Prefer to Classify: Improving Text Classifiers via Auxiliary Preference Learning

Jaehyung Kim [1 2] **Jinwoo Shin** [1] **Dongyeop Kang** [3]

Abstract

The development of largely human-annotated benchmarks has driven the success of deep neural networks in various NLP tasks. To enhance the effectiveness of existing benchmarks, collecting new additional input-output pairs is often too costly and challenging, particularly considering their marginal impact on improving the current model accuracy. Instead, additional or complementary annotations on the existing input texts in the benchmarks can be preferable as an efficient way to pay the additional human cost. In this paper, we investigate task-specific preferences between pairs of input texts as a new alternative way for such auxiliary data annotation. From 'pair-wise' comparisons with respect to the task, the auxiliary preference learning enables the model to learn an additional informative training signal that cannot be captured with 'instance-wise' task labels. To this end, we propose a novel multi-task learning framework, called prefer-to-classify (P2C), which can enjoy the cooperative effect of learning both the given classification task and the auxiliary preferences. Here, we provide three different ways to collect preference signals in practice: (a) *implicitly* extracting from annotation records (for free, but often unavailable), (b) collecting *explicitly* from crowd workers (high paid), or (c) pre-trained large language models such as GPT-3 (low paid). Given existing classification NLP benchmarks, we demonstrate that the proposed auxiliary preference learning via P2C on them is effective in improving text classifiers. Our codes are publicly available.[1]

[1]KAIST [2]Work was mainly done while visiting Minnesota NLP. [3]University of Minnesota. Correspondence to: Jaehyung Kim <jaehyungkim@kaist.ac.kr>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

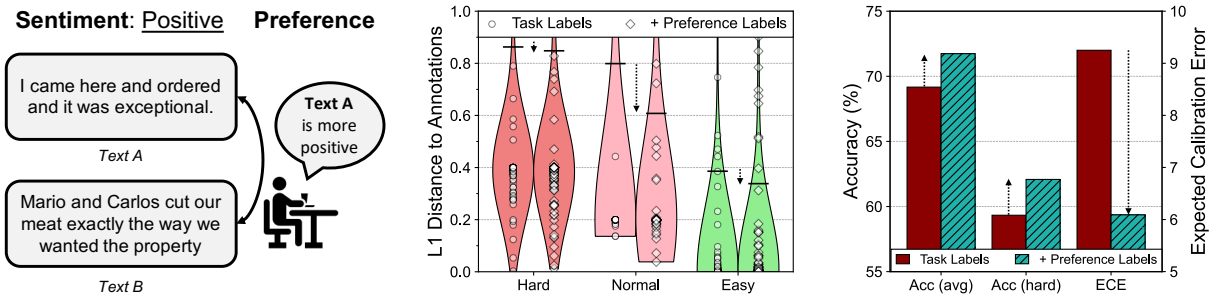
[1]https://github.com/minnesotanlp/p2c

1. Introduction

The recent development of natural language processing (NLP) systems significantly boosts state-of-the-art performances on various NLP tasks (Brown et al., 2020; Ouyang et al., 2022). This success of NLP systems has been driven by, among other things, the construction of largely human-annotated benchmarks, such as GLUE (Wang et al., 2019), SQuAD (Rajpurkar et al., 2016), or BIG-bench (Srivastava et al., 2022). These benchmarks are usually constructed by (a) collecting (or writing) the relevant input texts and (b) assigning output labels by human annotators. Here, (a) is arguably more costly and cumbersome in many practical scenarios; for example, input texts with distribution mismatch or spurious patterns could make the model suffer from learning the generalized representation (Gururangan et al., 2018; Karamcheti et al., 2021), and hence the much higher cost is often paid to the collection process to keep the quality of the constructed benchmark (Kaushik et al., 2020). Therefore, it is often preferable to pay the additional human cost to annotate the existing benchmarks in a complementary way (instead of collecting new input texts), *e.g.*, one can improve the label quality (Nie et al., 2020b; Fornaciari et al., 2021) by assigning multiple annotators to each input or obtain the finer task information with the new label space (Williams et al., 2022). In this paper, we investigate a new alternative way to *better exploit the existing benchmarks (input texts and task labels), with auxiliary annotation* to further improve the model performance.

Contribution. We introduce task-specific preferences between pairs of input texts as a new and auxiliary data annotation, to improve the text classification system upon the existing task annotations (Figure 1(a)). By relatively ordering a pair of two texts and better calibrating them with respect to the task through 'pair-wise' comparison, we expect that the auxiliary preference learning provides an additional informative training signal that cannot be captured with 'instance-wise' evaluation (see Figure 1(b)).

This preference signal could be obtained not only from human annotators (called *subjective* preference), but also from the existing annotation records (called *extractive* preference), if available, and even the recent strong pre-trained language models (called *generative* preference). To be specific, generative preference is obtained by querying the pref-



(a) Pair-wise preference signals (b) Alignment to human annotations (c) Improved text classification

Figure 1. (a) Example of a pair-wise preference signal in the sentiment classification. (b) Auxiliary preference learning makes the classifier capture the fine-grained task information; *e.g.*, predictions of the classifier become more aligned with human annotations. Test samples are divided into Hard, Normal, and Easy based on the annotators' disagreement. (c) Improvement from the collected preference and P2C in various aspects, *e.g.,* better accuracy and calibration. More results are presented in Section 4.4.

erence between two sentences to the recent language models (LMs), *e.g.*, GPT-3 (Brown et al., 2020), with a prompting. Next, extractive preference is constructed from the existing annotation records in datasets 'without additional cost'; if one sample has been less voted than the other, we treat the latter as a relatively higher preference between the two samples. Finally, we collect subjective preferences for 5,000 pairs of texts from (paid) crowd workers by asking them which text is more preferred to the task label.

To utilize both existing class labels and newly obtained preference labels with their cooperative effect, we propose a novel multi-task learning framework, coined *prefer-to-classify* (P2C), to effectively train the model from both classification and preference learning tasks. Specifically, we first introduce diverse multiple preference heads beside the classification head of the model for better learning from preference labels. Then, we introduce a new consistency regularization between classification and preference heads for imposing the model to have higher classification confidence in the preferred samples and hence enabling to detection of the inherent relationship between two tasks. Lastly, we propose two advanced sampling schemes to select more informative text pairs for improving the efficiency of training.

Through the extensive experiments on ten text classification datasets, we demonstrate the effectiveness of our new auxiliary preference learning framework via P2C; for example, P2C with generative preference from GPT-3 exhibited 11.55% relative test error reduction on average compared to the standard training method of the classifier. Next, P2C with extractive preference even outperforms the state-of-the-art methods utilizing annotation records with the 4.27% relative test error reduction. Lastly, the newly-collected subjective preference labels show the largest improvement compared to generative and extractive ones, which reveals the benefit of a more accurate preference signal; it does not only with the improvement in task performance but also with better calibration and task modeling; for example, 6.09% of expected calibration error while 9.19% from the same number of task labels. Overall, our work highlights the effectiveness of preference learning as an auxiliary method to improve the classification system, and we believe our work could inspire researchers to consider a new alternative way for data annotation.

2. Improving Text Classifiers via Auxiliary Preference Learning

In this section, we present prefer-to-classify (P2C), a novel multi-task learning framework to use the preference labels as an auxiliary data annotation for improving the text classifier. The auxiliary preference learning via P2C could provide a new informative training signal that cannot be captured with the existing 'instance-wise' evaluation by relatively ordering a pair of two texts and better calibrating them with respect to the task through 'pair-wise' comparison.

2.1. Preliminaries

Problem description. We describe the problem setup of our interest under a text classification scenario with K classes. Let $\mathcal{D}$ denote the given training dataset consisting of tuples $(\mathbf{x}, y_{\texttt{task}}) \in \mathcal{D}$ where $\mathbf{x} = [x_1, \ldots, x_L]$ is the sequence of input tokens x_i, and $y_{\texttt{task}}$ is the target task label. Our goal is to train a classifier $f_\theta := W_{\texttt{task}} \circ g_\phi$, composed with Transformer-based language model backbone g_ϕ (*e.g.,* BERT (Devlin et al., 2019)) and a random initialized classification head $W_{\texttt{task}}$, to minimize the task-specific loss $\mathcal{L}_{\texttt{task}}$ such as a cross-entropy loss where $p(\mathbf{x}) = \text{Softmax}\big(f_\theta(\mathbf{x})\big)$.

Preference learning. In this paper, we use a preference label between two data instances as an auxiliary learning signal to train the classifier. Specifically, the preference signals reflect the relative suitability between the two input

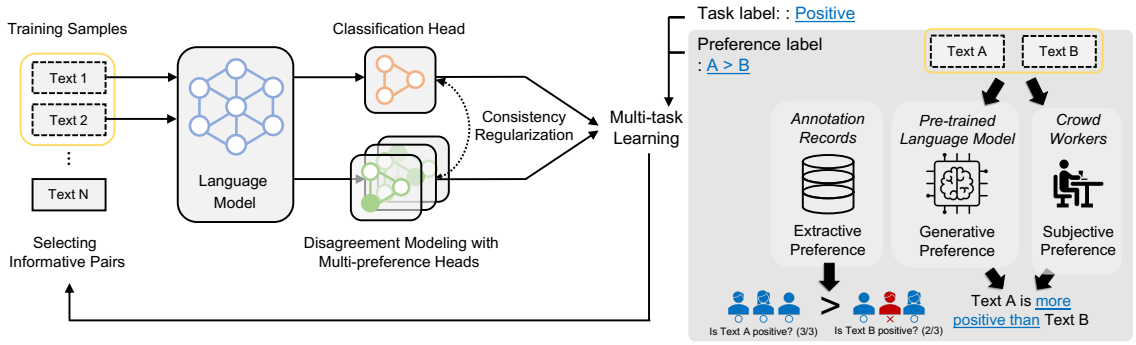


Figure 2. Visual illustration of the proposed auxiliary preference learning for improving the classifier. First, the preference label of the pair of samples is obtained among three different ways (*right*) - *Generative*, *Extractive*, or *Subjective* preference. Then, the preference label is jointly used to train the classifier with the original task label via the proposed **P**refer-**to**-**C**lassify (P2C) framework (*left*).

samples with respect to the given task. We first assume that the preference labels of the given dataset are available. Then, our goal is to train a preference predictor to learn from the given human preferences, by predicting which one among the two input samples is more preferred. To this end, we formulate a preference learning as a supervised learning problem following the approaches in other domains such as reinforcement learning and generative modeling (Christiano et al., 2017; Ziegler et al., 2019; Lee et al., 2021). Given a pair of two different input tokens $(\mathbf{x}^0, \mathbf{x}^1)$ and task label $y_{\texttt{task}}$, a preference label $y_{\texttt{pref}}$ is additionally given; it indicates which input is preferred considering $y_{\texttt{task}}$, *i.e.*, $y_{\texttt{pref}} \in \{0, 1, 0.5\}$, where 1 indicates $\mathbf{x}^1 \succ \mathbf{x}^0$ (*i.e.*, $\mathbf{x}^1$ is preferred than $\mathbf{x}^0$), 0 indicates $\mathbf{x}^0 \succ \mathbf{x}^1$, and 0.5 implies an equally preferable case. Each preference label is stored in a dataset $\mathcal{D}$ as a quadruplet $(\mathbf{x}^0, \mathbf{x}^1, y_{\texttt{task}}, y_{\texttt{pref}})$. Then, we predict a preference using the preference predictor h_ψ following (Bradley & Terry, 1952):

$$P_\psi[\mathbf{x}^1 \succ \mathbf{x}^0; y_{\texttt{task}}] = \frac{\exp\left(h_\psi(\mathbf{x}^1, y_{\texttt{task}})\right)}{\sum_{i\in\{0,1\}} \exp\left(h_\psi(\mathbf{x}^i, y_{\texttt{task}})\right)} \quad (1)$$

where $\mathbf{x}^i \succ \mathbf{x}^j$ implies that input i is preferable to input j. The underlying assumption of this model is that the probability of preferring an input depends exponentially on its output. Then, the preference predictor h_ψ is trained through supervised learning with the given human preferences, by minimizing the binary cross-entropy loss as follow[2]:

$$\mathcal{L}_{\texttt{pref}} = -\mathop{\mathbb{E}}_{\substack{(\mathbf{x}^0,\mathbf{x}^1,y_{\texttt{task}}\\,y_{\texttt{pref}})\sim\mathcal{D}}} \Big[y_{\texttt{pref}} \log P_\psi[\mathbf{x}^1 \succ \mathbf{x}^0; y_{\texttt{task}}] + (1 - y_{\texttt{pref}}) \log P_\psi[\mathbf{x}^0 \succ \mathbf{x}^1; y_{\texttt{task}}] \Big] \quad (2)$$

2.2. Prefer-to-classify (P2C)

Next, we present the specific techniques to train the classifier with given preference labels: (*a*) multi-task learning of classification and preference learning, (*b*) consistency regularization between classification and preference learning, and (*c*) informative pair sampling method based on the disagreement or inconsistency.

Multi-task learning with preference labels. To effectively learn from the given preference label $y_{\texttt{pref}}$ and the task label $y_{\texttt{task}}$, we train the classifier f_θ via multi-task learning (Ruder, 2017; Sener & Koltun, 2018) of both classification and preference learning. Specifically, we model the preference predictor h_ψ in Eq. 1 upon the classifier f_θ similar to the case of $W_{\texttt{task}}$. The preference prediction head $W_{\texttt{pref}}$ is added on the output of Transformer backbone $g_\phi(\mathbf{x})$ and task label $y_{\texttt{task}}$, *i.e.*, $h_\psi(\mathbf{x}, y_{\texttt{task}}) = W_{\texttt{pref}} \circ [g_\phi(\mathbf{x}); y_{\texttt{task}}]$[3] where $f_\theta(\mathbf{x}) = W_{\texttt{task}} \circ g_\phi(\mathbf{x})$.

Preference learning with diverse multi-preference heads. In addition, we introduce multiple preference heads $\{W^{(t)}_{\texttt{pref}}\}_{t=1}^T$ and trained with $\mathcal{L}_{\texttt{pref}}$ in Eq. 2 to fully exploit the given preference labels. As obtaining preference labels requires additional cost, it is crucial to find effective ways to exploit them. By incorporating multiple preference prediction heads, we can obtain diverse learning signals from each preference label, based on their different random initialization (Ganaie et al., 2021). However, these multiple preference heads easily collapse into identical ones, as they are built on the compact representation of the pre-trained Transformer shared with the classification head. Hence, we introduced diversity regularization between $\{W^{(t)}_{\texttt{pref}}\}_{t=1}^T$ during the training; we add a regularization $\mathcal{L}_{\texttt{div}}$ to encourage the diverse prediction for each preference head by maximizing KL-divergence (Wang et al., 2021):

$$\mathcal{L}_{\texttt{div}} = \frac{-1}{T-1} \sum_{j=1, j\neq i}^{T} D_{\texttt{KL}}\big(P_{\psi^{(i)}}(\mathbf{x}^1, \mathbf{x}^0; y_{\texttt{task}}) || P_{\psi^{(j)}}(\mathbf{x}^1, \mathbf{x}^0; y_{\texttt{task}})\big) \quad (3)$$

[2]Equally preferable case is learned with the same coefficients.

[3]$[A; B]$ means the concatenation between A and B.

where $P_\psi(\mathbf{x}^1, \mathbf{x}^0; y_{\texttt{task}})$ is the predictive distribution of the preference predictor h_ψ, *i.e.*, $P_\psi(\mathbf{x}^1, \mathbf{x}^0; y_{\texttt{task}}) = \big[P_\psi[\mathbf{x}^1 \succ \mathbf{x}^0; y_{\texttt{task}}], P_\psi[\mathbf{x}^0 \succ \mathbf{x}^1; y_{\texttt{task}}]\big]$. Overall, we train the classifier with the following multi-task learning objective $\mathcal{L}_{\texttt{multi}}$ under hyper-parameter $\lambda_{\texttt{div}}$:

$$\mathcal{L}_{\texttt{multi}} = \mathcal{L}_{\texttt{task}} + \mathcal{L}^{\text{all}}_{\texttt{pref}} + \lambda_{\texttt{div}}\mathcal{L}_{\texttt{div}} \tag{4}$$

where $\mathcal{L}^{\psi^{(t)}}_{\texttt{pref}}$ indicate the preference learning objective with each head $\psi^{(t)}$ and $\mathcal{L}^{\text{all}}_{\texttt{pref}} = \sum_{t=1}^{T} \mathcal{L}^{\psi^{(t)}}_{\texttt{pref}}$.

Consistency regularization between classification and preference learning. Even though multi-task learning is an effective way to train the model, it is still unclear whether or not the model can capture the relations between the two tasks explicitly. Accordingly, we hypothesize that *a more preferred instance should have higher confidence from the classifier*, *i.e.*, $p_y(\mathbf{x}^1) > p_y(\mathbf{x}^0)$ if $\mathbf{x}^1 \succ \mathbf{x}^0$ with the given task label y. Hence, to impose the model explicitly follows this intuition, we further propose a consistency regularization between the two tasks as follows:

$$\begin{aligned}\mathcal{L}_{\texttt{cons}} = {} & y_{\texttt{pref}} \max\{0, p_y(\mathbf{x}^1) - p_y(\mathbf{x}^0)\} \\ & + (1 - y_{\texttt{pref}}) \max\{0, p_y(\mathbf{x}^0) - p_y(\mathbf{x}^1)\}\end{aligned} \tag{5}$$

Additionally, when the degree of preference is explicitly provided, *i.e.*, $y_{\texttt{pref}} \in [0, 1]$ (see Section 4.3 of extractive preference case) rather than $y_{\texttt{pref}} \in \{0, 1, 0.5\}$, we further extend this consistency regularization with margin m which represents the degree of preference:

$$\begin{aligned}\mathcal{L}_{\texttt{cons}} = {} & y_{\texttt{pref}} \max\{0, m - \Delta p_y(\mathbf{x}^1, \mathbf{x}^0)\} \\ & + (1 - y_{\texttt{pref}}) \max\{0, \Delta p_y(\mathbf{x}^1, \mathbf{x}^0) - m\}\end{aligned} \tag{6}$$

where $\Delta p_y(\mathbf{x}^1, \mathbf{x}^0) = p_y(\mathbf{x}^0) - p_y(\mathbf{x}^1)$. We note that the previous consistency regularization Eq. 5 becomes the special case of Eq. 6 with $m = 0$. Overall, our training loss of the classifier is as follows:

$$\mathcal{L}_{\texttt{train}} = \mathcal{L}_{\texttt{multi}} + \lambda_{\texttt{cons}}\mathcal{L}_{\texttt{cons}} \tag{7}$$

where $\lambda_{\texttt{cons}}$ is a hyper-parameter.

Selecting informative pairs. As the number of pairs of samples $(\mathbf{x}^0, \mathbf{x}^1)$ is proportional to the square of the number of training samples, it is difficult to obtain the preference label for all possible pairs, and even harder to learn from them even if we have all the preference labels. Hence, we propose the following advanced sampling scheme to maximize preference learning's effectiveness during training: (1) *Disagreement-based* sampling, which selects pairs of instances with high variance across multiple preference predictors $\{h_{\psi^{(i)}}\}_{i=1}^{T}$, and (2) *Inconsistency-based* sampling, which seeks to reduce the mismatched pairs with high consistency loss $\mathcal{L}_{\texttt{cons}}$ in Eq. 5. We evaluate the effects of these sampling methods in Appendix C.

3. Collection of Preference Labels

In this section, we present the descriptions of three different types of preference labels (*generative*, *extractive*, and *subjective*) to apply auxiliary preference learning via P2C. The detailed procedure of collecting each preference label and comparison between them is presented in Appendix B.

Generative preference from large language models. First, we propose to use the recent generative pre-trained large language models (Brown et al., 2020; Ouyang et al., 2022) to obtain the preference between pair of samples, and call the obtained preference label as *generative preference*. These models have recently demonstrated the strong zero/few-shot generalization performance in various NLP tasks, and our high-level intuition is that such capability could be effective to provide a useful signal between the samples. To be specific, we use GPT-3 through the officially provided API,[4] by querying the pair of sentences along with the properly designed prompts, presented in Appendix B. For the experiments of the dataset with N samples, we randomly select one pair for each sample and acquire N generative preference labels for P2C.

Extractive preference from data annotation records. Another way is to extract the preference signals from the existing datasets; our high-level assumption is that annotation records of each data, which are naturally gathered during the construction of the dataset, implicitly capture the preference between data samples. For example, if one sample has higher voting (9 out of 10) than the other sample (6 out of 10) as positive sentiment, one can assume that the former has a relatively higher preference. We call this implicit preference label as *extractive preference*; since the extractive preference is derived from existing sources of the dataset, it can be obtained for any pair of samples *without additional cost*. Hence, for the dataset with N samples, one can obtain N^2 of extract preference labels at maximum. We randomly sample the pair of each sample and use their preference labels during training for P2C.

Subjective preference from crowd workers. Lastly, we consider directly collecting the human preference and call it as *subjective preference*; while it requires a high payment to hire human annotators, it is expected to be the most accurate as it is directly obtained by asking humans. Hence, to investigate the advantage of human preference, we construct the subjective preference dataset based on DynaSent-R2 dataset (Potts et al., 2021) for the sentiment classification task. Specifically, we gather the subjective preference of the sentence pairs by asking crowd workers to answer "*which sentence is more positive (neutral, or negative)?*" using Amazon's Mechanical Turk crowd-sourcing platform

[4] `text-davinci-003` in https://beta.openai.com/docs/models/gpt-3

Table 1. Examples of the collected generative, extractive, and subjective preference labels on the same pair of sentences.

A: I got 3 veggies and a side of fries for over a 11 dollars if you like homecooked food	**B**: She listened to my ideas, asked questions to get a better idea about my style, and was excellent at offering advice as if I were a total pleb.
Sentiment: Positive, Generative Preference: **A** $\succ$ **B**, Extractive Preference: **A** $\succ$ **B**, Subjective Preference: **B** $\succ$ **A**	
A: We enjoyed our first and last meal in Toronto at Bombay Palace, and I can't think of a better way to book our journey.	**B**: So glad I finally tried this place because if confirmed my suspicions about that critic who rated it a 10.
Sentiment: Positive, Generative Preference: **A** $\succ$ **B**, Extractive Preference: **B** $\succ$ **A**, Subjective Preference: **No preference**	
A: The buffalo chicken was not good, but very costly.	**B**: There was so much stuff from all over that I had to leave to find an ATM for more cash to pay for it all.
Sentiment: Negative, Generative Preference: **A** $\succ$ **B**, Extractive Preference: **B** $\succ$ **A**, Subjective Preference: **B** $\succ$ **A**	
A: The hotel offered complimentary breakfast.	**B**: My friends had a full acrylic and the other had a fill. It looked so good.
Sentiment: Positive, Generative Preference: **A** $\succ$ **B**, Extractive Preference: **A** $\succ$ **B**, Subjective Preference: **A** $\succ$ **B**	

Table 2. Comparison of three different types of preference labels.

Types	Cost	Accuracy	Accessibility
Generative Preference	Medium	Medium	High
Extractive Preference	Low	Medium	Medium
Subjective Preference	High	High	Low

(Crowston, 2012). Then, each worker should select one of the two sentences or answer "No Preference". Following (Nie et al., 2020a), we hire three crowd workers for each pair of sentences at the most, and the pairs are dynamically selected across multiple rounds to maximize the obtained information. Consequently, we collect a total of 5,000 pairs' subjective preference labels.

As described above, each type of preference has distinct characteristics as shown in Table 2; extractive preference could be freely obtained if the annotation records of the benchmark are available (*i.e.*, lowest cost). On the other hand, generative preference may require an additional cost, but it is not expensive and provides the easiest way to access the preference labels. While subjective preference is the most expensive (e.g., 1.6$ for 10 samples, while 8.0$ for 5,000 samples with GPT-3), it has a clear advantage of providing an accurate and human-aligned preference signal. To verify the effect of those differences, we present qualitative and quantitative examples in Table 1 and Appendix B.

4. Experiments

4.1. Setups

Datasets. For the experiments, we first use the following four text classification datasets: (1) *CoLA* (Warstadt et al., 2019), (2) *SMS Spam* (Almeida et al., 2011), (3) *Hate Speech* (Fišer et al., 2018), and (4) *Emotion* (Saravia et al., 2018). In addition, to demonstrate the effectiveness of extractive preference, we investigate the publicly available datasets providing the annotation records and use the following six text classification datasets. DynaSent (Potts et al., 2021) is a dynamically constructed sentiment classification benchmark with ternary (positive/negative/neutral) sentiments; we use the dataset from the first round, (5) *DynaSent-R1*, and from the second round, (6) *DynaSent-R2*, the dataset for our experiments. Standford politeness corpus (Danescu-Niculescu-Mizil et al., 2013) is a binary classification benchmark for predicting whether the given sentence is polite or impolite. Since there are two different input domains within this benchmark, we split them into two different datasets: (7) *Polite-Wiki* from Wikipedia, and (8) *Polite-SE* from StackExchange, following the original setup. (9) *Offensive* agreement dataset (Leonardelli et al., 2021) is a binary classification benchmark for predicting whether the given sentence is offensive or not. (10) *MultiNLI* (Williams et al., 2018) is a crowd-sourced collection of sentence pairs annotated with textual entailment information; as the only validation set includes the annotation records, we split it into 8:1:1 for training, validation, and test sets. All datasets have the annotation records from 5 annotators for each sample. More details of datasets are presented in Appendix A.1.

Baselines. We first compare the proposed P2C to a naïve training with a cross-entropy loss and majority voted task label, denoted by (a) *Vanilla*. Then, since P2C with extractive preference can be viewed as a new way to utilize the annotation records, we compare this method with a wide range of disagreement learning methods in Section 4.3, as listed as follows; (b) *Soft-labeling* (Fornaciari et al., 2021): using the probabilistic distribution of annotations as soft labels for training; (c) *Margin* (Sharmanska et al., 2016): training the model with hinge loss by setting a margin proportional to the annotators' agreements; (d) *Filtering* (Leonardelli et al., 2021): removing the training samples with a high disagreement. (e) *Weighting* (Uma et al., 2021): using weighted cross-entropy with smaller weights for the samples with high disagreements; (f) *Multi-annotator* (Davani et al., 2022): training the multiple classification heads

Table 3. Test accuracy of fine-tuned RoBERTa classifiers with specified training methods or GPT-3 with few-shot prompting on four different text classification datasets. For P2C, the generative preference labels are obtained from GPT-3. All the values and error bars are mean and standard deviation across 5 random seeds. The best and the second best results are indicated in **bold** and underline, respectively. In the case of few-shot GPT-3, we obtain standard deviation by 3 runs over randomly sampled few-shot examples for prompting.

	CoLA		SMS Spam		Hate Speech		Emotion	
Method	Mcc(↑)	ECE(↓)	bAcc(↑) / wAcc(↑)	ECE(↓)	bAcc(↑) / wAcc(↑)	ECE(↓)	bAcc(↑) / wAcc(↑)	ECE(↓)
Vanilla	63.7±1.0	3.6±1.6	96.9±0.3 / 95.1±1.5	1.3±0.3	81.1±1.8 / 69.9±4.6	5.1±1.0	88.6±2.3 / 76.1±7.8	4.0±1.1
Label Smoothing	63.9±0.3	4.6±1.2	96.9±0.8 / 94.0±1.5	1.1±0.3	81.5±0.9 / 71.3±3.2	6.6±1.0	89.8±0.8 / 76.9±6.6	4.0±0.9
Max Entropy	64.1±0.3	4.5±0.4	96.9±1.1 / 94.7±1.6	1.2±0.3	81.6±1.8 / 70.5±4.2	4.3±0.7	89.1±1.1 / 73.1±2.5	**3.6**±0.9
CS-KD	64.5±1.4	4.1±1.1	96.8±0.9 / 94.0±2.4	**1.1**±0.2	81.4±2.6 / 69.6±5.1	5.3±1.8	89.4±1.6 / 74.0±6.8	4.1±0.2
GPT-3 (0-shot)	60.4	-	90.3 / 84.3	-	68.7 / 41.6	-	50.2 / 23.3	-
GPT-3 (5-shot)	58.5±0.4	-	92.2±0.5 / 88.5±0.7	-	78.5±2.0 / 70.3±3.6	-	46.6±0.6 / 30.3±2.6	-
GPT-3 (20-shot)	58.3±1.4	-	95.8±0.4 / 94.4±0.7	-	77.8±0.5 / 69.0±1.5	-	47.5±1.0 / 30.8±4.5	-
P2C (Ours)	**65.4**±1.0	**2.8**±1.1	**97.4**±0.4 / **95.2**±1.0	**1.1**±0.3	**82.4**±1.3 / **73.6**±4.5	**4.0**±0.3	**90.7**±0.7 / **81.7**±4.7	**3.6**±0.8

for each annotation and using its ensemble for the evaluation. Furthermore, since we train the model with pair of samples, we also consider the baseline considering pair-wise training, (g) Class-wise Self-Knowledge Distillation (*CS-KD*) (Yun et al., 2020), which forces the similar predictive distribution between the same class samples to be similar. Lastly, we consider two regularization methods, (h) *Label Smoothing* (Müller et al., 2019) and (i) *Max Entropy* (Pereyra et al., 2017), as the baselines of P2C with generative preference in Section 4.2. Details are described in Appendix A.2.

Implementation details. All the experiments are conducted by fine-tuning RoBERTa-base (Liu et al., 2019) using Adam optimizer (Kingma & Ba, 2015) with a fixed learning rate 1e-5 and the default hyper-parameters of Adam. For all datasets, the model is fine-tuned using the specified training method with batch size 16 for 20 epochs. In the case of P2C, we use $T = 3$ preference heads $\{W^{(i)}_{\texttt{pref}}\}^{T}_{i=1}$ and 2-layer MLPs for each $W_{\texttt{pref}}$. We choose hyper-parameters from a fixed set of candidates based on the validation set: $\lambda_{\texttt{cons}}, \lambda_{\texttt{div}} \in \{1.0, 0.1\}$. We sample the pair of instances with the same task labels for efficiency. With the extractive preference, we apply the consistency loss with margin (Eq. 6) by using the difference of annotation as the margin m. For other cases, we apply P2C with consistency loss without margin (Eq. 5) on the pre-defined pairs of samples. More details and experimental supports for the design choices can be found in Appendix A.3 and C, respectively.

4.2. Experiments with generative preference

In this section, we first evaluate our framework with the generative preference labels obtained from the pre-trained large language model, GPT-3 (Brown et al., 2020). To validate the effectiveness of P2C under the more challenging scenario, we use the following four datasets which have a skewed label distribution without annotation records: *CoLA*, *SMS Spam*, *Hate Speech*, and *Emotion*. Since their test datasets are also imbalanced, we measure the balanced accuracy (*bAcc*) (Huang et al., 2016) and the worst-group accuracy (*wAcc*) (Sagawa et al., 2020), to evaluate the generalization capability of the model, except CoLA since it is usually used with own metric, Matthews correlation coefficient (*Mcc*) (Chicco & Jurman, 2020). In addition, to measure the calibration of the trained model, we report Expected Calibration Error (*ECE*) (Guo et al., 2017). Here, we commonly adopt the temperature scaling to measure ECE following (Guo et al., 2017). As the annotation records are unavailable, we compare P2C with the baseline methods incurring the smoothed prediction of classifier only using the task label: *Label Smoothing*, *Entropy Maximization*, and *CS-KD*. In addition, we use K-shot prompting predictions of GPT-3 ($K = 0, 5, 20$) as an additional baseline.

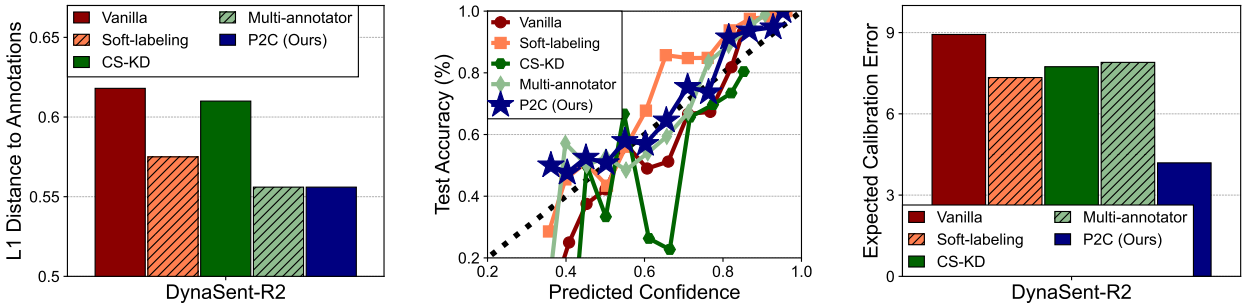
As shown in Table 3, the generative preference labels with P2C are consistently effective in improving the performance of the text classifier; for example, P2C exhibited 11.55% relative test error reduction on average compared to *Vanilla* while also improving the predictive calibration. At the same time, we note that P2C shows better performance than the considered baseline, which indicates that the training signal from the preference label is more than smoothing the prediction of the classifier. Finally, as shown in Table 3, P2C significantly outperforms GPT-3 baselines, which means that our framework does not just distill the 'instance-wise' knowledge of GPT-3, but obtains complementary information through the proposed 'pair-wise' comparisons.

4.3. Experiments with extractive preference

While the generative preference is an efficient way to apply auxiliary preference learning with P2C using the moderate cost, it would be much better if one could still benefit from P2C for free. In this section, we evaluate the effectiveness of P2C with the extractive preference labels, which could be freely obtained from the existing benchmarks if the anno-

Table 4. Test accuracy of fine-tuned RoBERTa classifiers with each annotation method on six different text classification datasets. For P2C, the extractive preference labels are obtained from the annotation records of each dataset. All the values and error bars are mean and standard deviation across five random seeds. The best and the second best results are indicated in **bold** and underline, respectively.

Method	Offensive	Polite-Wiki	Polite-SE	MNLI	DynaSent-R1	DynaSent-R2
Vanilla	75.88±0.72	89.35±1.53	70.00±1.49	81.92±0.70	80.43±0.30	71.23±1.05
Soft-labeling	76.08±1.44	89.57±1.76	70.35±1.68	<u>82.67</u>±0.50	81.10±0.33	<u>72.15</u>±1.59
Margin Loss	<u>76.67</u>±1.18	88.51±0.93	<u>70.51</u>±1.16	81.41±0.63	80.42±0.23	69.27±0.98
Filtering	76.13±1.18	89.50±0.87	68.28±2.43	82.13±0.67	80.38±0.34	69.86±0.78
Weighting	76.17±1.18	89.65±1.46	68.38±1.67	82.48±0.49	80.21±0.41	71.81±1.12
Multi-annotator	76.50±1.98	<u>89.88</u>±1.82	69.39±2.84	82.61±0.70	<u>81.14</u>±0.55	71.97±1.25
CS-KD	75.75±0.66	89.65±1.84	70.10±1.29	82.32±0.23	80.63±0.27	71.81±0.67
P2C (Ours)	**77.81**±0.21	**91.06**±0.64	**71.21**±0.93	**83.15**±0.29	**81.50**±0.39	**73.06**±0.31



(a) Disagreement Prediction (↓) (b) Reliability Diagram (／) (c) Expected Calibration Error (↓)

Figure 3. Additional experimental results of P2C with extractive preference on DynaSent-R2. (a) Average L1 distance between the predictions and the soft labels obtained from the annotation records. The lower distance (↓) means better alignment with annotators. (b) Reliability diagram shows accuracy as a function of confidence. Perfect calibration is plotted by dashed diagonals (／). (c) Expected Calibration Error (ECE) to quantitatively measure the calibration of the classifier. The lower ECE (↓) means better calibration.

tations records are available. We compare P2C with various disagreement learning schemes to fine-tune the RoBERTa-base classifier for each dataset, as they also utilize the annotation records for better training. Table 4 summarizes the results on six text classification datasets. Remarkably, P2C consistently outperforms the baseline methods for all six datasets. To be specific, P2C exhibits 7.59% relative test error reduction compared to the vanilla method in the average. Furthermore, compared to the previous best disagreement learning method of each dataset, P2C exhibits 4.27% relative test error reduction on average. These results show that extractive preferences successfully provide complementary training signals to the classifier from the pair-wise preference, and demonstrate the effectiveness of P2C as a training method to utilize the annotation records.

Next, on DynaSent-R2, we conduct additional experiments to verify how P2C improves the classifier. We first check whether the prediction of the trained model with P2C is similar to the annotators' judgment, as the extractive preference labels come from annotation records. Specifically, we compare the L1 distance between the predictions of the model and the soft labels from the annotation records in Figure 3(a). We verify that P2C achieves the lowest distance to the soft labels, showing the validity of our preference learning for better modeling of the given task. Moreover, we verify that the calibration of the classifier is more improved than the baselines, as a result of pair-wise preference modeling. To be specific, we provide a reliability diagram (Yun et al., 2020), which plots the expected sample accuracy as a function of the confidence of the classifier in Figure 3(b). We remark that the plotted identity function (dashed diagonal) implies perfect calibration (Guo et al., 2017), and our method is the closest one among the baselines. This calibration effect of P2C is further verified through ECE in Figure 3(c).

To validate the effectiveness of the proposed component of P2C in Section 2.2, we perform the ablation experiments, and the results are presented in Table 5, as the extractive preference of all pairs is accessible. It is observable that diverse multi-preference heads improve the effectiveness of preference labels with better modeling compared to the single preference head (*2-4th rows*). In addition, consistency regularization between classification and preference heads enables the classifier to fully utilize the pair-wise

Table 5. Ablation study with each component of P2C with extractive preference labels. Test accuracy of fine-tuned RoBERTa classifiers on DynaSent-R2 and Offensive are compared. All the values and error bars are mean and standard deviation across 5 random seeds.

Method	T	$\mathcal{L}_{\texttt{task}}$	$\mathcal{L}_{\texttt{pref}}$	$\mathcal{L}_{\texttt{div}}$	$\mathcal{L}_{\texttt{cons}}$	Sampling	DynaSent-R2	Offensive
Vanilla	-	✓	-	-	-	-	71.23$_{\pm 1.05}$	75.88$_{\pm 0.72}$
Preference	1	✓	✓	-	-	-	71.84$_{\pm 0.78}$	75.90$_{\pm 1.15}$
	3	✓	✓	-	-	-	71.92$_{\pm 0.66}$	76.43$_{\pm 0.32}$
	3	✓	✓	✓	-	-	72.05$_{\pm 1.30}$	76.67$_{\pm 1.38}$
	3	✓	✓	✓	✓	-	72.67$_{\pm 0.89}$	77.67$_{\pm 0.99}$
P2C (Ours)	3	✓	✓	✓	✓	✓	73.06$_{\pm 0.31}$	77.81$_{\pm 0.21}$

Table 6. Results of fine-tuned RoBERTa classifiers with different ways to obtain the labels on DynaSent-R2. $N_{\texttt{task}}$ and $N_{\texttt{pref}}$ indicate the number of used task labels and preference labels, respectively. $\text{d}_{\texttt{hard}}$ and $\text{d}_{\texttt{easy}}$ are the l_1 distance to annotations with hard and easy samples. Here, the difficulty is defined on the disagreement between annotators. All the values and error bars are mean and standard deviation across 5 random seeds. The best and the second best results are indicated in **bold** and underline, respectively.

Method	$N_{\texttt{task}}$	$N_{\texttt{pref}}$	$\text{Acc}_{\texttt{avg}}(\uparrow)$	$\text{Acc}_{\texttt{hard}}$ / $\text{Acc}_{\texttt{easy}}(\uparrow)$	ECE($\downarrow$)	$\text{d}_{\texttt{hard}}$ / $\text{d}_{\texttt{easy}}(\downarrow)$
Vanilla	7.5k	-	69.03$_{\pm 1.29}$	59.33$_{\pm 2.57}$ / 80.00$_{\pm 1.22}$	9.25$_{\pm 1.39}$	0.856$_{\pm 0.01}$ / 0.405$_{\pm 0.03}$
Task Labels	12.5k	-	71.17$_{\pm 1.35}$	57.86$_{\pm 2.31}$ / **84.21**$_{\pm 1.05}$	9.19$_{\pm 1.36}$	0.878$_{\pm 0.04}$ / **0.327**$_{\pm 0.02}$
Generative Preference	7.5k	5k	71.46$_{\pm 1.16}$	61.77$_{\pm 0.94}$ / 82.28$_{\pm 1.01}$	6.64$_{\pm 0.79}$	0.850$_{\pm 0.02}$ / 0.361$_{\pm 0.02}$
Extractive Preference	7.5k	5k	71.36$_{\pm 1.19}$	61.16$_{\pm 1.91}$ / 83.11$_{\pm 1.78}$	6.75$_{\pm 0.78}$	0.847$_{\pm 0.03}$ / 0.351$_{\pm 0.03}$
Subjective Preference	7.5k	5k	**71.74**$_{\pm 1.04}$	**62.08**$_{\pm 0.94}$ / 83.01$_{\pm 1.27}$	**6.09**$_{\pm 0.31}$	**0.828**$_{\pm 0.02}$ / 0.356$_{\pm 0.02}$

training signal to solve the task, hence the performance is significantly improved (*5th row*). The performance is further improved by selecting the informative pairs during the training (*6th row*). More results are in Appendix C.

4.4. Experiments with subjective preference

Lastly, we verify the effectiveness of collected subjective preference labels compared to other types of labels. To this end, we consider the scenario that the specific types of labels are additionally obtained on top of the existing task labels. Namely, task labels could be collected more with additional training samples, or preference labels between the existing samples could be obtained. Table 6 summarizes the experimental results. Here, it is observed that subjective preference labels are the most effective for improving the test accuracy ($\text{Acc}_{\texttt{avg}}$) along with a better calibration effect. Remarkably, it is noticeable that the preference labels significantly improve the accuracy on relatively hard samples ($\text{Acc}_{\texttt{hard}}$) regardless of its type, while the additional task labels are effective for the relatively easy samples.[5] Somewhat surprisingly, one can observe that the additional 5k generative preference labels by GPT-3 are more effective than the same number of task labels, although the former is much cheaper to obtain than the latter; it indicates that our framework can serve as a new effective way to evolve the existing benchmarks along with the recent development of pre-trained large language models at a considerable cost.

[5]We define the difficulty based on the disagreement of annotators, *i.e.,* more disagree indicates more difficult.

4.5. Applications of P2C beyond text classification

While we primarily demonstrate the effectiveness of P2C on the text classification datasets, our approach has the potential to be applicable beyond text classification. To empirically verify such advantages, we have conducted additional experiments on an image classification task to validate our approach's applicability. Specifically, we used the publicly available SUN Attribute dataset (Patterson et al., 2014) and constructed multiple binary scene attribute classification tasks from it, following the setups in (Sharmanska et al., 2016). Here, we considered the largest five attributes for the experiments. As this dataset includes annotation records, we constructed extractive preference labels to apply P2C. For experiments, we commonly trained ResNet-18 (He et al., 2016) from scratch, for 100 epochs using the SGD optimizer with a weight decay of 0.01 and a learning rate of 0.1 (decreased to 0.01 and 0.001 at 50 and 75 epochs, respectively).

In Table 7, one can observe that our P2C approach effectively improves the performance of the image classifier, with an average relative test error reduction of 6.66% compared to the Vanilla method. These results indicate that the effectiveness of our approach is not only limited to text classification and can be extended to broader applications.

Table 7. Test accuracy of ResNet-18 classifiers on five different binary attribute classification tasks, following the setups in (Sharmanska et al., 2016). For P2C, the extractive preference labels are obtained from the annotation records of each attribute. All the values and error bars are mean and standard deviation across five random seeds. The best results are indicated in **bold**.

Method / Tasks	nat.light	man-made	open	enclosed	nohorizon	Average
Vanilla	78.67$\pm$1.05	62.50$\pm$2.65	79.00$\pm$1.43	85.83$\pm$0.61	76.42$\pm$0.62	76.48$\pm$1.25
P2C (Ours)	**79.83**$\pm$0.94	**64.83**$\pm$1.71	**82.33**$\pm$0.72	**86.25**$\pm$1.43	**77.00**$\pm$1.06	**78.05**$\pm$1.17

5. Related Works

Preference learning. Preference learning is about modeling the preference using a set of instances, associated with an order relation (Fürnkranz & Hüllermeier, 2010). Since it is much easier for humans to make relative judgments, i.e., comparing behaviors as better or worse, preference-based learning becomes an attractive alternative; hence, extensive research has been conducted to address this problem by proposing different techniques to learn from human judgments (Bıyık et al., 2020; Chu & Ghahramani, 2005). One of the most representative fields that adopt preference-based learning is Reinforcement Learning (RL), to learn RL algorithms from the preferences rather than the explicit design of reward function (Wirth et al., 2017). After the successful scale-up of preference-based learning with deep neural networks (Christiano et al., 2017; Lee et al., 2021), this research direction has been extensively explored in other domains such as NLP (Stiennon et al., 2020; Ziegler et al., 2019) and computer vision (Kazemi et al., 2020), especially focused on the generation tasks, *e.g.,* text summarization and image generation. However, preference learning is yet under-explored for classification tasks, despite its great potential to provide complementary and informative training signals via pair-wise comparison of samples.

Learning-to-rank. Preference learning shares a close relationship with learning-to-rank (LTR), a prevalent framework for constructing models or functions to rank objects, as both seek to establish the specific order among samples (Hüllermeier et al., 2008). While preference learning focuses on developing a model to predict preferences between objects, LTR primarily aims to generate ranked lists of items based on their relevance to a given query or context (Fürnkranz & Hüllermeier, 2010). Consequently, LTR has become a key component of various information retrieval problems, such as document retrieval and web search (Burges et al., 2005; Cao et al., 2007). Simultaneously, several works have applied LTR to classification tasks; for instance, (Chang et al., 2020) illustrates the efficacy of LTR for multi-label classification. Furthermore, (Atapour-Abarghouei et al., 2021) transforms classification into LTR and demonstrates its potential for broader classification problems. Compared to these works, our work introduces a novel approach to integrating pairwise comparison for generic classification problems through a multi-task learning framework, accompanied by new methods for obtaining pairwise comparisons between samples.

Auxiliary data annotation. As the development and deployment of NLP systems are directly affected by the quality of benchmarks, various approaches have been recently explored to construct more effective and robust benchmarks. For example, one line of works propose to continuously evolve the benchmark to prevent it becomes obsolete or human-aligned by collecting the adversarial samples of the state-of-the-art models (Nie et al., 2020a; Potts et al., 2021) or incorporating human in the data construction loop (Kiela et al., 2021; Yuan et al., 2021). However, as the collection of new examples is costly, another line of work focuses on finding a better way to annotate the existing benchmarks. For example, some recent works investigate the alternative labeling method rather than a simple majority voting from the annotation records, to avoid sacrificing the valuable nuances embedded in the annotators' assessments and their disagreement (Fornaciari et al., 2021; Leonardelli et al., 2021; Davani et al., 2022). Our work suggests a new alternative way for a better annotation of the existing benchmark via preference between pairs of samples.

6. Conclusion

In this paper, we introduce task-specific preference signals between pairs of samples as a new and auxiliary data annotation to improve the existing text classification system, which relies on instance-wise annotations. To this end, we propose a novel multi-task learning framework, called prefer-to-classify (P2C), to effectively train the classifier from both task and preference labels, and demonstrate this framework under three different types of preference labels.

Acknowledgements. We thank Jongjin Park and Jihoon Tack for providing helpful feedback. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST); No. 2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics) and the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) (NRF-2018R1A5A1059921).

References

Almeida, T. A., Hidalgo, J. M. G., and Yamakami, A. Contributions to the study of sms spam filtering: New collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pp. 259–262, 2011.

Atapour-Abarghouei, A., Bonner, S., and McGough, A. S. Rank over class: The untapped potential of ranking in natural language processing. In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 3950–3959. IEEE, 2021.

Bıyık, E., Huynh, N., Kochenderfer, M. J., and Sadigh, D. Active preference-based gaussian process regression for reward learning. *arXiv preprint arXiv:2005.02575*, 2020.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to rank using gradient descent. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2007.

Chang, W.-C., Yu, H.-F., Zhong, K., Yang, Y., and Dhillon, I. S. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3163–3171, 2020.

Chicco, D. and Jurman, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21: 1–13, 2020.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Chu, W. and Ghahramani, Z. Preference learning with gaussian processes. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

Crowston, K. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the future of ict research. methods and approaches*, pp. 210–221. Springer, 2012.

Danescu-Niculescu-Mizil, C., Sudhof, M., Jurafsky, D., Leskovec, J., and Potts, C. A computational approach to politeness with application to social factors. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013.

Davani, A. M., Díaz, M., and Prabhakaran, V. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics (TACL)*, 10:92–110, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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 (NAACL-HLT)*, 2019.

Fišer, D., Huang, R., Prabhakaran, V., Voigt, R., Waseem, Z., and Wernimont, J. Proceedings of the 2nd workshop on abusive language online (alw2). In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 2018.

Fornaciari, T., Uma, A., Paun, S., Plank, B., Hovy, D., and Poesio, M. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021.

Fürnkranz, J. and Hüllermeier, E. Preference learning and ranking by pairwise comparison. In *Preference learning*, pp. 65–82. Springer, 2010.

Ganaie, M. A., Hu, M., et al. Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*, 2021.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Huang, C., Li, Y., Loy, C. C., and Tang, X. Learning deep representation for imbalanced classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Hüllermeier, E., Fürnkranz, J., Cheng, W., and Brinker, K. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.

Karamcheti, S., Krishna, R., Fei-Fei, L., and Manning, C. D. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Kaushik, D., Hovy, E., and Lipton, Z. C. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations (ICLR)*, 2020.

Kazemi, H., Taherkhani, F., and Nasrabadi, N. Preference-based image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3404–3413, 2020.

Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., et al. Dynabench: Rethinking benchmarking in nlp. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*, 2020.

Lee, K., Smith, L., and Abbeel, P. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Leonardelli, E., Menini, S., Aprosio, A. P., Guerini, M., and Tonelli, S. Agreeing to disagree: Annotating offensive language datasets with annotators' disagreement. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. Adversarial nli: A new benchmark for natural language understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020a.

Nie, Y., Zhou, X., and Bansal, M. What can we learn from collective human opinions on natural language inference data? In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020b.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Patterson, G., Xu, C., Su, H., and Hays, J. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108:59–81, 2014.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., and Hinton, G. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

Potts, C., Wu, Z., Geiger, A., and Kiela, D. Dynasent: A dynamic benchmark for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Saravia, E., Liu, H.-C. T., Huang, Y.-H., Wu, J., and Chen, Y.-S. Carer: Contextualized affect representations for emotion recognition. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Sharmanska, V., Hernández-Lobato, D., Miguel Hernandez-Lobato, J., and Quadrianto, N. Ambiguity helps: Classification with disagreements in crowdsourced annotations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Uma, A. N., Fornaciari, T., Hovy, D., Paun, S., Plank, B., and Poesio, M. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470, 2021.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2019.

Wang, X., Lian, L., Miao, Z., Liu, Z., and Yu, S. X. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations (ICLR)*, 2021.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, 2018.

Williams, A., Thrush, T., and Kiela, D. Anlizing the adversarial natural language inference dataset. In *Proceedings of the Society for Computation in Linguistics 2022*, 2022.

Wirth, C., Akrour, R., Neumann, G., Fürnkranz, J., et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136): 1–46, 2017.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010.

Yuan, A., Ippolito, D., Nikolaev, V., Callison-Burch, C., Coenen, A., and Gehrmann, S. Synthbio: A case study in human-ai collaborative curation of text datasets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yun, S., Park, J., Lee, K., and Shin, J. Regularizing class-wise predictions via self-knowledge distillation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Appendix

Prefer to Classify: Improving Text Classifiers via Auxiliary Preference Learning

A. Experimental Details

A.1. Datasets

As described in Section 4.1, we extensively demonstrate the effectiveness of P2C on the multiple classification datasets; 10 text classification datasets and 1 image classification dataset. For text data, we commonly set the maximum length L as 256 for the tokenization of given data. First, we use the following four text classification datasets in Section 4.2, which do not release the annotation records; hence, the extractive preference labels are not available:

CoLA (Warstadt et al., 2019) is a binary single sentence classification task, where the goal is to predict whether the given English sentence is linguistically valid or not. It is composed of 8.5k training samples and 1k development samples. We remark that the CoLA dataset is part of the popular benchmark, GLUE (Wang et al., 2019), and the dataset is officially available at https://huggingface.co/datasets/glue.

SMS Spam (Almeida et al., 2011) is a public set of SMS-labeled messages that have been collected for mobile phone spam research. It has one collection composed of 5,574 English, real and non-encoded messages, tagged according to being legitimate (ham) or spam. We split the dataset into an 8:1:1 ratio to construct training, validation, and test datasets. SMS Spam is officially available at https://huggingface.co/datasets/sms_spam.

Hate Speech (Fišer et al., 2018) is constructed by extracting the texts from Stormfront, a white supremacist forum. A random set of forum posts have been sampled from several subforums and split into sentences. Those sentences have been manually labeled as containing hate speech or not, according to certain annotation guidelines. Overall, it is composed of 10,703 sentences. We split the dataset into an 8:1:1 ratio to construct training, validation, and test datasets. Hate Speech is officially available at https://huggingface.co/datasets/hate_speech18.

Emotion (Saravia et al., 2018) is a dataset of English Twitter messages with six basic emotions: sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5). In the given Emotion dataset, there are 16,000 training, 2,000 validation, and 2,000 test samples. We use the Emotion dataset at https://huggingface.co/datasets/dair-ai/emotion.

Next, we use the following six text classification datasets, obtained from the following three different sources, which release the annotation records during the construction of the datasets. Here, all datasets have the annotation records of 5 different annotators for each data; however, the annotators can be different among data, *i.e.*, there are more than 5 annotators overall.

DynaSent (Potts et al., 2021) is a sentiment classification benchmark with ternary (positive/negative/neutral) sentiments. It is dynamically constructed through multiple iterations of training a classifier model and finding its adversarial samples by involving a human annotator in the loop. In our experiments, we use the dataset from the first round, *DynaSent-R1*, and the dataset from the second round, *DynaSent-R2*. DynaSent-R1 comprises 80,488 training samples, 3,600 validation samples, and 3,600 test samples, respectively. DynaSent-R2 comprises 13,065 training samples, 720 validation samples, and 720 test samples. All the validation and test samples are fully balanced between the three classes. DynaSent dataset and more details of the dataset are officially available at https://github.com/cgpotts/dynasent.

Standford politeness corpus (Danescu-Niculescu-Mizil et al., 2013) is a binary classification benchmark for predicting whether the given sentence is polite or impolite. Since there are two different input domains within this benchmark, we split them into two different datasets: *Polite-Wiki* from Wikipedia, and *Polite-SE* from Stack Exchange, following the original paper (Danescu-Niculescu-Mizil et al., 2013). Here, two classes: polite and impolite, are defined as the top and, respectively, bottom quartile of sentences when sorted by their politeness score. The classes are therefore balanced, with each class consisting of 1,089 samples for the Wikipedia domain and 1,651 samples for the Stack Exchange domain. We split each dataset into an 8:1:1 ratio to construct training, validation, and test datasets. The source data and more details of the dataset are officially available at https://www.cs.cornell.edu/~cristian/Politeness.html.

Offensive agreement dataset (Leonardelli et al., 2021) is a binary classification benchmark for predicting whether the given sentence is offensive or not. Each sentence is collected from Twitter using Twitter public APIs, based on the hashtags and keywords on three different domains: Covid-19, US Presidential elections and the Black Lives Matter (BLM) movement. Remarkably, some of the original samples are not available anymore due to the elimination of tweets from the user side; for

example, 10,735 samples are collected initially (Leonardelli et al., 2021), but only 6,513 samples are now available. To address the issue of the reduced number of samples, we slightly modify the dataset to keep the setups of the original paper, *e.g.*, balanced among the classes and domains. Specifically, we gather the given splits of the dataset into the unified one and then re-split it as much be balanced as possible. This re-constructed dataset has 2,400 training samples, 400 validation samples, and 400 test samples. Also, the ratio between Covid-19, Election, and BLM is 3:3:2. The dataset is officially available with the request to authors at https://github.com/dhfbk/annotators-agreement-dataset.

Multi-Genre Natural Language Inference (MultiNLI) (Sener & Koltun, 2018) is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information: for a given premise sentence, one should classify whether the given hypothesis sentence is *entailment*, *neutral*, or *contradiction* to the premise (ternary classification). Since the annotation records are only available with the validation set, we construct the datasets by splitting it into 8:1:1 for training, validation, and test sets. This re-constructed dataset has 15,717 training samples, 1,964 validation samples, and 1,966 test samples. The source data and more details of the dataset are officially available at https://cims.nyu.edu/~sbowman/multinli.

Finally, to demonstrate the applicability of P2C beyond NLP tasks, we use the SUN Attribute dataset (Patterson et al., 2014):

SUN Attribute dataset (Patterson et al., 2014) is constructed by conducting Amazon's Mechanical Turk crowd-sourcing platform (Crowston, 2012) to annotate the presence of the target attribute in the given image. The dataset consists of 14,340 scene images from the SUN dataset (Xiao et al., 2010), which has 102 scene attributes such as sunny, natural, man-made, etc. In this dataset, the presence of the attribute is measured as an average score of three binary user responses, *i.e.*, it contains the annotation records and hence we use it to construct the extractive preference for our framework. The source data and more details of the dataset are officially available at https://cs.brown.edu/~gmpatter/sunattributes.html.

A.2. Baselines

We first introduce some notations for a clear explanation. For each sample $\mathbf{x}$, there are annotation records $n_y(\mathbf{x}) \in \mathbb{N}^K$ where K is the number of class and $n_{\texttt{vote}}(\mathbf{x}) = \sum_y n_y(\mathbf{x})$ is the number of votes[6]. Then, the majority voted target label is obtained by finding the most agreed labels, *i.e.*, $y_{\texttt{task}}(\mathbf{x}) = \arg\max_y n_{\texttt{vote}}(\mathbf{x})$, and simply denoted by $y_{\texttt{task}}$. Here, our goal is to train a classifier $f_\theta := W_{\texttt{task}} \circ g_\phi$, composed with Transformer-based language model backbone g_ϕ (*e.g.,* BERT (Devlin et al., 2019)) and a random initialized classification head $W_{\texttt{task}}$, where the prediction for $\mathbf{x}$ is obtained with softmax, *i.e.*, $p(\mathbf{x}) = \texttt{Softmax}\big(f_\theta(\mathbf{x})\big)$. For the analysis in Figure 3, we only include four baselines with high performance based on the results in Table 4.

Vanilla: as described in Section 2.1, the model f_θ is trained with the following cross-entropy loss:

$$\mathcal{L}_{\texttt{train}} = \ell_{\texttt{xe}}(p(\mathbf{x}), y_{\texttt{task}})$$

Soft-labeling (Fornaciari et al., 2021): instead of using majority voted label $y_{\texttt{task}}$, it use the soft-labels $q(\mathbf{x}) = n_y(\mathbf{x})/n_{\texttt{vote}}(\mathbf{x})$ with a cross entropy loss:

$$\mathcal{L}_{\texttt{train}} = \ell_{\texttt{xe}}(p(\mathbf{x}), q(\mathbf{x})) = \sum_y -q_y(\mathbf{x}) \log p_y(\mathbf{x})$$

Margin Loss (Sharmanska et al., 2016): instead of using majority voted label and cross-entropy loss, it uses the soft-labels $q(\mathbf{x})$ as a margin for the multi-class hinge loss:

$$\mathcal{L}_{\texttt{train}} = \sum_y \max\{0, q_y(\mathbf{x}) - p_y(\mathbf{x})\}$$

Filtering (Leonardelli et al., 2021): following the setups in the original paper, we exclude the ambiguous samples that have a low agreement between the annotators. Specifically, we exclude the samples with $n_{y_{\texttt{task}}} = 3$ since there are 5 annotators for all considered datasets by following (Leonardelli et al., 2021), and use majority voting for the others.

[6]All the used datasets commonly have $n_{\texttt{vote}} = 5$

$$\mathcal{L}_{\texttt{train}} = \mathbb{1}[n_{y_{\texttt{task}}}(\boldsymbol{x}) > 3]\ \ell_{\texttt{xe}}(p(\mathbf{x}), y_{\texttt{task}})$$

Weighting (Uma et al., 2021): using weighted cross entropy that down-weigh the samples with a low agreement:

$$\mathcal{L}_{\texttt{train}} = \mathbf{w}(\mathbf{x})\ \ell_{\texttt{xe}}(p(\mathbf{x}), y_{\texttt{task}})$$

where $\mathbf{w}(\mathbf{x}) = n_{y_{\texttt{task}}}(\mathbf{x})/n_{\texttt{vote}}(\mathbf{x})$.

Multi-annotator (Davani et al., 2022): instead of aggregating the different annotators' annotation records, it introduces multiple classification heads $W^{(t)}_{\texttt{task}}$ for learning from each annotator's annotation $y^{(t)}_{\texttt{task}}$. Since each annotator does not annotate all the samples, we simply separate the $n_{\texttt{vote}}(\mathbf{x})$ annotations and train each classification head where $t = 1, \ldots, n_{\texttt{vote}}$. For the inference of test samples, the ensemble of multiple classification heads is used.

$$\mathcal{L}_{\texttt{train}} = \frac{1}{n_{\texttt{vote}}(\mathbf{x})} \sum_{t}\ \ell_{\texttt{xe}}(p^{(t)}(\mathbf{x}), y^{(t)}_{\texttt{task}})$$

where $p^{(t)}(\mathbf{x}) = W^{(t)}_{\texttt{task}} \circ g_\phi(\mathbf{x})$.

CS-KD (Yun et al., 2020): for each sample $\mathbf{x}$, the sample $\widehat{\mathbf{x}}$ within the same class, defined by a majority voted label $y_{\texttt{task}}$, is also sample and the consistency regularization is additionally imposed between their prediction with a temperature τ. Following the original paper, we use $\tau = 4$.

$$\mathcal{L}_{\texttt{train}} = \ell_{\texttt{xe}}(p(\mathbf{x}), y_{\texttt{task}}) + \ell_{\texttt{xe}}(\widetilde{p}(\mathbf{x}), \widetilde{p}(\widehat{\mathbf{x}}))$$

where $\widetilde{p}(\mathbf{x}) = \texttt{Softmax}(f_\theta(\mathbf{x})/\tau)$.

Label Smoothing (Müller et al., 2019): instead of directly using majority voted label $y_{\texttt{task}}$, it first constructs the soft-label $q(\mathbf{x})$ by subtracting τ for the class $y_{\texttt{task}}$ and equally distributing it to remaining classes, *i.e.*, $\tau/(K-1)$. We find the best hyper-parameter τ among $[0.05, 0.1, 0.15]$ using the validation set. Then, this soft label $q(\mathbf{x})$ is used to train the model with a cross-entropy loss:

$$\mathcal{L}_{\texttt{train}} = \ell_{\texttt{xe}}(p(\mathbf{x}), q(\mathbf{x})) = \sum_{y} -q_y(\mathbf{x}) \log p_y(\mathbf{x})$$

Max Entropy (Pereyra et al., 2017): in addition to the cross-entropy loss with the majority voted label $y_{\texttt{task}}$, the regularization loss to increase the entropy of the prediction $p(\mathbf{x})$ is used as a training loss with a hyper-parameter λ. λ is tuned among $[0.1, 0.5, 1.0]$ using the validation set:

$$\mathcal{L}_{\texttt{train}} = \ell_{\texttt{xe}}(p(\mathbf{x}), y_{\texttt{task}}) + \lambda \sum_{y} p_y(\mathbf{x}) \log p_y(\mathbf{x})$$

A.3. Prefer-to-Classify (P2C)

In this section, we describe the details of P2C. We first note that the details are slightly different between extractive preference learning (Section 4.3) and subjective preference learning (Section 4.4) due to the difference in experimental setups between them. As described in Section 4, we commonly use $T = 3$ preference heads $\{W^{(i)}_{\texttt{pref}}\}_{i=1}^{T}$ and 2-layer MLPs with `tanh` activation for each $W_{\texttt{pref}}$. We choose hyper-parameters from a fixed set of candidates based on the validation set; $\lambda_{\texttt{pref}}, \lambda_{\texttt{div}} \in \{1.0, 0.1\}$. Also, we only sample the pair of instances with the same majority voted labels for efficiency.

In the case of learning with extractive preference in Section 4.3, we apply the consistency regularization with margin (Eq. 6) by using the difference of annotation as the margin m. Specifically, we set a margin of class y between two samples $\mathbf{x}^1$ and $\mathbf{x}^0$ as the difference of their soft-labels $m_y = q_y(\mathbf{x}^1) - q_y(\mathbf{x}^0)$, defined in Section A.2. Then, we apply the consistency regularization to all classes $y \in [0, 1]^K$. In addition, we apply the inconsistency-based sampling for the experiments with extractive preference labels based on the superior experimental results, presented in Section C.

Algorithm 1 `Prefer-to-Classify` (`P2C`) `with extractive preference labels`

Input: Classifier from a pre-trained language model f_θ, training dataset $\mathcal{D}$ with preference labels $\{(\mathbf{x}^0, \mathbf{x}^1, y_{\texttt{task}}, y_{\texttt{pref}}) | \mathbf{x}^0, \mathbf{x}^1 \in \mathcal{D}\}$, preference predictors $\{h_{\psi^{(t)}}\}_{t=1}^T$, mini-batch size B, and hyper-parameter $\lambda_{\texttt{cons}}$

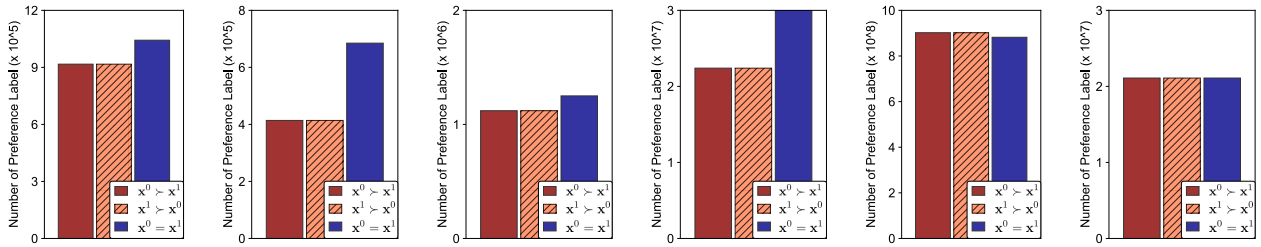
for each iteration **do**
  Draw a mini-batch $\mathcal{B} = \{(\mathbf{x}_i, y_{\texttt{task},i})_{i=1}^B\}$ and the corresponding pairs with preference labels $\widetilde{\mathcal{B}} = \{(\widetilde{\mathbf{x}}_i, y_{\texttt{pref},i})_{i=1}^B\}$ from $\mathcal{D}$ with the inconsistency-based sampling (see Section 2.2)
  Obtain $f_\theta(\mathbf{x})$ by forwarding $\mathcal{B}$, then calculate $\mathcal{L}_{\texttt{multi}}$ in Eq. 4
  Obtain $h_\psi(\mathbf{x})$ by forwarding $\mathcal{B}$ and $\widetilde{\mathcal{B}}$, then calculate $\mathcal{L}_{\texttt{cons}}$ in Eq. 6
  Update parameters θ and $\psi^{(t)}$ to minimize $\mathcal{L}_{\texttt{train}} = \mathcal{L}_{\texttt{multi}} + \lambda_{\texttt{cons}}\mathcal{L}_{\texttt{cons}}$
end for

Algorithm 2 `Prefer-to-Classify` (`P2C`) `with subjective/generative preference labels`

Input: Classifier from a pre-trained language model f_θ, original training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$, collected dataset $\widetilde{\mathcal{D}}$ with preference labels $\{(\mathbf{x}^0, \mathbf{x}^1, y_{\texttt{task}}, y_{\texttt{pref}}) | \mathbf{x}^0, \mathbf{x}^1 \in \mathcal{D}\}$ where $|\widetilde{\mathcal{D}}| = N_{\texttt{pref}}$, preference predictors $\{h_{\psi^{(t)}}\}_{t=1}^T$, a mini-batch size B and hyper-parameter $\lambda_{\texttt{cons}}$

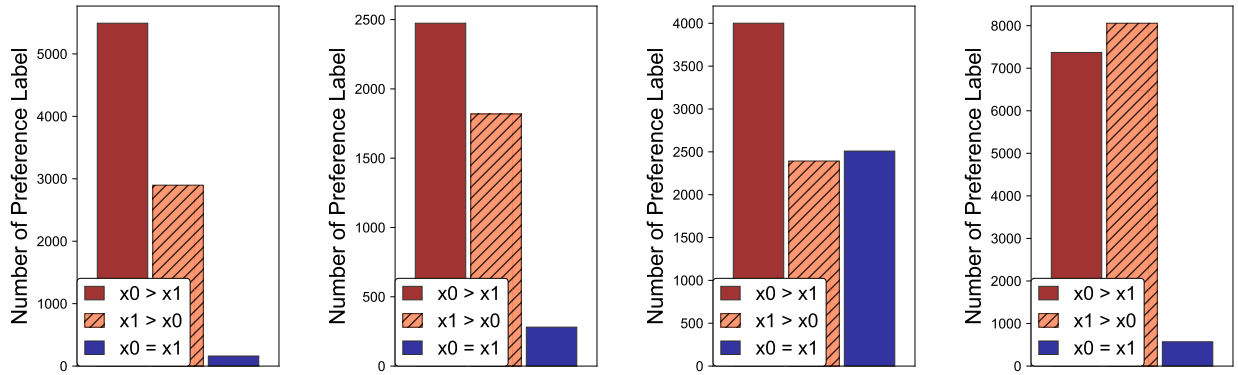
for each iteration **do**
  Draw a mini-batch $\mathcal{B} = \{(\mathbf{x}_i, y_{\texttt{task},i})_{i=1}^B\}$ from $\mathcal{D}$
  Draw an another mini-batch $\widetilde{\mathcal{B}} = \{(\mathbf{x}_i, \widetilde{\mathbf{x}}_i, y_{\texttt{task},i}, y_{\texttt{pref},i})_{i=1}^B\}$ from $\widetilde{\mathcal{D}}$
  Obtain $f_\theta(\mathbf{x})$ by forwarding $\mathcal{B}$, then calculate $\mathcal{L}_{\texttt{multi}}$ in Eq. 4
  Obtain $h_\psi(\mathbf{x})$ by forwarding $\widetilde{\mathcal{B}}$, then calculate $\mathcal{L}_{\texttt{cons}}$ in Eq. 5
  Update parameters θ and $\psi^{(t)}$ to minimize $\mathcal{L}_{\texttt{train}} = \mathcal{L}_{\texttt{multi}} + \lambda_{\texttt{cons}}\mathcal{L}_{\texttt{cons}}$
end for

In the case of learning with subjective and generative preference labels in Section 4.4 and 4.2, we apply the consistency regularization without margin (Eq. 5) since the explicit degree of preference is not given. Also, since the number of pairs with subjective preference labels is limited, we use all of them in training without applying the sampling methods described in Section 2.2. We introduce the additional mini-batch from these pairs to optimize the model with consistency regularization. The full procedures of P2C with extractive and subjective preference are described in Algorithm 1 and 2, respectively.



(a) Offensive (b) Polite-Wiki (c) Polite-SE (d) MultiNLI (e) DynaSent-R1 (f) DynaSent-R2

Figure 4. Distribution of the extractive preference labels from the annotation records.



(a) CoLA (b) SMS Spam (c) Hate Speech (d) Emotion

Figure 5. Distribution of the generative preference labels obtained by querying to GPT-3.

B. Collection of Preference Labels

B.1. More details of preference labels

Extractive preference. For a formal description of the process of collecting extractive preference, we borrow some notations introduced in Section A.2. As described in Section 3, we obtain the extractive preference label $y_{\texttt{pref}}$ by comparing the number of votes $n_{y_{\texttt{task}}}(\mathbf{x})$ with the given task label $y_{\texttt{task}}$: if $n_{y_{\texttt{task}}}(\mathbf{x}^1) > n_{y_{\texttt{task}}}(\mathbf{x}^0)$, then we assign $y_{\texttt{pref}} = 1$ where it indicates $\mathbf{x}^1 \succ \mathbf{x}^0$. Similarly, we assign $y_{\texttt{pref}} = 0$ when $n_{y_{\texttt{task}}}(\mathbf{x}^1) < n_{y_{\texttt{task}}}(\mathbf{x}^0)$ and $y_{\texttt{pref}} = 0.5$ when $n_{y_{\texttt{task}}}(\mathbf{x}^1) = n_{y_{\texttt{task}}}(\mathbf{x}^0)$, respectively. To reduce the noisy signal and focus on the effective pair, we only compare the samples that have the same majority voted labels, *i.e.*, $y_{\texttt{task}}(\mathbf{x}^1) = y_{\texttt{task}}(\mathbf{x}^0)$. The resulting distribution of extractive preference labels for each data is presented in Figure 4.

Generative preference. As we denote in Section 3, we collect the generative preference labels by querying the pair of samples to the recent large pre-trained language model, GPT-3 (Brown et al., 2020). Specifically, we use the officially provided API[7]. To this end, we design our prompt as Figure 6; for ith pair of sentences, we provide two sentences along with their task labels. The resulting distribution of generative preference labels for each data is presented in Figure 5.

Subjective preference. We collect the subjective preference labels based on paired samples from DynaSent-R2 dataset (Potts et al., 2021) for the sentiment classification task. To be specific, we gather the subjective preference of the pairs by asking crowd workers to answer "*which sentence is more positive (neutral, or negative)?*" using Amazon's Mechanical Turk crowd-sourcing platform (Crowston, 2012). Then, each worker should select one of the two sentences or answer "No Preference". Following (Nie et al., 2020a), we initially provide each pair of sentences to two crowd workers. If two workers give the same preference label, this pair is labeled with that. If they disagree, we ask a third crowd worker to break the tie. If

[7] `text-davinci-003` in https://beta.openai.com/docs/models/gpt-3

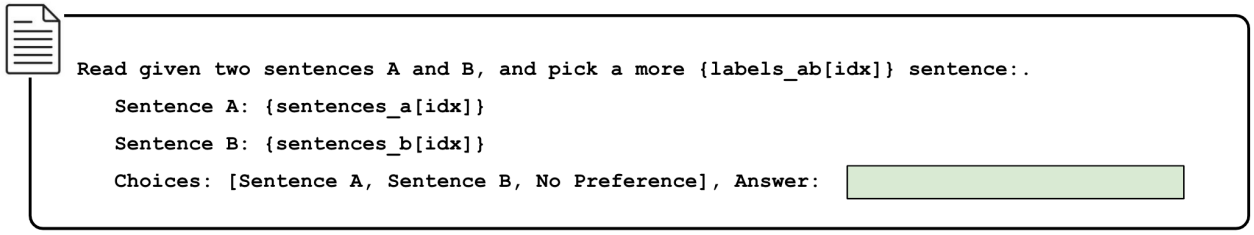


Figure 6. Prompt design to collect generative preference labels from GPT-3 (Brown et al., 2020).

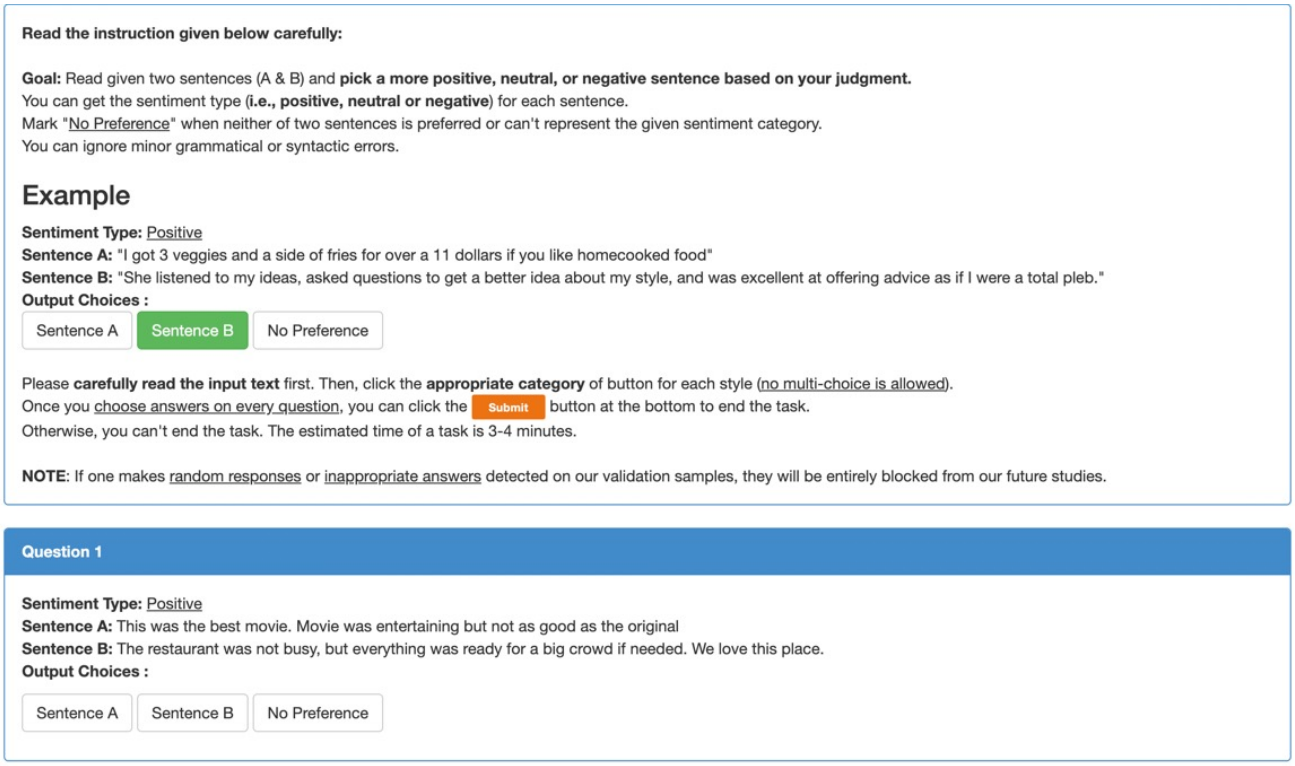
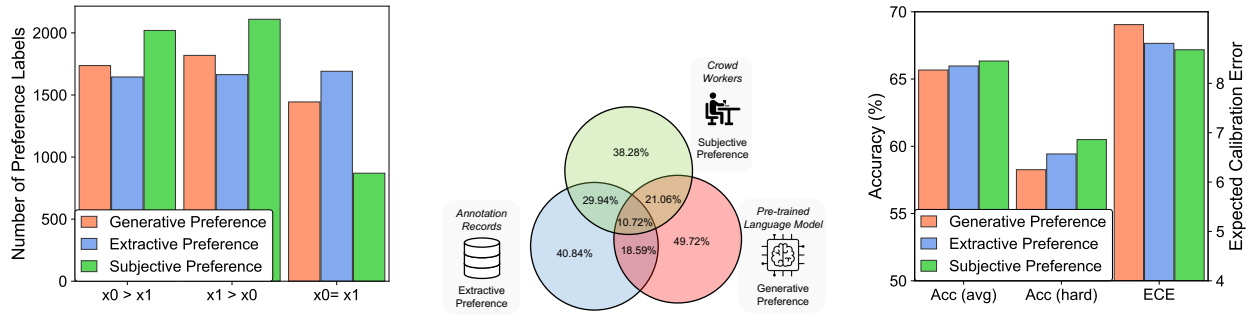


Figure 7. Interface to collect subjective preference from crowd workers for sentiment classification (DynaSent-R2 (Potts et al., 2021)).

they still fail to reach a consensus, this pair is labeled with "No Preference".

Under this procedure, we first gather 1,000 subjective preference labels of randomly selected pairs of sentences. Then, we dynamically collect the additional subjective preference labels to maximize the information of collected pairs, motivated by the recent dynamic benchmark constructions (Kiela et al., 2021; Nie et al., 2020a). Namely, we first train the model with existing subjective preference labels. Then, we find the most informative pairs in the aspect of the trained model, using the disagreement-based sampling introduced in Section 2.2 and query their preference labels in the next stage. We select an equal number of pairs for each class to balance the label distribution. Overall, starting with 1,000 random pairs, we collect the preference of 2,000 pairs at each round and iterate this procedure for 2 rounds, *i.e.*, a total of 5,000 pairs' subjective preference labels are collected.

Figure 7 shows the interface used to collect subjective preference labels from crowd workers for sentiment analysis based on DynaSent-R2 (Potts et al., 2021). The top provides the instructions, and then one example is shown. The whole task has 10 items per Human Interface Task (HIT). Workers were paid US$0.8 per HIT on average, and all workers were paid for their work. To improve the quality of collected preference labels, we only hire the Master workers identified as high-performing workers from Amazon's Mechanical Turk system.



(a) Preference Label Distribution (b) Overlap between Preference Labels (c) Results on DynaSent-R2

Figure 8. Comparison of three different types of preference labels using 5,000 pair of samples on DynaSent-R2. (a) Distribution of each type of preference label. (b) Venn Diagram to denote the similarity (*i.e.*, overlap) between different preference labels. (c) Performance of fine-tuned RoBERTa on the mutually exclusive datasets between all of three different types of preference labels.

B.2. Comparison between different types of preference labels

As generative, extractive, and subjective preferences come from different sources of knowledge, they are naturally expected to have different characteristics. To this end, we first compare the distribution of preference labels with 5,000 pair of samples on DynaSent-R2, which are exactly used to collect the subjective preference labels; as shown in Figure 8(a), they have clearly different label distributions. This discrepancy is more clearly verified when we measure the coincidence between the preference labels (Figure 8(b)); three preference labels output the same label for only 10.72 % of pairs, while outputting the mutually exclusive one for 19.78 % of pairs. To further investigate the effectiveness of each preference label, we fine-tuned RoBERTa model only using those pair of samples with the different preference labels for each method; as shown in Figure 8(c), the subjective preference shows the best performance, while the generative preference shows the worst performance. It implies the importance of the quality of preference labels and the effectiveness of generative preference could be from approximating extractive or subjective ones in a cheap way.

B.3. More examples of preference labels

In Table 8, we present more examples in our extractive, subjective, and generative preference labels on DynaSent-R2 dataset, similar to Table 1.

Table 8. More examples in our extractive, subjective, and generative preference labels on DynaSent-R2.

A: I noticed when I walked in they looked at me, the eyes of them reflecting.	**B**: I've been to the restaurant a more times and I can understand why this dichotomy may exist.
Sentiment: Neutral, Generative Preference: **No preference**, Extractive Preference: **B** $\succ$ **A**, Subjective Preference: **B** $\succ$ **A**	
A: The pet clinic was very unprofessional.	**B**: Fast forward to today 2 months later and still I have not received my plates that I paid for and I am driving around on their temp paper plate. I was angry.
Sentiment: Negative, Generative Preference: **B** $\succ$ **A**, Extractive Preference: **No preference**, Subjective Preference: **No preference**	
A: The fresh bread of the bagel available here.	**B**: Since it isn't a big restaurant, to get the attention from the waitress isn't that hard.
Sentiment: Positive, Generative Preference: **A** $\succ$ **B**, Extractive Preference: **B** $\succ$ **A**, Subjective Preference: **A** $\succ$ **B**	
A: I expect everything to turn out well.	**B**: We tried a new place. We couldn't recommend them more highly.
Sentiment: Positive, Generative Preference: **B** $\succ$ **A**, Extractive Preference: **A** $\succ$ **B**, Subjective Preference: **A** $\succ$ **B**	
A: But his humor isn't for everyone. I love humor.	**B**: That may have been the norm, but they were above average.
Sentiment: Positive, Generative Preference: **B** $\succ$ **A**, Extractive Preference: **A** $\succ$ **B**, Subjective Preference: **No Preference**	
A: Management is an embarrassement.	**B**: I saw table of guys feasting on a whole pigs head and having a great time, but it made me pretty sick.
Sentiment: Negative, Generative Preference: **B** $\succ$ **A**, Extractive Preference: **No Preference**, Subjective Preference: **B** $\succ$ **A**	
A: they put one inside the grocery store.	**B**: We moved here based on reviews and selected South shores with distance of hours
Sentiment: Neutral, Generative Preference: **B** $\succ$ **A**, Extractive Preference: **No Preference**, Subjective Preference: **No Preference**	

C. More Ablation Study

In this section, we provide more ablation studies on the design choices of P2C. Here, all experiments are conducted on DynaSent-R2 (Potts et al., 2021) and Offensive (Leonardelli et al., 2021) datasets with extractive preference labels, as same as we have done in Section 4.3. The values and error bars are the mean and standard deviation across five random seeds. The results with the chosen design in Section 4.3 are indicated in **bold**.

Multiple preference heads for preference learning. In Section 2.2, we introduce multi-preference heads with diversity regularization (Eq. 3) to effectively learn the given preference labels. To see the effect, we compare it with two different designs for preference heads: 1) single-preference head and 2) multi-preference heads without diversity regularization. Remark that the other components, consistency regularization, and inconsistency-based sampling, are still applied to separately verify the effect from different designs of the preference head. As shown in Table 9, one can verify that a single preference head is not enough to exploit the given preference labels fully; hence, the empirical gain is relatively small compared to multi-preference heads. Also, it is observable that the proposed regularization is more effective to impose diversity than only relying on random initialization.

Table 9. Effect of different designs for preference head.

Dataset	Single-Pref Head	Multi-Pref Heads without diversity	Multi-Pref Heads with diversity
DynaSent-R2	72.22$\pm$0.55	72.75$\pm$0.42	**73.06**$\pm$0.31
Offensive	77.08$\pm$0.57	77.25$\pm$0.92	**77.81**$\pm$0.21

Auxiliary loss for preference learning. As described in Section 2.2, we use a consistency regularization (Eq. 5 and 6) between classification and preference learning as an auxiliary loss for learning preference; specifically, consistency regularization with margin (Eq. 6) is used in Section 4.3. To clarify the effectiveness of this regularization, we compare

it with 1) consistency regularization without margin (Eq. 5). We also compare it to 2) soft-labeling, which also uses the annotation records to construct soft-labels instead of the preference and margin. Here, we use random sampling instead of inconsistency-based sampling since it is designed explicitly for consistency regularization while using the multi-preference heads. Table 10 shows the results of these auxiliary losses; although consistency regularization is effective in improving the performance without margin, the gain is smaller than the consistency regularization with margin since the latter utilizes the additional knowledge about the given preference label. In addition, the result with soft-labeling validates that the gain from our consistency loss is not from the use of the annotation records but from the regularization that imposes the following intuition: *more preferred instance should have higher confidence from the classifier*.

Table 10. Effect of different auxiliary losses to learn.

Dataset	Soft -labeling	Consistency without margin	Consistency with margin
DynaSent-R2	72.29$\pm$0.88	72.40$\pm$0.71	**72.67**$\pm$0.89
Offensive	77.04$\pm$1.05	77.54$\pm$0.95	**77.67**$\pm$0.99

Sampling of pairs for preference learning. To improve the efficiency of preference learning by sampling the informative pairs during the training, we introduce two advanced sampling methods: (1) *disagreement-based* sampling and (2) *inconsistency-based sampling* in Section 2.2. Remark that the other components, consistency regularization with margin and multi-preference heads, are still applied to verify the effect from different sampling methods separately. In Table 11, we compare both sampling methods to random sampling. Here, one can verify that both ways are more effective than random sampling, and inconsistency-based sampling is slightly better than disagreement-based sampling. Hence, we commonly used inconsistency-based sampling in Section 4.3.

Table 11. Effect of different sampling methods.

Dataset	Random	Disagreement	Inconsistency
DynaSent-R2	72.67$\pm$0.89	72.73$\pm$0.66	**73.06**$\pm$0.31
Offensive	77.67$\pm$0.99	77.75$\pm$1.49	**77.81**$\pm$0.21

Sensitivity to $\mathcal{L}_{\texttt{div}}$. To verify the sensitivity of our method with $\mathcal{L}_{\texttt{div}}$, we conduct the experiments by introducing $\lambda_{\texttt{div}}$, a coefficient of $\mathcal{L}_{\texttt{div}}$, and varying it to investigate its effect. In Table 12, one can observe that KL divergence does not dominate the entire loss until the certain level of $\lambda_{\texttt{div}}$ including the original value ($\lambda_{\texttt{div}}$=1), but it can diverge with too large value (*e.g.*, $\lambda_{\texttt{div}} = 10$). Hence, we recommend using the original value or investigating $\lambda_{\texttt{div}}$ with smaller than 1.

Table 12. Effect of diversity regularization between multi-preference heads with $\lambda_{\texttt{div}}$.

Dataset	$\lambda_{\texttt{div}} = 0$	$\lambda_{\texttt{div}} = 1$	$\lambda_{\texttt{div}} = 2$	$\lambda_{\texttt{div}} = 10$
DynaSent-R2	72.75$\pm$0.42	**73.06**$\pm$0.31	71.44$\pm$0.68	57.05$\pm$2.14
Offensive	77.25$\pm$0.92	**77.81**$\pm$0.21	75.35$\pm$1.03	65.05$\pm$6.70

D. Additional Experimental Results

Smaller training samples. Here, we validate the effectiveness of P2C with extractive preferences for the smaller training samples. Specifically, we control the number of training samples (N) of the DynaSent-R2 dataset from $N = 250$ to $N = 4000$ and compare our method with three representative baselines with high performance: Vanilla, Soft-labeling, and Multi-annotator. As shown in Table 13, P2C shows significant improvement, especially when the dataset size is smaller. We also remark that P2C shows consistent improvement for all cases while other baselines do not.

Table 13. Results with the smaller training samples.

Method	$N = 250$	$N = 500$	$N = 1000$	$N = 2000$	$N = 4000$
Vanilla	54.89$_{\pm 2.46}$	60.36$_{\pm 2.98}$	63.61$_{\pm 0.92}$	66.50$_{\pm 0.76}$	68.69$_{\pm 1.41}$
Soft-labeling	57.75$_{\pm 2.35}$	60.03$_{\pm 1.46}$	62.81$_{\pm 1.45}$	66.78$_{\pm 1.16}$	68.17$_{\pm 1.09}$
Multi-annotator	57.33$_{\pm 3.23}$	61.39$_{\pm 1.76}$	63.00$_{\pm 0.87}$	66.19$_{\pm 0.84}$	68.78$_{\pm 1.46}$
P2C (Ours)	**58.94**$_{\pm 1.16}$	**61.83**$_{\pm 1.15}$	**64.13**$_{\pm 1.04}$	**67.72**$_{\pm 0.46}$	**69.83**$_{\pm 0.64}$

Compatibility with other types of models. While we have previously used a model built over RoBERTa-base (Liu et al., 2019), the proposed P2C is not limited to the specific model. To verify this, we conduct additional experiments based on DynaSent-R2 with extractive preference labels from the annotation records. As shown in Table 14, the proposed P2C consistently improves the test accuracy of classifiers of other language models: BERT-base (Devlin et al., 2019), ALBERT-base (Lan et al., 2020), and RoBERTa-large.

Table 14. Results with other types of language models.

Method	BERT-base	ALBERT-base	RoBERTa-large
Vanilla	67.26$_{\pm 1.15}$	62.72$_{\pm 0.73}$	75.62$_{\pm 0.60}$
P2C (Ours)	68.26$_{\pm 0.56}$	65.00$_{\pm 1.13}$	77.71$_{\pm 0.36}$