0/0.19	0.82/1.0	1.0/1.0
ALBEBRT	UKP-A	**69.9**±0.9	69.5±1.2	69.3±0.9$^{(2)}$	68.5±0.8$^{(2,4)}$	1.0/0.55	1.0/0.11	1.0/0.0
	BWS	57.8±0.2	58.1±0.3$^{(1,3)}$	**58.2±0.4**$^{(1,3)}$	**58.2±0.3**$^{(1,3)}$	0.0/1.0	0.0/1.0	0.0/1.0
	Evi-St	**80.3±2.1**	79.3±3.0$^{(2)}$	79.5±2.4$^{(2)}$	79.4±2.2$^{(2)}$	1.0/0.21	1.0/0.26	1.0/0.29
	SICK-NLI	81.7±2.4	**82.6±0.6**$^{(1)}$	82.3±0.7$^{(1)}$	**82.6±0.7**$^{(1)}$	0.01/1.0	0.08/1.0	0.01/1.0
	SICK-REL	89.2±0.2	**89.5±0.1**$^{(1,3)}$	**89.5±0.1**$^{(1,3)}$	89.3±0.2	0.0/1.0	0.0/1.0	0.84/1.0
RoBERTa	UKP-A	72.4±0.9	73.2±0.5$^{(1,3)}$	73.2±0.7$^{(1,3)}$	**73.5±1.0**$^{(1,3)}$	0.0/1.0	0.01/1.0	0.0/1.0
	BWS	**63.7±0.3**	62.8±0.6$^{(2,4)}$	63.0±0.3$^{(2,4)}$	62.4±0.6$^{(2,4)}$	1.0/0.0	1.0/0.0	1.0/0.0
	Evi-St	88.0±1.6	88.7±2.0	88.6±0.8	**89.4±1.3**$^{(1,3)}$	1.0/1.0	1.0/1.0	0.0/1.0
	SICK-NLI	82.2±0.8	**83.3±0.7**$^{(1,3)}$	82.7±0.5$^{(1,3)}$	82.7±1.0$^{(1)}$	0.0/1.0	0.0/1.0	0.21/1.0
	SICK-REL	89.3±0.1	89.5±0.1$^{(1,3)}$	**89.6±0.2**$^{(1,3)}$	89.3±0.1$^{(2)}$	0.0/1.0	0.09/1.0	1.0/0.12

Table 12: Results of the model agnostic evaluation concerning BERT, BERT-Cross, BERT-Large, ALBEBRT, and RoBERTa on SICK-REL, SICK-NLI, UKP-A, BWS, and Evi-St. Statistically significant improvements (*ASO*$^{(1)}$, U-test$^{(3)}$) and deteriorations (*ASO*$^{(2)}$, U-test$^{(4)}$) are indicated. The best performance for each task is **bold** marked.