Leveraging Moment Injection for Enhanced Semi-supervised Natural Language Inference with Large Language Models

Seo Yeon Park
Computer Science & Engineering
Hanyang University (ERICA)
seoyeonpark@hanyang.ac.kr

Abstract

Natural Language Inference (NLI) is crucial for evaluating models' Natural Language Understanding (NLU) and reasoning abilities. The development of NLI, in part, has been driven by the creation of large datasets, which require significant human effort. This has spurred interest in semi-supervised learning (SSL) that leverages both labeled and unlabeled data. However, the absence of hypotheses and class labels in NLI tasks complicates SSL. Prior work has used class-specific fine-tuned large language models (LLMs) to generate hypotheses and assign pseudo-labels but discarded many LLM-constructed samples during training to ensure the quality. In contrast, we propose to leverage all LLM-constructed samples by handling potentially noisy samples by injecting the moments of labeled samples during training to properly adjust the level of noise. Our method outperforms strong baselines on multiple NLI datasets in low-resource settings.

1 Introduction

Natural Language Inference (NLI) is a sentence pair classification task aimed at identifying the relationship between two sentences by determining whether they reflect *entailment, neutral,* or *contradiction*. NLI plays a key role in assessing a model's capacity for Natural Language Understanding (NLU) and reasoning. The advancement of NLI, partially, has been fueled along with the creation of large datasets such as SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and ANLI (Nie et al., 2020). However, creating a large-scale NLI benchmark requires a considerable amount of human effort since human annotators should generate texts that requires logical reasoning and inference. For example, during the creation of the SNLI and MNLI datasets, human workers are given unlabeled premises and are prompted to ***generate hypotheses*** corresponding to each class label—*entailment, neutral, contradiction*. The high cost and complexity of labeling NLI data have driven interest in semi-supervised learning (SSL), which utilizes both labeled and unlabeled data. However, unlike single-sentence classification, unlabeled data in NLI is more challenging to handle because one of the sentence pairs (typically the hypothesis) and the class label are missing, requiring significant human annotation. Consequently, to effectively use unlabeled data for SSL in NLI, the challenge of missing hypotheses and class labels should be addressed.

To address the challenge of missing hypotheses and class labels in semi-supervised learning (SSL) for Natural Language Inference (NLI), Sadat and Caragea (2022) proposed a method that generates hypotheses and assigns initial pseudo-labels using class-specifically fine-tuned Large Language Models (LLMs; e.g., BART (Lewis et al., 2020)). For each unlabeled premise, one hypothesis is generated for each class in the labeled dataset. However, since LLMs may not always generate the most relevant or accurate output on the first attempt, the resulting data possibly contains noisy samples that degrade performance if used directly. To mitigate this, they employed self-training, specifically pseudo-labeling (Lee, 2013). In their proposed approach, a task classifier (e.g., BERT) generates a pseudo-label for each LLM-generated sample. If the pseudo-label from the class-specifically fine-tuned LLM does not match the one from the task classifier, the sample is considered low quality and discarded. Furthermore, even when the pseudo-labels match, they discard less confident (noisy) samples, following the common practice in pseudo-labeling. Previous research on pseudo-labeling typically uses a fixed (or even flexible) confidence threshold, assuming that pseudo-labels with confidence scores above the threshold are of high quality, while those below are of low quality so discard (Chen et al., 2020; Sohn et al., 2020; Zhang

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 641–648
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

et al., 2021; Wang et al., 2023). This possibly results in excluding a substantial number of samples. To tackle this, Chen et al. (2023) proposed to utilize all pseudo-labeled samples by applying lower weights to less confident pseudo-labeled samples during training. While this approach significantly enhances the diversity of the training data compared to earlier methods, erroneous pseudo-labels can still be included with high weights as training continues.

To this end, we propose a method of leveraging LLM-generated pseudo-labeled samples without discarding any to ensure a model is exposed to a wide range of data while minimizing the impact of noisy samples. In our approach, we first construct pseudo-labeled samples by using one of the recent state-of-the-art LLMs, Llama 3. Specifically, given a small amount of labeled data, we first fine-tune Llama 3 with Low-Rank Adaptation (LoRA; Hu et al. (2021)) for every class. We then use these class-specific LoRA-tuned LLMs for generating hypotheses for a given unlabeled premise along with assigning the initial pseudo-label. For example, given a premise ‘A man cutting down a tree during winter’, we produce three hypotheses, one for each class, ‘entailment,’ ‘contradiction,’ and ‘neutral,’ by using the corresponding class-specific LoRA-tuned LLM.

Afterward, unlike the previous SSL research that usually discards samples, we propose to leverage all LLM-constructed samples but with injecting the moments of labeled data into LLM-constructed data. This allows us to calibrate the noisiness of the potentially mislabeled LLM-constructed samples, making them more beneficial for training. Our proposed method is inspired by the work proposed by Li et al. (2021) which revealed that the moments (a.k.a., mean and standard deviation) of latent features obtained from various layers of deep networks play a central role in image recognition tasks. They showed that swapping a sample’s moments of latent features to another sample allows a model to capture the underlying structure of both samples through the normalized features (from the original image) and the moments (from the other image). Inspired by this, we inject the moments of labeled data into LLM-constructed data that makes the LLM-constructed samples follow the distribution and underlying structure of labeled samples. This results in potentially noisy LLM-constructed samples behaving as labeled samples but with a proper noise level. Consequently, we effectively harness LLM-constructed data to boost the performance of SSL on NLI. We validate our method on various NLI datasets and show our method achieves competitive performance compared to strong baselines.

2 Proposed Approach

LLM-constructed Data Creation Let $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1,\cdots,n}$ be a labeled training set where $x_i = (p_i, q_i)$ refers to a premise and hypothesis sentence-pair in NLI, and y_i represents one of three NLI classes (i.e., ‘contradiction’, ‘entailment’, ‘neutral’). Furthermore, let $\mathcal{D}_u = \{p_j^u\}_{j=1,\cdots,N}$ be a set of unlabeled premises of size N where $N \gg n$. To create Large Language Model (LLM) constructed data, we first fine-tune Llama 3 with LoRA for each class using labeled samples corresponding to that class. We then provide an unlabeled premise into these class-specific LoRA-tuned Llama 3 to generate hypotheses, each assigned a pseudo-label by the corresponding model. Thus, we ensure comprehensive coverage of all classes within LLM-constructed samples. We formulate LLM-constructed data as follows:

$$\mathcal{D}_{\text{pseudo}} = \{(\hat{x}_j = (p_j^u, \hat{q}_j = \phi^c(p_j^u)), \hat{y}_j^{\text{llm}})\} \\ j = 1 \ldots c \cdot N, c \in C$$

where p_j^u is a premise, ϕ^c is a LoRA-tuned Llama 3 on class c, $\hat{q}_j$ is a generated hypothesis, and $\hat{y}_j^{\text{llm}}$ is a pseudo-label assigned by ϕ^c.

Semi-supervised learning with Moment Injection Let φ be a task classifier (i.e., a pre-trained language model such as BERT). For each sample x (either labeled x_i or LLM-constructed $\hat{x}_j$), we generate a hidden state representation H from the last layer of φ where $H \in \mathbb{R}^{L \times K}$ represents the hidden states of all tokens in the sequence. Here, L denotes the sequence length (i.e., the number of tokens in the input sentence x), and K is the hidden size (e.g., for BERT-base, $K = 768$). We then calculate the sample’s mean μ_x and standard deviation σ_x of x as follows:

$$\begin{aligned} \mu_x &= \frac{1}{L} \sum_{\ell=1}^{L} H_\ell \\ \sigma_x &= \sqrt{\frac{1}{L} \sum_{\ell=1}^{L} (H_\ell - \mu_x)^2} \end{aligned} \tag{1}$$

where H_ℓ represents the hidden state of the ℓ-th token in the sequence. Given two randomly

	RTE	SICK	SNLI-2.5K	MNLI-2.5k$_m$	MNLI-2.5k$_{mm}$
Fine-tuning (FT) BERT (Devlin et al., 2019)	$60.90_{1.6}$	$84.63_{0.7}$	$79.03_{0.1}$	$69.26_{0.9}$	$70.26_{0.7}$
GPT-2 ICL (Brown et al., 2020)	$54.94_{2.2}$	$59.38_{3.2}$	$33.37_{0.3}$	$33.51_{1.3}$	$33.09_{0.4}$
Llama 3-8B-Instruct ICL	$68.22_{0.0}$	$55.31_{0.0}$	$59.67_{0.0}$	$59.74_{0.0}$	$58.72_{0.0}$
Mistral-7B ZSL (Jiang et al., 2023)	$60.41_{0.0}$	$48.82_{0.0}$	$45.34_{0.0}$	$47.27_{0.0}$	$49.69_{0.0}$
Llama 2-7B ZSL (Touvron et al., 2023)	$67.30_{0.0}$	$49.06_{0.0}$	$56.70_{0.0}$	$55.04_{0.0}$	$57.23_{0.0}$
Llama 3-8B-Instruct ZSL	$68.88_{0.0}$	$55.47_{0.0}$	$60.19_{0.0}$	$58.87_{0.0}$	$59.61_{0.0}$
LM-BFF (Gao et al., 2021)	$60.64_{0.9}$	$81.59_{0.8}$	$73.91_{0.6}$	$62.89_{1.2}$	$65.54_{0.8}$
LM-BFF + Demo	$61.26_{1.8}$	$82.22_{0.5}$	$74.56_{0.9}$	$62.55_{1.2}$	$64.09_{0.5}$
Back Translation (Edunov et al., 2018)	$61.22_{1.3}$	$84.38_{1.1}$	$79.15_{1.2}$	$72.01_{1.0}$	$73.38_{0.9}$
TMix (Chen et al., 2020)	$61.59_{1.5}$	$83.23_{1.9}$	$79.13_{1.0}$	$71.86_{0.6}$	$73.21_{0.8}$
UDA (Xie et al., 2020)	$65.53_{0.9}$	$85.46_{0.8}$	$80.06_{0.4}$	$\underline{72.97}_{0.5}$	$\underline{73.82}_{0.5}$
MixText (Chen et al., 2020)	$\underline{68.49}_{2.1}$	$85.44_{0.6}$	$80.11_{0.2}$	$72.45_{0.8}$	$73.42_{1.0}$
SSL for NLI (Sadat and Caragea, 2022)	$68.32_{2.3}$	$\underline{85.77}_{0.7}$	$80.26_{1.1}$	$72.56_{0.3}$	$73.48_{0.1}$
FixMatch (Sohn et al., 2020)	$67.69_{2.8}$	$85.01_{0.6}$	$80.65_{0.9}$	$71.76_{0.5}$	$72.31_{0.6}$
FlexMatch (Zhang et al., 2021)	$67.87_{0.5}$	$84.87_{1.1}$	$79.91_{0.2}$	$72.21_{0.3}$	$73.59_{0.4}$
FreeMatch (Wang et al., 2023)	$67.75_{1.8}$	$84.65_{0.6}$	$80.52_{1.2}$	$72.59_{0.8}$	$73.21_{1.1}$
SoftMatch (Chen et al., 2023)	$68.11_{1.3}$	$84.36_{0.7}$	$\underline{80.83}_{1.2}$	$72.35_{0.5}$	$73.11_{0.6}$
Ours	$\mathbf{71.73}^{\dagger}_{2.0}$	$\mathbf{87.05}_{0.8}$	$\mathbf{82.70}^{\dagger}_{0.4}$	$\mathbf{74.73}^{\dagger}_{0.6}$	$\mathbf{74.96}^{\dagger}_{0.4}$

Table 1: The comparison of test accuracy (%) of our method and baselines. The underlined text shows the best performance baseline methods. We report the mean and standard deviation across three training runs with random restarts. †: our method improves the the best baseline at $p < 0.05$ with paired t-test.

chosen samples—one labeled sample x_i and one LLM-constructed sample $\hat{x}_j$—along with their corresponding `[CLS]` hidden states[1], $h_i = H^i_{[CLS]}$ and $\hat{h}_j = \hat{H}^j_{[CLS]}$, we inject the first and second moments, μ_{x_i} and σ_{x_i}, into the LLM-constructed `[CLS]` hidden states $\hat{h}_j$ as follows:

$$\hat{h}^i_j = \frac{\hat{h}_j - \mu_{\hat{x}_j}}{\sigma_{\hat{x}_j}} \cdot \sigma_{x_i} + \mu_{x_i} \tag{2}$$

Accordingly, we allow LLM-constructed samples to follow the distribution of labeled samples while preserving the underlying structure of both LLM-constructed samples and labeled samples that lie in LLM-constructed samples' moments $(\mu_{\hat{x}_j}, \sigma_{\hat{x}_j})$ and labeled samples' moments $(\mu_{x_i}, \sigma_{x_i})$. This leads potentially noisy LLM-generated samples to act like labeled samples while maintaining an appropriate level of noise. We calculate the unsupervised loss on LLM-constructed data as follows:

$$\mathcal{L}_{\text{unsup}} = \mathbb{1}(\max(y_j) > \tau) \cdot CE(P(y_j|\hat{h}^i_j), \hat{y}^{\text{llm}}_j) \tag{3}$$

where CE is a cross-entropy loss, τ is a hyperparameter and $P(y_j|\hat{h}^i_j)$ is a class distribution of an LLM-constructed sample given the LLM-constructed sample's feature representation having moments of labeled sample's feature representation. We set τ as 0 so that we encourage a model to leverage all LLM-constructed samples regardless of their confidence. To achieve the final objective, we calculate the cross-entropy loss on the labeled samples $\mathcal{L}_{\text{sup}}$ and add it to $\mathcal{L}_{\text{unsup}}$.

3 Experiments

3.1 Evaluation Setup

Datasets We evaluate our method on RTE (Wang et al., 2018), SICK (Marelli et al., 2014), SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018). For RTE and SICK, we use the entire training data as labeled samples due to their small number in size, and extract unlabeled premises from WikiPedia and CNNDM (Nallapati et al., 2016) for RTE, and from 8k ImageFlickr dataset and WikiPedia for SICK, respectively. For SNLI and MNLI, we extract 2,500 labeled samples per class and considered the premises of the remaining examples as unlabeled data. For each dataset, we select 15,000 unlabeled premises to create LLM-constructed data.

Comparison Methods We compare our proposed method with (1) the standard labeled data fine-tuning using only labeled data on **BERT** (Devlin et al., 2019), (2) LLM baselines including **In-context Learning (ICL)** (Brown et al., 2020)[2], **Zero-Shot Learning (ZSL)** (Brown et al., 2020), and a prompt based fine-tuning **LM-BFF** (Gao et al., 2021), (3) Data Augmentation including **Back Translation** (Edunov et al., 2018) and **TMix** (Chen et al., 2020), (4) semi-supervised learning (SSL) baselines that shows effectiveness in general (**UDA** (Xie et al., 2020), and **MixText** (Chen et al., 2020)), and SSL baselines that leverages pseudo-labeling **SSL for NLI** (Sadat and Caragea, 2022), **FixMatch** (Sohn et al., 2020), **FlexMatch** (Zhang

[1] Note that we use the `[CLS]` hidden representations as features, as they are primarily utilized for training our SSL model.

[2] The prompt is constructed by referring to Brown et al. (2020) as shown in A.2. We follow the evaluation protocol provided by Gao et al. (2021).

	RTE	SICK	SNLI	MNLI$_m$	MNLI$_{mm}$
FT BERT, 500 labeled data	58.16	81.48	63.35	55.79	56.88
SoftMatch, 500 labeled data	65.38	83.26	73.72	62.21	62.81
Ours, 500 labeled data	68.15	83.54	78.94	69.77	70.51
FT BERT, 1,000 labeled data	60.90	84.63	71.89	64.85	65.37
SoftMatch, 1,000 labeled data	68.11	84.36	77.35	66.78	66.63
Ours, 1,000 labeled data	71.73	87.05	79.02	70.51	70.81
FT BERT, 2,500 labeled data	-	-	79.03	69.26	70.26
SoftMatch, 2,500 labeled data	-	-	80.83	72.35	73.11
Ours, 2,500 labeled data	-	-	82.70	74.73	74.96

Table 2: The comparison on various low-resource settings. The maximum number of samples in each class for RTE and SICK is 1,000 since these datasets are small in size.

	RTE	SICK	SNLI-2.5k	MNLI-2.5k$_m$	MNLI-2.5k$_{mm}$
Ours	**71.73**$_{2.0}$	**87.05**$_{0.8}$	**82.70**$_{0.4}$	**74.73**$_{0.6}$	**74.96**$_{0.4}$
w/o Moment Injection	68.47$_{0.5}$	85.52$_{0.8}$	81.76$_{0.5}$	74.20$_{0.9}$	74.58$_{0.9}$
w/ discard unconfident	69.33$_{0.5}$	86.63$_{0.9}$	81.88$_{0.4}$	73.54$_{0.7}$	73.73$_{0.4}$
w/ PL by task classifier	70.64$_{1.3}$	85.37$_{0.8}$	80.93$_{0.7}$	71.87$_{0.8}$	72.81$_{0.3}$

Table 3: The results comparisons of ablation study.

et al., 2021), **FreeMatch** (Wang et al., 2023), and **SoftMatch** (Chen et al., 2023)). We provide detailed information on baseline implementations in the Appendix.

Implementation Details We use Llama-3-8B-Instruct as LLMs and use BERT-base as a task classifier from HuggingFace Transformers library. The hyper-parameters settings are shown in Appendix.

3.2 Results

Main results We observe our method improves over all baseline methods as shown in Table 1. We also observe that LLM baselines (i.e., *In-Context Learning (ICL), Zero-Shot Learning (ZSL), and LM-BFF),* and data augmentation baselines (i.e., *Back Translation, TMix*), generally perform significantly worse compared to SSL baselines that use the same LLM-constructed data as unlabeled data as our approach (i.e., *UDA, MixText, SSL for NLI, FixMatch, FlexMatch, FreeMatch, SoftMatch*). We conclude that leveraging LLM-constructed data boosts performance more than using labeled data. Still, our method achieves better performance than the best SSL baseline. In particular, our method outperforms SoftMatch which also leverages all samples from the unlabeled data. This supports that our method that incorporates all LLM-constructed samples after injecting the moments of labeled samples is effective.

Reducing the quantity of labeled data For a thorough evaluation of our proposed method on various low-resource settings, we reduce the number of labeled samples per class to 500 and 1,000, and present the results in Table 2. The amount of LLM-constructed data remains constant at 15,000 samples per class as reported in Table Table 1. Our method achieves the best performance compared to baselines on all settings.

3.3 Ablation Study

Without Moment Injection To explore the impact of moment injection in our proposed method, we show the results without using it in Table 3 under the line "*w/o Moment Injection*". We observe a drop in performance which shows that LLM-constructed data possibly contains noisy samples which can harm the performance if directly used. We conclude that our proposed method which uses the moment injection allows to incorporating of these noisy samples appropriately, hence, leading to performance improvement.

Discard Unconfident LLM-constructed Data To explore the impact of discarding less confident LLM-constructed samples in our proposed method, we set a threshold value in Eq. (3) as 0.9 following the common practice of using a high fixed threshold (Sadat and Caragea, 2022; Sohn et al., 2020; Chen et al., 2020). We show results in Table 3 under the line "*w/ discard unconfident*" We observe the performance degradation when discarding less confident (i.e., potentially noisy) LLM-constructed samples, clearly demonstrating that our method, which leverages all LLM-constructed samples with moment injections, is the more effective approach.

Confirmation Bias In our method, we calculate the unsupervised loss on LLM-constructed samples in Eq. (3) by using the pseudo-label assigned by class-specifically LoRA-tuned Llama 3. We hypothesize that using the pseudo-label obtained by the task classifier results in performance degradation due to confirmation bias where a model is prone to confirm its mistakes (Tarvainen and Valpola, 2017; Arazo et al., 2020; Zhang et al., 2016)). This is because the task classifier produces pseudo-labels that are potentially mislabeled. This is because LLM-constructed data contains significant noisy data, and the task classifier fits for noisy data. To explore this, we conduct an ablation study. Instead of using class-specifically LoRA-tuned LLM-constructed pseudo-labels (i.e., $\hat{y}_u^{\text{llm}}$) in Eq. (3), we use the task classifier BERT generated pseudo-labels (i.e., $\hat{y}_u = \text{argmax} P(y_u|h_u^l)$). We report the results in Table 3 under the line "*w/ Pseudo Label (PL) by task classifier*. We observe performance drops in all datasets, which supports our hypothesis.

4 Conclusion

We proposed an enhanced semi-supervised learning framework for Natural Language Inference

(NLI), which constructs pseudo-labeled samples using large language models (LLMs), and introduced moment injection to ensure the quality of LLM-constructed samples since LLM might fail to be accurate on their first try. Our proposed method leverages all LLM-generated samples instead of discarding them if less confident as in the previous works, so enhances the exposure of a model to a broader range of samples. We empirically validate that our method achieves competitive performance compared to strong baselines for various NLI datasets in low-resource settings.

5 Limitations

Our proposed method can be computationally expensive since it requires additional training overhead for creating Large Language Model (LLM)-constructed data. In addition to this, we encourage utilizing all LLM-constructed samples, rather than discarding less confident (i.e., noisy) ones. This possibly increases another computational overhead. To address this limitation, we use a smaller language model for the task classifier, ensuring that the overall training time remains reasonable. Empirically, we demonstrate significant performance improvements across various Natural Language Inference (NLI) datasets. We believe our method represents an important step forward for semi-supervised learning in NLI, offering valuable insights—specifically, that potentially noisy LLM-constructed samples can be effectively managed through moment injection using labeled samples.

Acknowledgements

We would like to thank all reviewers for their insightful comments and suggestions to help improve the paper. This work was supported by the research fund of Hanyang University ERICA (HY-2024-3084).

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. 2023. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. volume abs/2301.10921.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157.

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

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Dong-Hyun Lee. 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q Weinberger. 2021. On feature normalization and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12383–12392.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Mobashir Sadat and Cornelia Caragea. 2022. Learning to infer from unlabeled data: A semi-supervised learning approach for robust natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4763–4776, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. 2023. Freematch: Self-adaptive thresholding for semi-supervised learning. In *Eleventh International Conference on Learning Representations*. OpenReview.net.

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

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34.

Zixing Zhang, Fabien Ringeval, Bin Dong, Eduardo Coutinho, Erik Marchi, and Björn Schüller. 2016. Enhanced semi-supervised learning for multimodal emotion recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5185–5189. IEEE.

A Appendix

A.1 Training Details

LLM-constructed Data Creation We mainly use Llama-3-8B-Instruct as large language

Premise	*Generated Hypothesis*	*Pseudo-label*
Bob Dylan is in the hospital with a chest infection.	Bob Dylan is in the hospital with a broken arm.	contradiction
Marjorie, it's been a delight to correspond with you this week.	It's been a pleasure to talk to you this week.	contradiction
But Clinton advisers said that they expected Mr. Lott's comments to backfire.	Clinton's advisors said they expected Lott's comments to backfire.	entailment
Contrary to David Plotz's Assessment, Winnie-the-Pooh is neither American nor British.	Winnie-the-Pooh is neither American nor British.	entailment
The next day they were all dead but Thorn.	Thorn was the only one who survived the attack.	neutral
The emphasis switched from heaven to earth.	The emphasis switched from the afterlife to the present.	neutral

Table 4: Examples of LLM-constructed data by using LoRA-tuned Llama 3

models (LLMs) in LLM-constructed data creation using HuggingFace Transformers library[3]. For LoRA-tuned Llama 3 (Low-Rank Adaptation; Hu et al. (2021)), we set a learning rate as 2e-3, training epoch as 3, LoRA alpha as 8, LoRA dropout as 0.05, train batch size as 1, gradient accumulation steps as 64. We set the LoRA rank value as 4 for RTE, 16 for SICK, and 8 for both SNLI and MNLI datasets. We use the system prompt as follows: "`<s>[INST] «SYS»\nYou are a helpful, respectful, and honest assistant. Always follow the instructions provided and answer honestly.\n«/SYS»\n\n`" and provide customized prompts depending on target labels as follows: (1) entailment: "`We will give you the sentence. Using only the given sentence and what you know about the world. Write one alternate sentence that is definitely a` **`true`** `description of the given sentence. Sentence: {premise}`", (2) contradiction: "`We will give you the sentence. Using only the given sentence and what you know about the world. Write one alternate sentence that is definitely a` **`false`** `description of the given sentence. Sentence: {premise}`" (3) neutral: "`We will give you the sentence. Using only the given sentence and what you know about the world. Write one alternate sentence that` **`might be a true`** `description of the given sentence. Sentence: {premise}`". We construct the system prompt as suggested by the Llama 3 pre-training step while constructing task-dependent prompts by referring to the instructions provided when generating a large-scale Natural Language Inference (NLI) benchmark as in Bowman et al. (2015). The LLM-constructed data creation takes less than an hour using two NVIDIA RTX A6000 GPUs. It took less than ≈ 1 hour to generate the hypotheses for each dataset using the same GPUs.

Task Classifier We use `bert-base-uncased` as a task classifier model where we use the final layer of BERT `[CLS]` token output representations with a maximum of 3 epochs. We optimize the models by using AdamW (Loshchilov and Hutter, 2018). We set a batch size of 32 for both labeled and LLM-constructed data, a learning rate of 2e-5, a gradient clip of 1.0, and no weight decay. We report the mean and standard deviation across three training runs with random restarts.

Training a task classifier is done with a single NVIDIA RTX A6000 GPU with a total time for fine-tuning a single model being less than an hour. For semi-supervised learning baseline methods, we use batch size 16 across all datasets. We set $\tau = 0.95$ in FixMatch (Sohn et al., 2020), set $\tau = 0.95$ in FlexMatch (Zhang et al., 2021), and $\lambda = 0.3$ to obtain τ in FreeMatch (Wang et al., 2023).

A.2 Baseline prompting

To report the results of Large Language Models (LLMs) baseline prompting methods such as in-context and zero-shot learning, we design the prompts based on Brown et al. (2020) as follows: `premise \nQuestion: hypothesis True, False, or Neither?\nAnswer:` . For in-context learning, we prepend the prompts with 10 randomly selected labeled examples (approximately 3 examples per class), including their answers. We follow the same evaluation protocol following Gao et al. (2021).

[3] https://huggingface.co/docs/ transformers/index

A.3 Examples of LLM-constructed Data

We show examples from the LLM-constructed data on MNLI in Table 4. We find that LLM-constructed data include samples that may lead to spurious correlations in Natural Language Inference (NLI). For instance, there is often a high word overlap between the premise and hypothesis in samples labeled as '*entailment*' We find that many LLM-constructed samples that have a class of '*contradiction'* are erroneously labeled. For example, *Marjorie, it's been a delight to correspond with you this week.* and '*It's been a pleasure to talk to you this week.*' should not have 'contradiction' label since both sentences imply the same semantics. Along with this, we find that LLM-constructed samples that have a class of *'neutral*' are indistinct. Hence, we conclude that LLM-generated data contains many noisy samples, which can harm performance if directly incorporated into training.