College of Engineering
Computer Science

LLM-Guided Semi-Supervised Learning

Cornelia Caragea

Department of Computer Science

University of Illinois Chicago

Smart Diaspora 2025

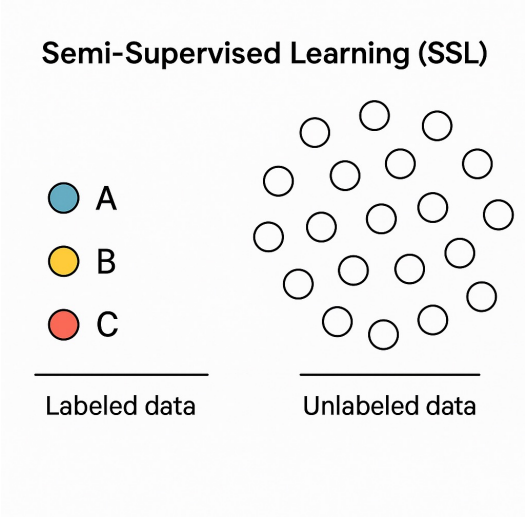
Motivation

- **Deep learning models** achieve impressive performance in many **supervised learning settings**.
  - This is due to their scalability, being able to produce better results when they are trained on large datasets in a supervised fashion.

- However, the performance benefit conferred by the use of a larger dataset comes at a significant cost since labeling data often requires intensive human labor.

Semi-Supervised Learning (SSL)

- SSL is a powerful approach for training models on a large amount of data without requiring a large amount of labels.
  - SSL mitigates the requirement for labeled data by providing a means of leveraging unlabeled data.



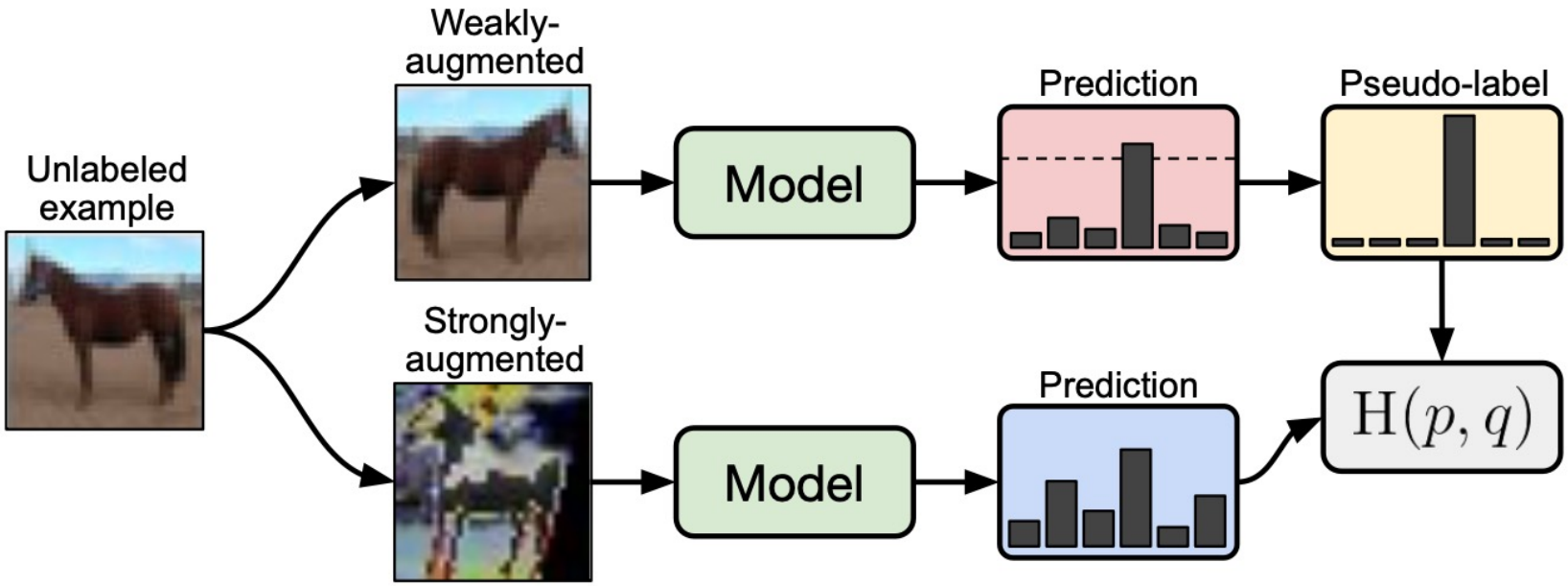
- Since unlabeled data can often be obtained with minimal human labor, any performance boost conferred by SSL often comes with low cost.

Popular SSL Approach

- A widely used SSL approach is **pseudo-labeling** that relies on the model itself to generate pseudo-labels for unlabeled data and uses these pseudo-labels during training.
  - Pseudo-labeling uses a model's predictions of unlabeled examples as labels to train against.

FixMatch

- FixMatch:
  - the artificial label is produced based on a weakly-augmented unlabeled image (e.g., using only flip-and-shift) which is used as a target when the model is fed a strongly-augmented version of the same image.



Sohn et al. (2020). FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence.

FixMatch

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \{(x_b, p_b) : b \in (1, \ldots, B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, \ldots, \mu B)\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.

$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ {*Cross-entropy loss for labeled data*}

for $b = 1$ **to** μB **do**

  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ {*Compute prediction after applying weak data augmentation of* u_b}

end for

$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \mathrm{H}(\arg\max(q_b), p_{\mathrm{m}}(y \mid \mathcal{A}(u_b))$ {*Cross-entropy loss with pseudo-label and confidence for unlabeled data*}

return $\ell_s + \lambda_u \ell_u$

FixMatch relies heavily on a high-confidence fixed threshold to compute the unsupervised loss.

Limitations of FixMatch:

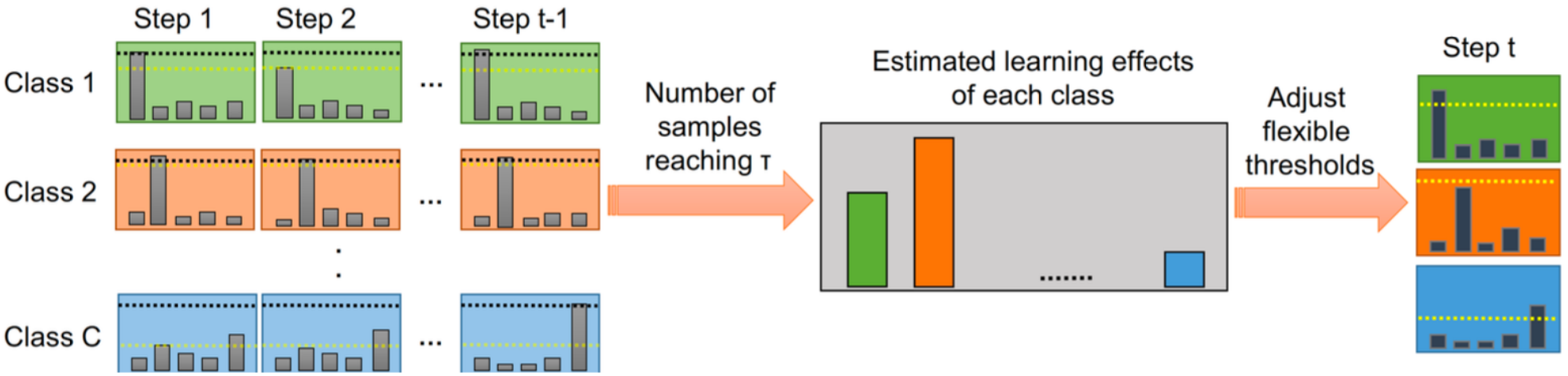
- Ignores a considerable amount of unlabeled data.
- Handles all classes equally without considering their different learning difficulties.

Sohn et al. (2020). FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence.

FlexMatch

- FlexMatch [Zhang et al., 2021] substitutes the pre-defined thresholds with flexible thresholds that are dynamically adjusted for each class according to the current learning status.

- Key assumption is that when the threshold is high, the learning effect of a class can be reflected by the number of samples whose predictions fall into this class and above the threshold.



Zhang et al. (2021). FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling.

FlexMatch

$$\sigma_t(c) = \sum_{n=1}^{N} \mathbb{1}(\max(p_{m,t}(y|u_n)) > \tau) \cdot \mathbb{1}(\arg\max(p_{m,t}(y|u_n) = c).$$

$$\beta_t(c) = \frac{\sigma_t(c)}{\max_c \sigma_t},$$

$$\mathcal{T}_t(c) = \beta_t(c) \cdot \tau.$$

$$\mathcal{L}_{u,t} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) > \mathcal{T}_t(\arg\max(q_b))) H(\hat{q}_b, p_m(y|\Omega(u_b)))$$

$$\mathcal{L}_t = \mathcal{L}_s + \lambda \mathcal{L}_{u,t}$$

Zhang et al. (2021). FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling.

Limitations of Thresholding Approaches

- While FlexMatch handles class difficulty better than FixMatch, both FlexMatch and FixMatch either
  - Discard a large number of diverse pseudo-labeled examples

    or
  - Are susceptible to error accumulation.

Limitations of Thresholding Approaches

- While FlexMatch handles class difficulty better than FixMatch, both FlexMatch and FixMatch either
  - Discard a large number of diverse pseudo-labeled examples

    or
  - Are susceptible to error accumulation.

- SoftMatch [Chen et al., 2023] argued that the quantity-quality trade-off problem of pseudo-labeling with thresholding may prohibit learning.

SoftMatch

- SoftMatch aims at overcoming the trade-off by maintaining both high quantity and high quality of pseudo-labels during training, effectively exploiting the unlabeled data.
  - The authors derived a truncated Gaussian function to weight samples based on their confidence.

Algorithm 1 SoftMatch algorithm.

Input: Number of classes C, labeled batch $\{\mathbf{x}_i, \mathbf{y}_i\}_{i\in[B_L]}$, unlabeled batch $\{\mathbf{u}_i\}_{i\in[B_U]}$, and EMA momentum m.

Define: $\mathbf{p}_i = \mathbf{p}(\mathbf{y}|\omega(\mathbf{u}_i))$

$\mathcal{L}_s = \frac{1}{B_L}\sum_{i=1}^{B_L} \mathcal{H}(\mathbf{y}_i, \mathbf{p}(\mathbf{y}|\omega(\mathbf{x}_i)))$ ▷ Compute $\mathcal{L}_s$ on labeled batch

$\hat{\mu}_b = \frac{1}{B_U}\sum_{i=1}^{B_U} \max(\mathbf{p}_i)$ ▷ Compute the mean of confidence

$\hat{\sigma}^2 = \frac{1}{B_U}\sum_{i=1}^{B_U} (\max(\mathbf{p}_i) - \hat{\mu}_b)^2$ ▷ Compute the variance of confidence

$\hat{\mu}_t = m\hat{\mu}_{t-1} + (1-m)\hat{\mu}_b$ ▷ Update EMA of mean

$\hat{\sigma}_t^2 = m\hat{\sigma}_{t-1}^2 + (1-m)\frac{B_U}{B_U-1}\hat{\sigma}_b^2$ ▷ Update EMA of variance

for i = 1 to B_U **do**

$$\lambda(\mathbf{p}_i) = \begin{cases} \exp\left(-\frac{(\max(\mathrm{UA}(\mathbf{p}_i)) - \hat{\mu}_t)^2}{2\hat{\sigma}_t^2}\right), & \text{if } \max(\mathrm{UA}(\mathbf{p}_i)) < \hat{\mu}_t, \\ 1.0, & \text{otherwise.} \end{cases}$$ ▷ Compute loss weight

end for

$\mathcal{L}_u = \frac{1}{B_U}\sum_{i=1}^{B_U} \lambda(\mathbf{p}_i)\mathcal{H}(\hat{\mathbf{p}}_i, \mathbf{p}(\mathbf{y}|\Omega(\mathbf{u}_i)))$ ▷ Compute $\mathcal{L}_u$ on unlabeled batch

Return: $\mathcal{L}_s + \mathcal{L}_u$

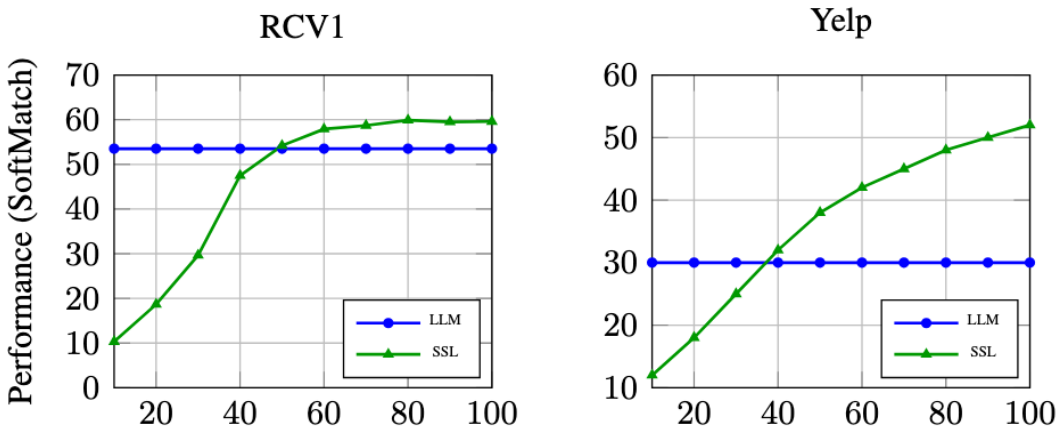
Our Focus

- We posit that relying on the model confidence in predictions can be problematic especially if the model is not well calibrated [Guo et al., 2017; Hosseini and Caragea, 2022; Park and Caragea, 2022] or has poor performance.

- Our focus: the design of approaches to improving pseudo-labeling.

Our Design: LLM-Guided SSL

- LLMs achieve excellent results in many NLP tasks.
  - The quality of their predictions are good but not perfect.

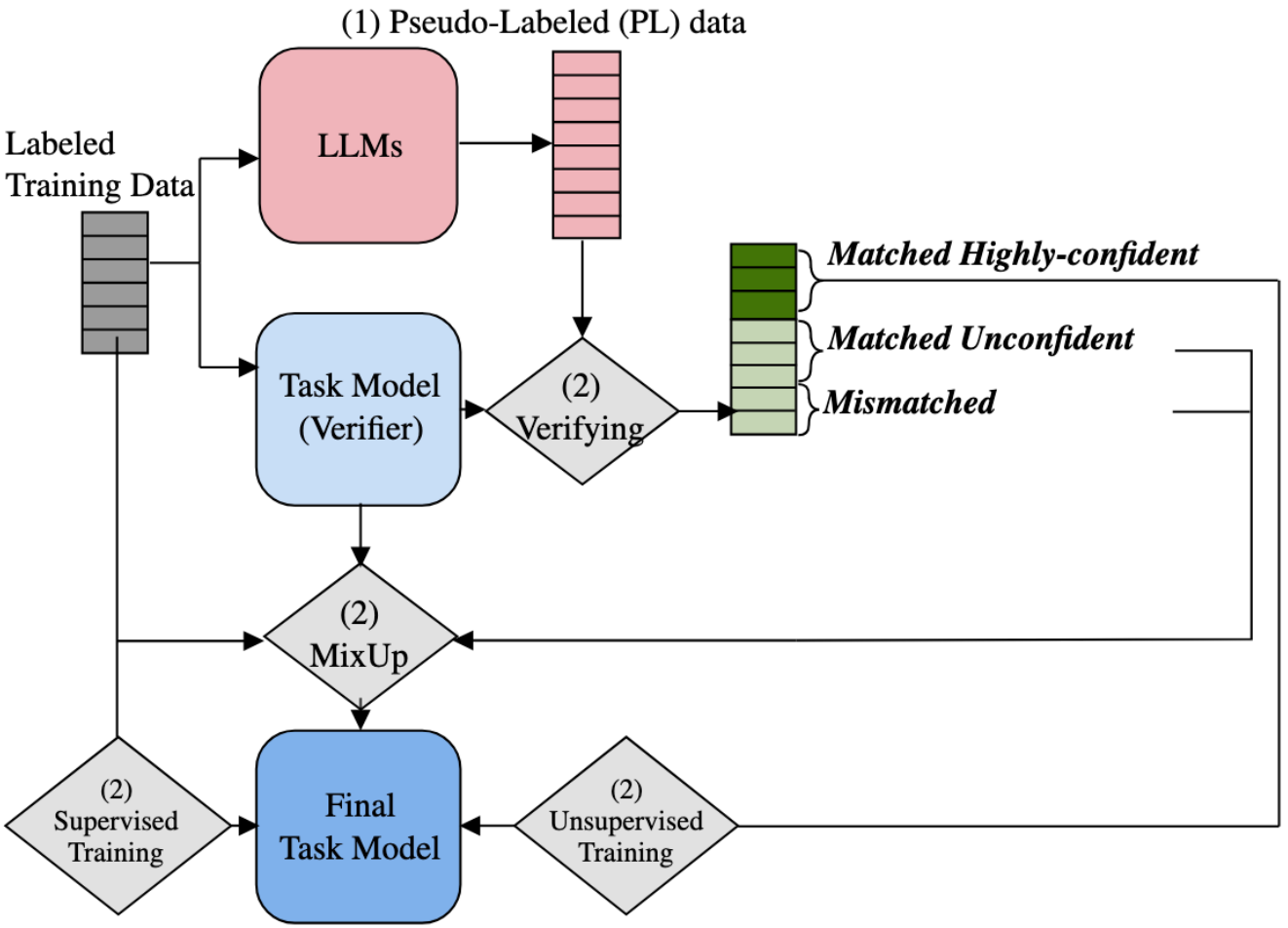


- Our approaches: LLM-Guided SSL
  - VerifyMatch
  - LG-CoTrain

VerifyMatch:
An LLM-Guided Semi-supervised Learning Approach

[Park and Caragea, EMNLP 2024]

VerifyMatch



VerifyMatch

Algorithm 1 : VerifyMatch

Inputs: Labeled data $\mathcal{D}_l$; unlabeled data $\mathcal{D}_u$; SSL model (i.e., a task classifier, verifier) θ, class-specifically fine-tuned LLMs ϕ^c for every $c \in C$

Construct Pseudo-Generated (PG) data $\mathcal{D}_{pseudo} = \{(\hat{x}_i, \hat{y}_i^{llm})\}_{i=1,\cdots,cN}$ where $\hat{x}_i = (\hat{p}_i, \hat{h}_i = \phi^c(\hat{p}_i))$, $p_i \in \mathcal{D}_u$, and $\hat{y}_i^{llm}$ is a initial pseudo-label assigned by ϕ^c, $\hat{y}_i^{llm} = c$

for $t = 1$ to T **do**

  while $\mathcal{D}_{pseudo}$ not exhausted **do**

    Randomly sample labeled batch B_l from $\mathcal{D}_l$, and pseudo-generated batch B_{pseudo} from $\mathcal{D}_{pseudo}$

    Initialize B_m, B_{mm} and B_{unconf} as empty sets, $B_m, B_{mm}, B_{unconf} \leftarrow \emptyset, \emptyset$

    for each $(\hat{x}_i, \hat{y}_i^{llm}) \in B_{pseudo}$ **do**

      Obtain the pseudo-label $\hat{y}_i$ from the task model θ, $\hat{y}_i = \text{argmax}\, P_\theta(y|\hat{\mathbf{x}}_i)$

      if $\hat{y}_i = \hat{y}_i^{llm}$ **then**

        $B_m \leftarrow B_m \cup \{(\hat{\mathbf{x}}_i, \hat{y}_i)\}$

      else # Identify *"mismatched pseudo-generated (PG) data"*

        $(\tilde{\mathbf{x}_k}, \tilde{y}_k)$= $\texttt{MixUp}((\hat{\mathbf{x}}_i, \hat{y}_i^{llm}), (\mathbf{x}_j, y_j))$ using Eq. (1) where $(\mathbf{x}_j, y_j)$ is randomly sampled from B_l

        $B_{mm} \leftarrow B_{mm} \cup \{(\tilde{\mathbf{x}_k}, \tilde{y}_k)\}$

      end if

    end for

    Compute the mean of confidence $\bar{P}$ on B_m using Eq. (3)

    for each $(\hat{x}_i, \hat{y}_i) \in B_m$ **do**

      if $max(P_\theta(y|\hat{\mathbf{x}}_i)) < \bar{P}$ **then** # Identify *"unconfident pseudo-generated (PG) data"*

        $(\tilde{\mathbf{x}}_k, \tilde{y}_k) = \texttt{MixUp}((\hat{\mathbf{x}}_i, \hat{y}_i), (\mathbf{x}_j, y_j))$ using Eq (1) where $(\mathbf{x}_j, y_j)$ is randomly sampled from B_l

        $B_{unconf} \leftarrow B_{unconf} \cup (\tilde{\mathbf{x}}_k, \tilde{y}_k)$

      end if

    end for

    $\mathcal{L}_{sup} = \frac{1}{|B_l|} \sum_{i=1}^{|B_l|} \mathcal{H}(y_i, P_\theta(y|\mathbf{x}_i))$,

    $\mathcal{L}_{unsup} = \frac{1}{|B_m^{\geq \bar{P}}|} \sum_{i=1}^{|B_m^{\geq \bar{P}}|} \mathcal{H}(\hat{y}_i, P_\theta(y|\hat{\mathbf{x}}_i)))$

    $\mathcal{L}_{mm} = \frac{1}{|B_{mm}|} \sum_{k=1}^{|B_{mm}|} \mathcal{H}(\tilde{y}_k, P_\theta(y|\tilde{\mathbf{x}}_k))$

    $\mathcal{L}_{unconf} = \frac{1}{|B_{unconf}|} \sum_{k=1}^{|B_{unconf}|} \mathcal{H}(\tilde{y}_k, P_\theta(y|\tilde{\mathbf{x}}_k))$

    Update the verifier parameter θ using $\mathcal{L}_{sup} + \mathcal{L}_{unsup} + \mathcal{L}_{mm} + \mathcal{L}_{unconf}$

  end while

end for

Evaluation

- Datasets:
  - RTE
  - SICK
  - SNLI
  - MNLI
  - FEVER
  - QQP

- Performance measures: accuracy

Main Results

	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., 2023b)	$67.75_{1.8}$	$84.65_{0.6}$	$80.52_{1.2}$	$72.59_{0.8}$	$73.21_{1.1}$
SoftMatch (Chen et al., 2023)	$68.11_{1.3}$	$84.36_{0.7}$	$\underline{80.83}_{1.2}$	$72.35_{0.5}$	$73.11_{0.6}$
VerifyMatch (Ours)	$\mathbf{71.03}^{\dagger}_{2.1}$	$\mathbf{86.96}^{\dagger}_{0.8}$	$\mathbf{82.06}_{0.3}$	$\mathbf{74.20}^{\dagger}_{0.5}$	$\mathbf{74.10}^{\dagger}_{0.3}$

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. †: VerifyMatch improves the the best baseline at $p < 0.05$ with paired t-test.

Results on Other NLU Tasks

	FEVER-2.5k			QQP-2.5k	
	Dev	Symm-v1	Symm-v2	Test	TwitterPPDB
FT BERT (Devlin et al., 2019)	79.75	49.37	57.31	76.08	84.68
Back Translation (Edunov et al., 2018)	80.54	48.66	57.04	76.60	85.48
FixMatch (Sohn et al., 2020)	82.21	51.16	58.37	78.59	84.69
UDA (Xie et al., 2020)	82.89	52.38	60.57	79.28	84.09
FreeMatch (Wang et al., 2023b)	81.07	49.23	57.58	79.76	84.71
SoftMatch (Chen et al., 2023)	83.72	53.08	61.37	79.85	84.83
VerifyMatch (Ours)	**85.68**†	**54.95**	**62.12**†	**80.89**†	**86.03**†

Evaluation on fact verification (FEVER) and paraphrase detection (QQP) on in- and out-of-domain test data. †: VerifyMatch improves the best baseline at $p < 0.05$ with paired t-test.

VerifyMatch outperforms competitive baselines on both ID and OOD test data, proving its effectiveness.

LG-CoTrain:
LLM-Guided Co-Training for Text Classification

[Rahman and Caragea, EMNLP 2025]

LG-CoTrain

- We develop a novel co-training method that utilizes a small human labeled training set and a large LLM labeled training set.
- Our co-training approach takes the signal from the LLM labels, but their quality is discerned by two classifiers simultaneously trained on different regions of the data.
- Unlike existing co-training approaches which discard pseudo-labeled examples based on some quality assurance filters, our proposed method assigns importance weights indicative of the LLM labels' quality which controls their impact on further classifier training.

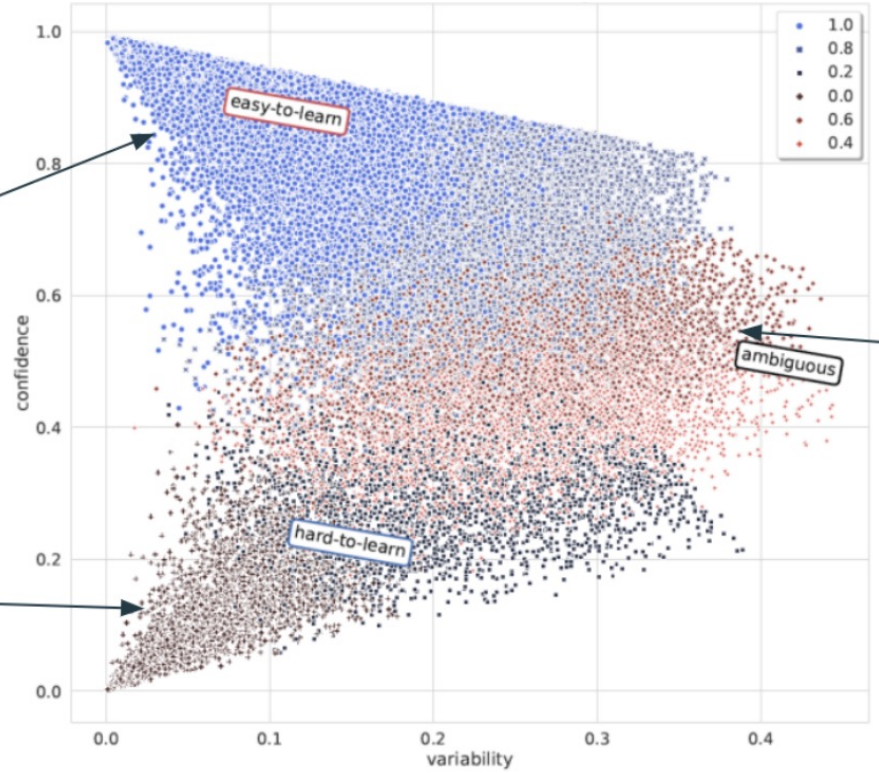
Data Cartography/Map

- Dataset cartography is a tool for mapping a dataset by analyzing the training dynamics i.e., the model's behavior on each example during training.
  - Confidence: mean of the probabilities predicted for the assigned label of each training example.
  - Variability: standard deviation of the probabilities predicted for the assigned label of each training example.
  - Correctness: the fraction of epochs in which the classifier's predicted label matches the assigned label for each training example.

Data Map

Model prediction consistently matches with the assigned label.

Model prediction disagrees with assigned label consistently.



Model is uncertain, and predictions occasionally match with the assigned labels.

Initial Weight Assignment

- Train two initial classifiers using two equal randomly divided subsets of human labeled data.
- While training, record the probabilities predicted for the LLM annotated labels at each epoch.
- Calculate two sets of initial importance weights (W1 and W2) for each LLM annotated example to kick-start the co-training epochs.

Importance Weight Assignment

- Let C1 and C2 be the classifiers that we co-train.
- At each epoch, for each LLM annotated example, two weights are calculated to be used for training classifiers in the next epoch.
  - W1 = confidence (C1) + variability (V1).
  - W2 = confidence (C2) - variability (V2).
- This weighting strategy results in
  - a high weight for the easy examples (more likely to be clean) for both classifiers.
  - a low weight for the hard examples (more likely to be noisy) for both classifiers.
  - a contrasting weight for the ambiguous examples ensuring divergence between the classifiers.

Co-training Epochs

- At each epoch:
  - W2 * cross-entropy loss for each example for C1.
  - W1 * cross-entropy loss for each example for C2.
  - Update the parameters of both C1 and C2 using the scaled losses.
  - Update both sets of confidences, and variabilities.
  - Update both W1 and W2 to be used in the next epoch.

Main Results

Dataset	IMDB		AG News		Amazon Review		Yahoo! Answer		Yelp Review	
# Label	20	100	40	200	250	1000	500	2000	250	1000
Sup (full)	$5.69_{0.15}$		$5.73_{0.11}$		$36.38_{0.01}$		$24.86_{0.07}$		$31.98_{0.20}$	
Sup (lb)	$20.31_{2.79}$	$14.02_{1.22}$	$15.06_{1.08}$	$14.25_{0.97}$	$52.31_{1.28}$	$47.53_{0.69}$	$37.43_{0.29}$	$33.26_{0.10}$	$51.22_{0.98}$	$46.71_{0.37}$
Sup (lb+ps)	$10.68_{1.17}$	$7.88_{0.01}$	$11.88_{0.07}$	$10.92_{0.25}$	$47.77_{0.53}$	$42.23_{0.42}$	$33.38_{0.34}$	$32.15_{0.52}$	$44.44_{0.49}$	$39.57_{0.23}$
UDA	$49.97_{0.04}$	$50.0_{0.0}$	$41.0_{24.96}$	$53.68_{30.15}$	$60.76_{13.61}$	$68.38_{16.44}$	$71.3_{26.45}$	$70.5_{27.58}$	$69.33_{15.08}$	$66.95_{18.46}$
Dash	$8.34_{0.86}$	$7.55_{0.35}$	$31.67_{13.19}$	$13.76_{1.67}$	$47.1_{0.74}$	$43.09_{0.6}$	$35.26_{0.33}$	$31.19_{0.29}$	$45.24_{2.02}$	$40.14_{0.79}$
FixMatch	$7.72_{0.33}$	$7.33_{0.13}$	$30.17_{1.87}$	$11.71_{1.95}$	$47.61_{0.83}$	$43.05_{0.54}$	$33.03_{0.49}$	$30.51_{0.53}$	$46.52_{0.94}$	$40.65_{0.46}$
FlexMatch	$7.82_{0.77}$	$7.41_{0.38}$	$16.38_{3.94}$	$12.08_{0.73}$	$45.73_{1.6}$	$42.25_{0.33}$	$35.61_{1.08}$	$31.13_{0.18}$	$43.35_{0.69}$	$40.51_{0.34}$
SoftMatch	$7.76_{0.58}$	$7.97_{0.72}$	$11.9_{0.27}$	$11.72_{1.58}$	$45.29_{0.95}$	$42.21_{0.2}$	$33.07_{0.31}$	$30.44_{0.62}$	$44.09_{0.5}$	$39.76_{0.13}$
SimMatch	$7.93_{0.55}$	$7.08_{0.33}$	$14.26_{1.51}$	$12.45_{1.37}$	$45.91_{0.95}$	$42.21_{0.3}$	$33.06_{0.2}$	$30.16_{0.21}$	$46.12_{0.48}$	$40.26_{0.62}$
CRMatch	$8.96_{0.88}$	$7.16_{0.09}$	$12.28_{1.43}$	$11.08_{1.24}$	$45.49_{0.98}$	$43.07_{0.5}$	$32.51_{0.4}$	$29.98_{0.07}$	$45.71_{0.63}$	$40.62_{0.28}$
AdaMatch	$8.09_{0.99}$	$7.11_{0.2}$	$11.73_{0.17}$	$11.22_{0.95}$	$46.72_{0.72}$	$42.27_{0.25}$	$32.75_{0.35}$	$30.44_{0.31}$	$45.4_{0.96}$	$40.16_{0.49}$
VerifyMatch $_R$	$7.58_{0.61}$	$7.38_{0.43}$	$11.95_{0.18}$	$11.64_{0.21}$	$39.97_{0.29}$	$40.94_{0.44}$	$32.03_{0.08}$	$32.14_{0.71}$	$37.63_{0.10}$	$37.16_{2.23}$
CoTeach $_R$	$9.04_{2.07}$		$11.04_{0.48}$		$39.38_{0.56}$		$31.31_{0.18}$		$36.35_{1.27}$	
DivideMix $_R$	$7.67_{0.17}$		$11.05_{0.14}$		$39.34_{0.44}$		$30.83_{0.66}$		$35.34_{0.55}$	
Phi-3	**4.78**		12.67		39.32		33.53		34.62	
LG-CoTr $_B$	$7.65_{0.05}$	$7.63_{0.08}$	$11.36_{0.06}$	$10.77_{0.06}$	$38.12_{0.06}$	$37.66_{0.27}$	$29.38_{0.16}$	$\mathbf{28.14^{\dagger}_{0.23}}$	$33.87_{0.14}$	$33.52_{0.12}$
LG-CoTr $_R$	$6.77_{0.32}$	$6.58_{0.15}$	$11.35_{0.17}$	$\mathbf{10.41}_{0.20}$	$\mathbf{37.15^{\dagger}}_{0.19}$	$\mathbf{37.04^{\dagger}_{0.13}}$	$\mathbf{29.31^{\dagger}_{0.11}}$	$28.16_{0.06}$	$\mathbf{33.15^{\dagger}_{0.27}}$	$\mathbf{32.93^{\dagger}_{0.53}}$

Error rate (%) on our classification datasets.

Results on Other NLU Tasks

Dataset	QQP		SWAG		MNLI-m		MNLI-mm	
# Label	100	500	200	1000	150	750	150	750
Phi-3	24.66		23.33		19.36		21.53	
CoTeach	$24.51_{0.64}$		$23.49_{0.28}$		$20.62_{0.46}$		$21.75_{0.40}$	
Vanilla Cotrain	$32.01_{0.25}$	$23.26_{1.28}$	$33.57_{0.38}$	$27.49_{0.52}$	$63.62_{1.40}$	$48.72_{6.44}$	$59.91_{3.27}$	$49.10_{9.01}$
LG-CoTrain	$\mathbf{20.22}^{\dagger}{}_{0.77}$	$\mathbf{17.48}^{\dagger}{}_{0.25}$	$\mathbf{22.53}^{\dagger}{}_{0.27}$	$\mathbf{20.91}^{\dagger}{}_{0.30}$	$\mathbf{16.82}^{\dagger}{}_{1.0}$	$\mathbf{15.29}^{\dagger}{}_{0.29}$	$\mathbf{17.32}^{\dagger}{}_{0.94}$	$\mathbf{15.19}^{\dagger}{}_{0.22}$

Error rate (%) on NLU datasets.

LG-CoTrain outperforms competitive baselines on a variety of NLP tasks, proving its effectiveness.

Conclusion and Future Work

- We proposed novel LLM-guided SSL approaches for text classification which utilize all unlabeled examples to exploit the diversity in the unlabeled data.
- In our approaches, we used either the mixup strategy to clean up ambiguous or noisy samples or proposed a weighting scheme that exploits training dynamics of two classifiers as a quality assurance measure for the LLM pseudo-labels.
- In the future, we will focus on dynamic LLM labels.

Thank you!















Contact: cornelia@uic.edu