PIEClass: Weakly-Supervised Text Classification with Prompting and Noise-Robust Iterative Ensemble Training

Task

Method

Source: EMNLP 2023
Advisor: JIA-LING KOH
Speaker: FAN-CHI-YU
Date:2023/02/27

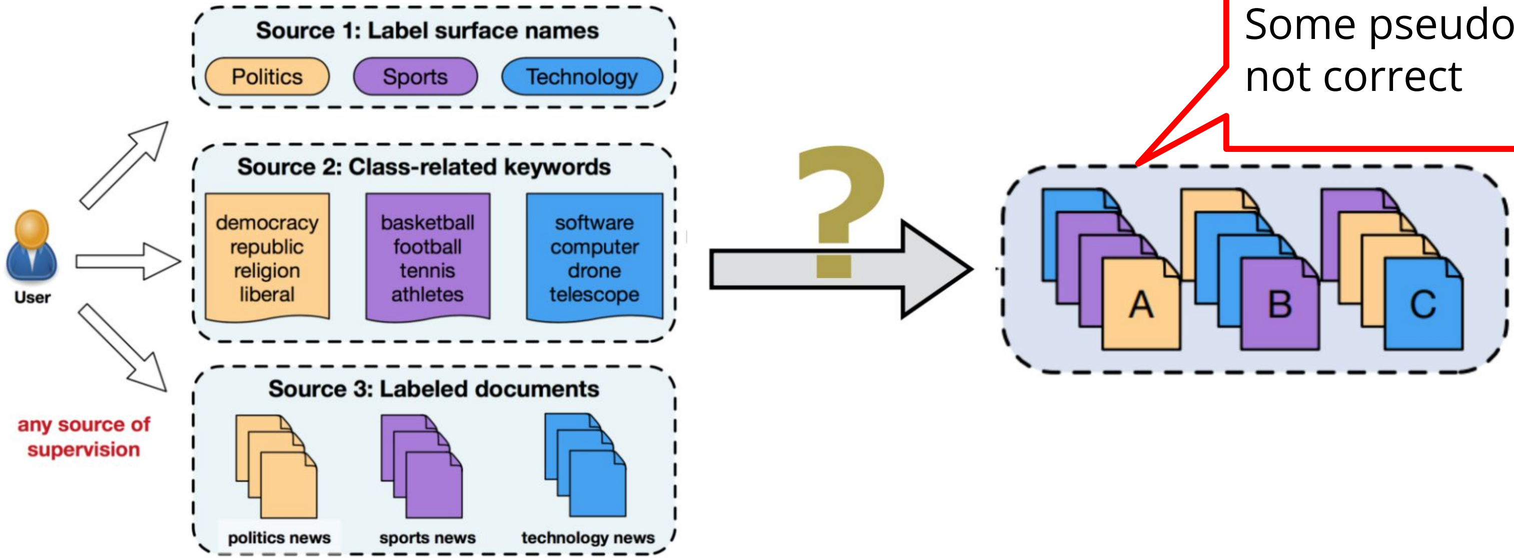
Outline

- Introduction
- Method
- Experiment
- Conclusion

Introduction

Weakly-Supervised Text Classification

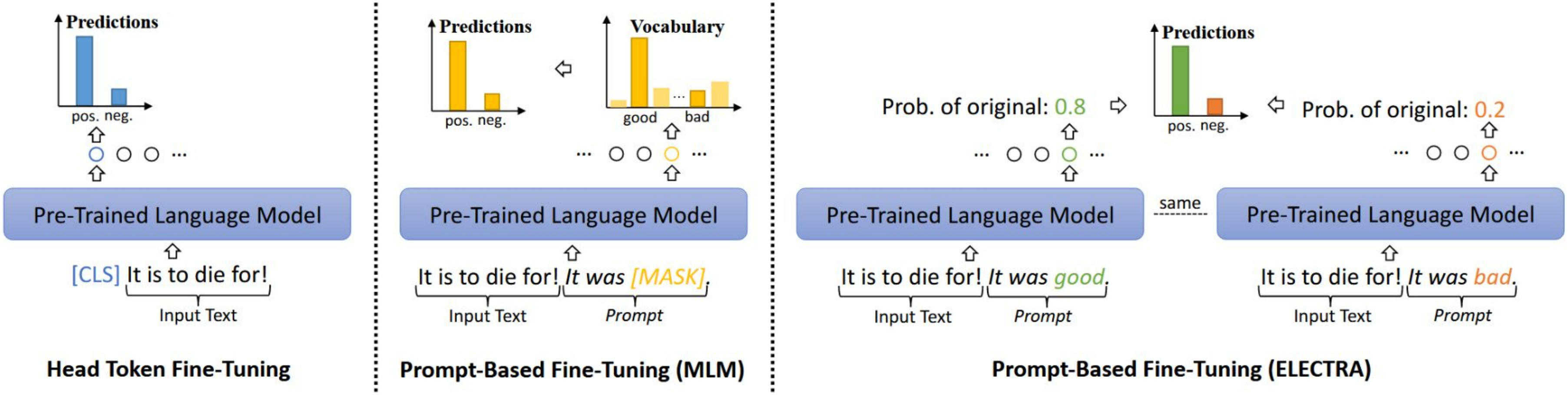
Any **labeled documents** are not allowed, **suface names** or **limited word-level descriptions** of each category can be used.



Some pseudo label is not correct

Fine-Tuning

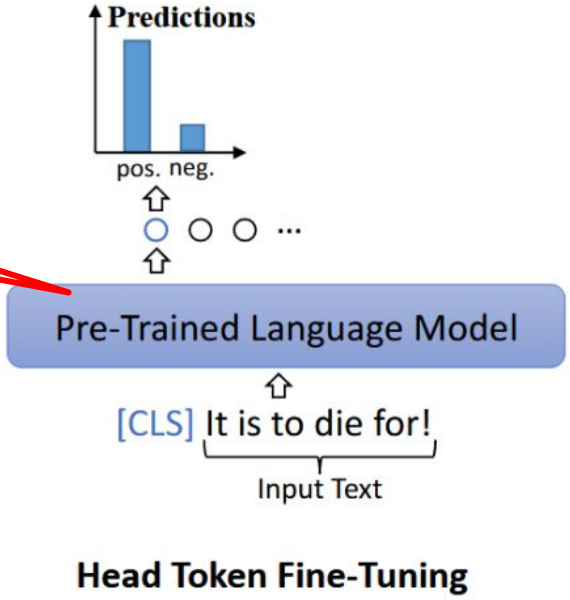
Type of fine tuning



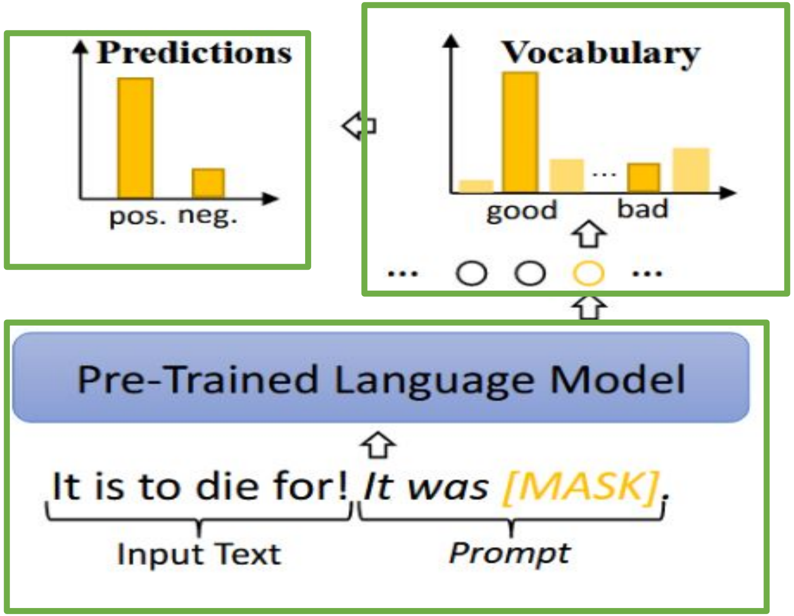
Head Token Fine-Tuning

Classifier

$$p(c|d) = \textbf{Softmax}(g(\mathbf{h}^{\text{CLS}}))$$



Prompt-Base Fine-Tuning(MLM)



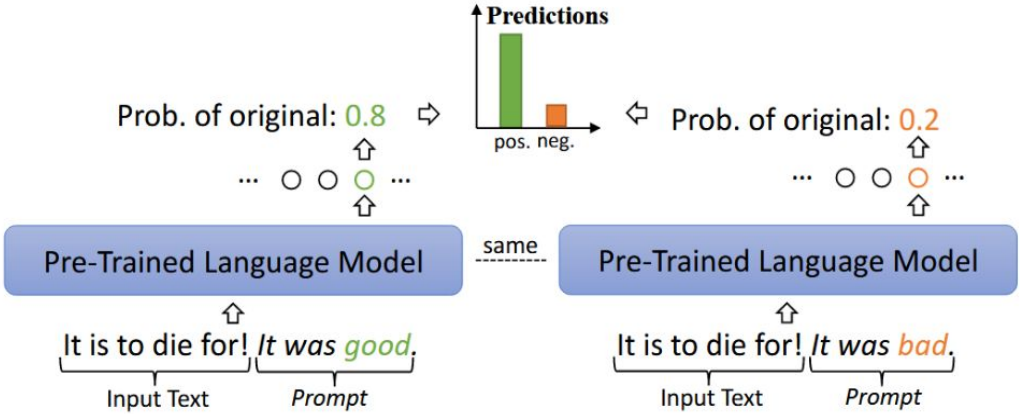
Prompt-Based Fine-Tuning (MLM)

$$\mathcal{T}^{\text{MLM}}(d) = d \text{ It was [MASK].}$$

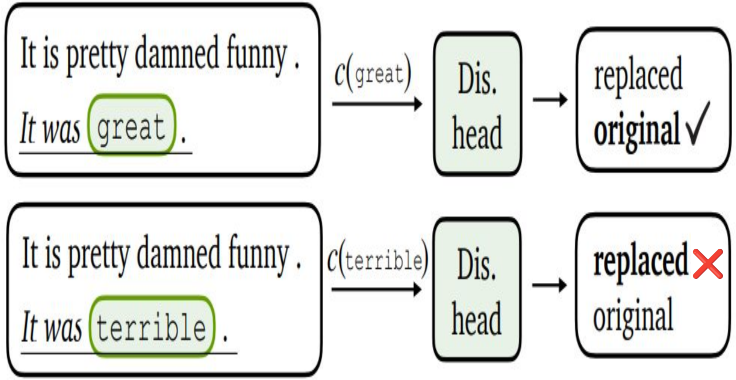
$$p(w|d) = \text{Softmax}(f(\mathbf{h}^{\text{MASK}})). \qquad (7)$$

$$p(l(c)|d)$$

Prompt-Base Fine-Tuning(ELECTRA)

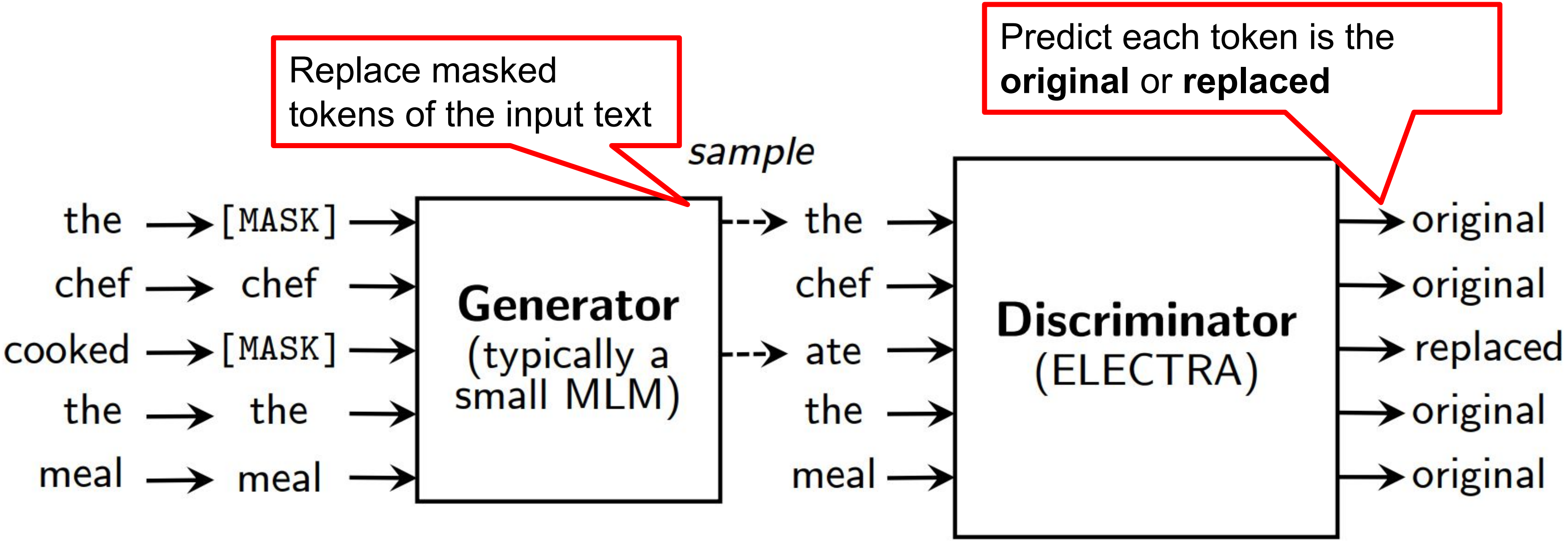


Prompt-Based Fine-Tuning (ELECTRA)

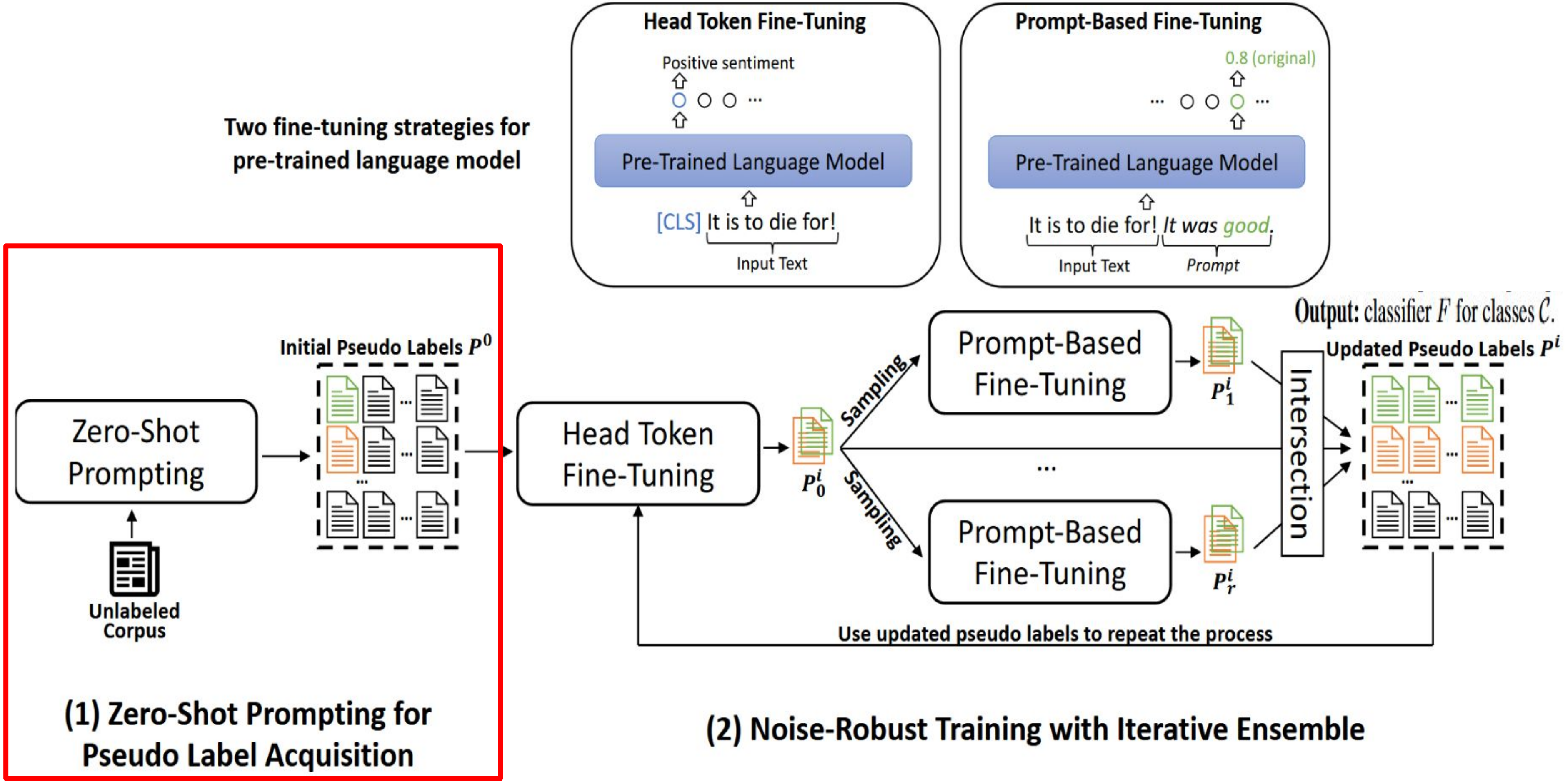


ELECTRA Pre-train model

Cast the word prediction problem into a binary classification problem



Method


Head Token Fine-Tuning
Positive sentiment
Pre-Trained Language Model
[CLS] It is to die for!
Input Text
Prompt-Based Fine-Tuning
0.8 (original)
Pre-Trained Language Model
It is to die for! It was good.
Input Text
Prompt
Two fine-tuning strategies for pre-trained language model
Zero-Shot Prompting
Unlabeled Corpus
Initial Pseudo Labels P^0
Head Token Fine-Tuning
P_0^i
Sampling
Sampling
Prompt-Based Fine-Tuning
Prompt-Based Fine-Tuning
P_1^i
P_r^i
Intersection
Output: classifier F for classes C.
Updated Pseudo Labels P^i
Use updated pseudo labels to repeat the process
(1) Zero-Shot Prompting for Pseudo Label Acquisition
(2) Noise-Robust Training with Iterative Ensemble

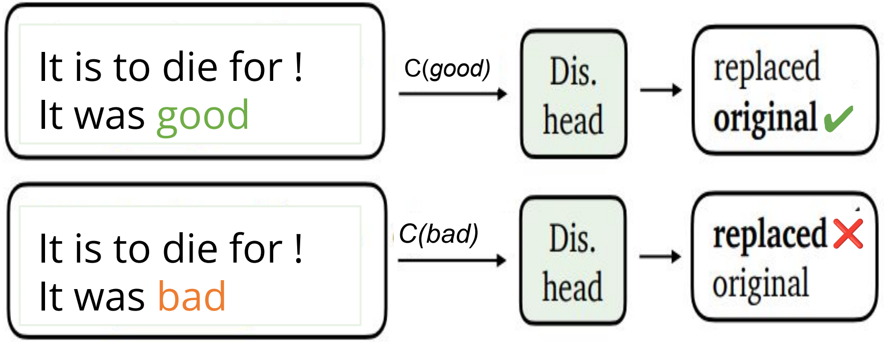
Method

Zero-Shot Prompting for Pseudo Label Acquisition

Construct input with the template

$$\mathcal{T}^{\text{ELECTRA}}(d, \text{good}) = d \text{ It was } \underline{\text{good}}.$$

$$\mathcal{T}^{\text{ELECTRA}}(d, \text{bad}) \quad = d \text{ It was } \underline{\text{bad}}.$$



Input: A corpus $\mathcal{D}$; a set of classes $\mathcal{C}$ and their label names $l(c), c \in \mathcal{C}$; a pre-trained language model E; a template $\mathcal{T}$ for prompting.

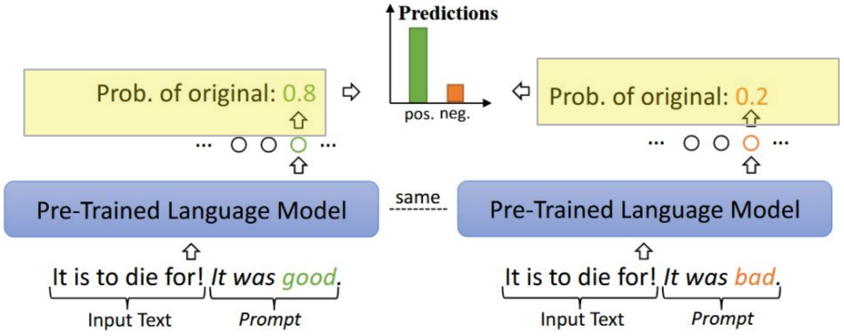
// Zero-Shot Prompting for Pseudo Label Acquisition;
for $d \in \mathcal{D}$ **do**
  for $c \in \mathcal{C}$ **do**
    $\mathcal{T}(d, l(c)) \leftarrow$ Construct input with the template;
    $p(l(c)|d) \leftarrow$ Prompt E with Eq. (1);
  $p(c|d) \leftarrow$ Eq. (2);
$P^0 \leftarrow$ top t^0 percentage of predictions;

Zero-Shot Prompting for Pseudo Label Acquisition

$$p(l(c)|d) = \mathrm{Sigmoid}(f(\mathbf{h}^{l(c)})), \quad (1)$$

$$p(c|d) = \frac{p(l(c)|d)}{\sum_{c' \in \mathcal{C}} p(l(c')|d)}. \quad (2)$$

$$P^0 = topk(p(c|d))$$



Prompt-Based Fine-Tuning (ELECTRA)

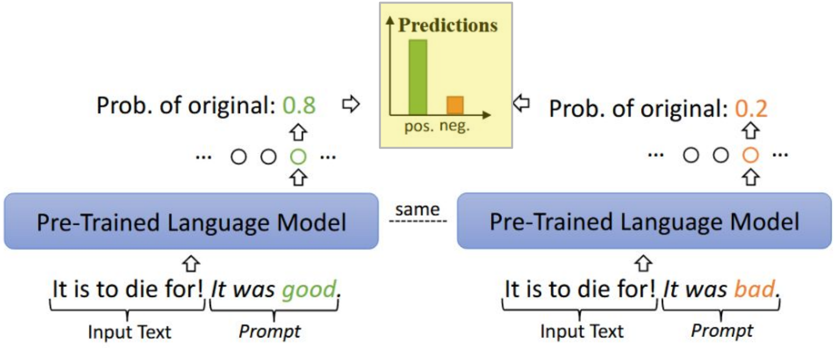
// Zero-Shot Prompting for Pseudo Label Acquisition;
for $d \in \mathcal{D}$ **do**
  for $c \in \mathcal{C}$ **do**
    $\mathcal{T}(d, l(c)) \leftarrow$ Construct input with the template;
    $p(l(c)|d) \leftarrow$ Prompt E with Eq. (1);
  $p(c|d) \leftarrow$ Eq. (2);
$P^0 \leftarrow$ top t^0 percentage of predictions;

Zero-Shot Prompting for Pseudo Label Acquisition

$$p(l(c)|d) = \text{Sigmoid}(f(\mathbf{h}^{l(c)})), \quad (1)$$

$$p(c|d) = \frac{p(l(c)|d)}{\sum_{c' \in \mathcal{C}} p(l(c')|d)}. \quad (2)$$

$$P^0 = topk(p(c|d))$$



Prompt-Based Fine-Tuning (ELECTRA)

// Zero-Shot Prompting for Pseudo Label Acquisition;
for $d \in \mathcal{D}$ **do**
  for $c \in \mathcal{C}$ **do**
    $\mathcal{T}(d, l(c)) \leftarrow$ Construct input with the template;
    $p(l(c)|d) \leftarrow$ Prompt E with Eq. (1);
  $p(c|d) \leftarrow$ Eq. (2);
$P^0 \leftarrow$ top t^0 percentage of predictions;

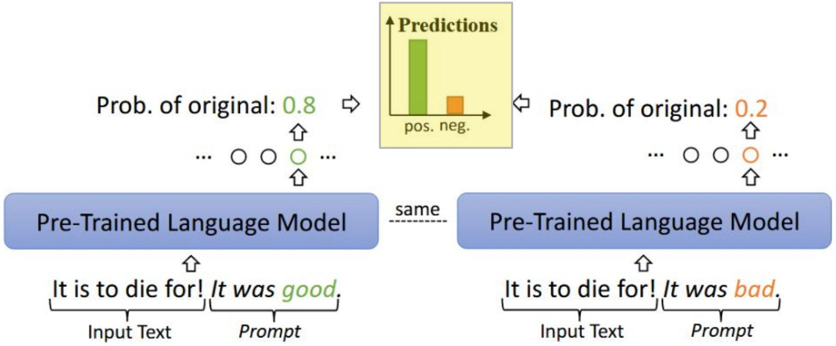
Method

Zero-Shot Prompting for Pseudo Label Acquisition

$$p(l(c)|d) = \mathrm{Sigmoid}(f(\mathbf{h}^{l(c)})), \qquad (1)$$

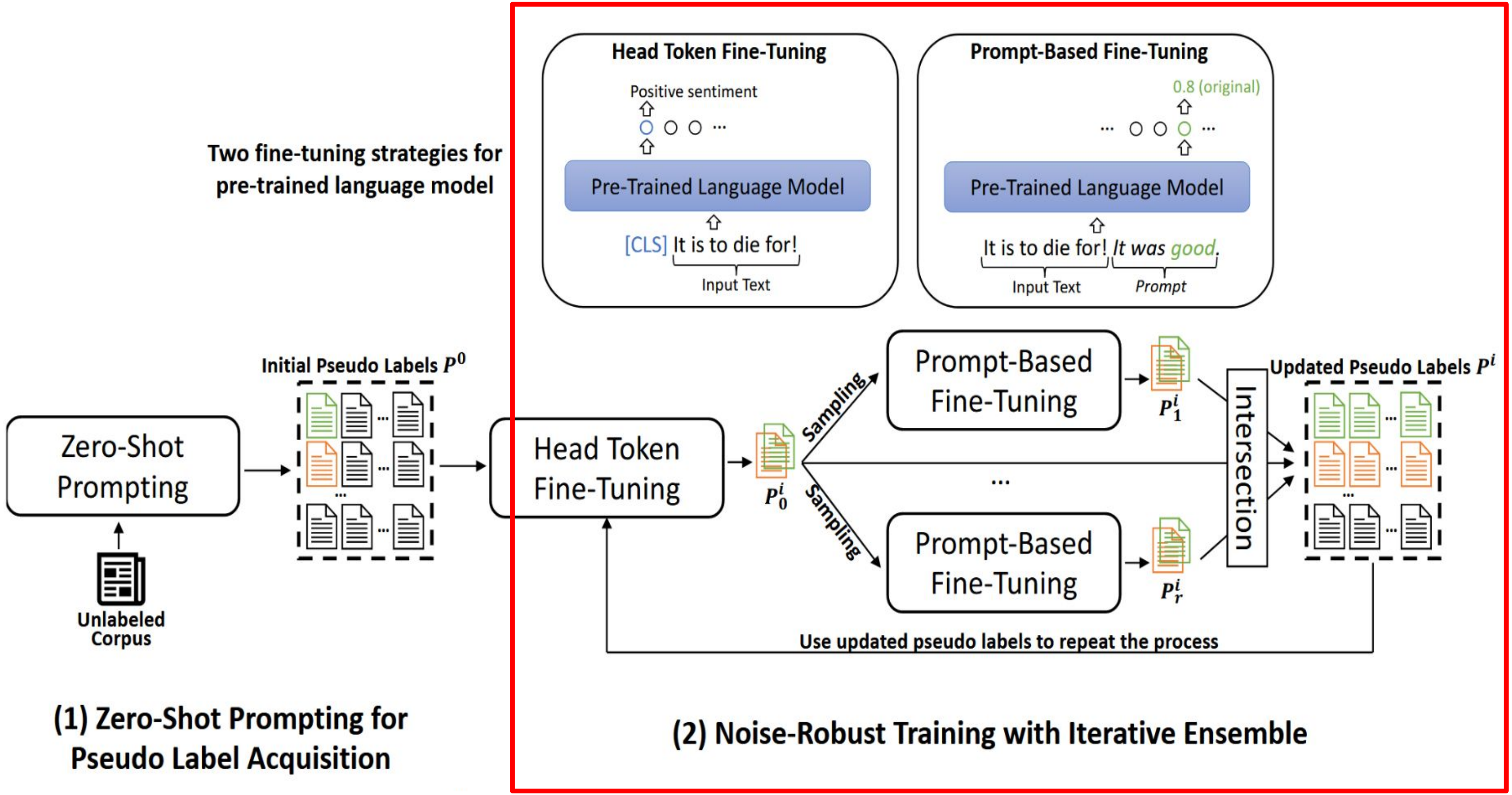
$$p(c|d) = \frac{p(l(c)|d)}{\sum_{c' \in \mathcal{C}} p(l(c')|d)}. \qquad (2)$$

$$\boxed{P^0 = topk(p(c|d))}$$



Prompt-Based Fine-Tuning (ELECTRA)

// Zero-Shot Prompting for Pseudo Label Acquisition;
for $d \in \mathcal{D}$ **do**
  for $c \in \mathcal{C}$ **do**
    $\mathcal{T}(d, l(c)) \leftarrow$ Construct input with the template;
    $p(l(c)|d) \leftarrow$ Prompt E with Eq. (1);
  $p(c|d) \leftarrow$ Eq. (2);
$P^0 \leftarrow$ top t^0 percentage of predictions;

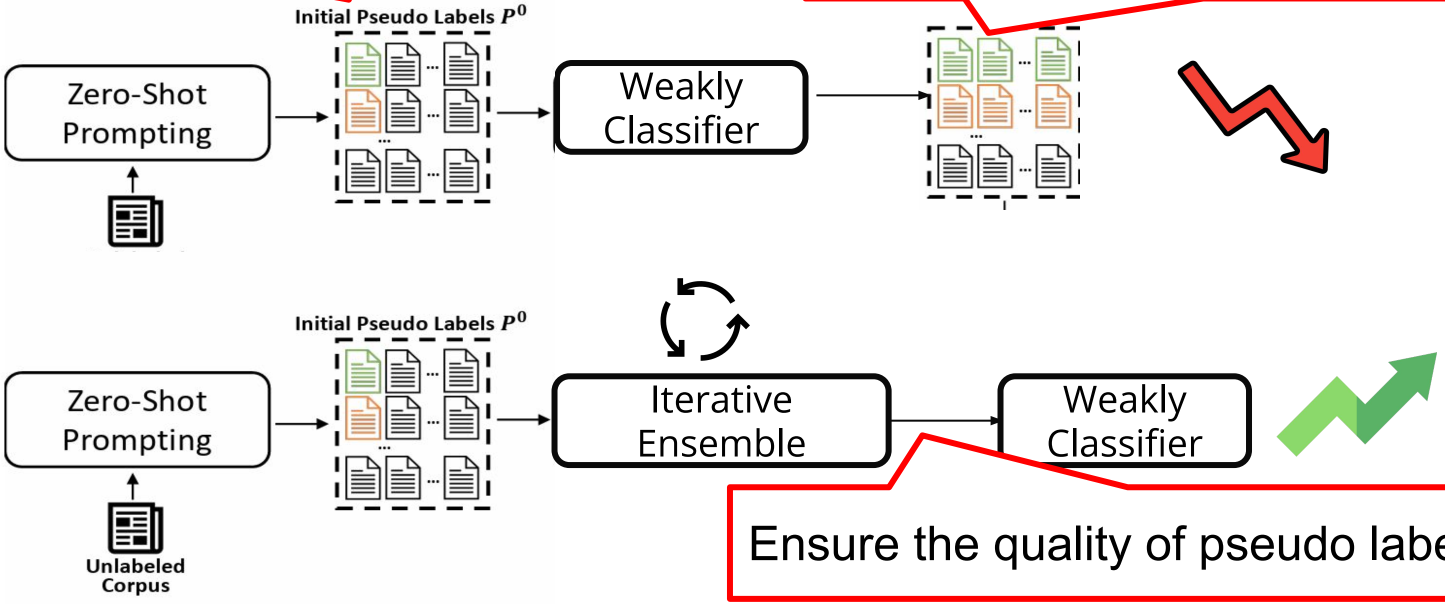

Two fine-tuning strategies for pre-trained language model
Head Token Fine-Tuning
Positive sentiment
Pre-Trained Language Model
[CLS] It is to die for!
Input Text
Prompt-Based Fine-Tuning
0.8 (original)
Pre-Trained Language Model
It is to die for! It was good.
Input Text
Prompt
Zero-Shot Prompting
Unlabeled Corpus
Initial Pseudo Labels P^0
Head Token Fine-Tuning
P_0^i
Sampling
Sampling
Prompt-Based Fine-Tuning
Prompt-Based Fine-Tuning
P_1^i
P_r^i
Intersection
Updated Pseudo Labels P^i
Use updated pseudo labels to repeat the process
(1) Zero-Shot Prompting for Pseudo Label Acquisition
(2) Noise-Robust Training with Iterative Ensemble

Method

Noise-Robust Training with Iterative Ensemble

pseudo labelsnoisy range (**15%-50%)**

Class result decrease by noise
(large gap between fully-supervised)

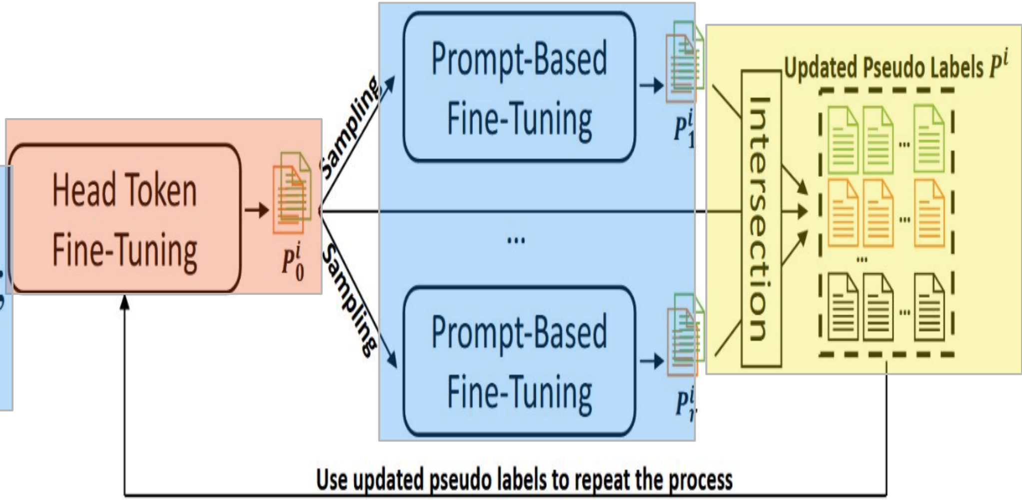


Ensure the quality of pseudo labels

Noise-Robust Training with Iterative Ensemble

for $i \leftarrow 1$ *to* T **do**

- $F_0^i \leftarrow$ Head token fine-tuning using P^{i-1};
- $P_0^i \leftarrow$ Select top t_i predictions by F_0^i;
- $S \leftarrow$ Randomly sample r subsets of P_0^i;
- **for** $S_k \in S$ **do**
  - $F_k^i \leftarrow$ Prompt-based fine-tuning using S_k;
  - $P_k^i \leftarrow$ Select top t_i percentage by F_k^i;
- $P^i \leftarrow$ Eq. (4);



Noise-Robust Training with Iterative Ensemble

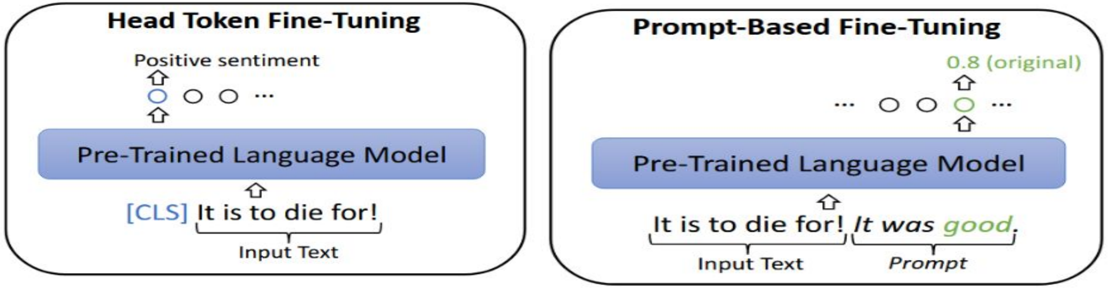
Utilize two PLM fine-tuning methods to ensure the quality of pseudo labels

improve the self-training quality

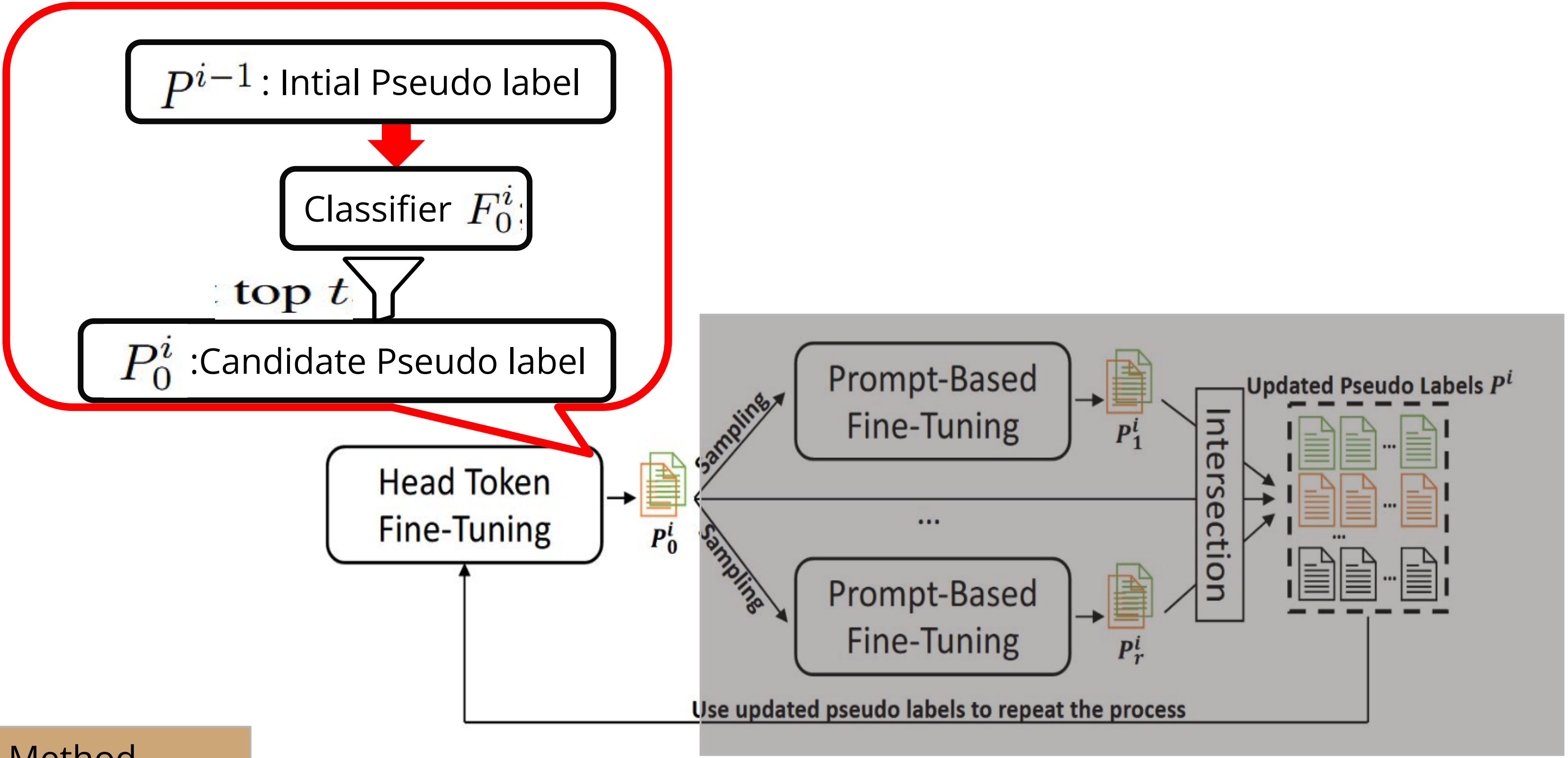
1. **Head token fine-tuning**: Capturing the information of the entire document

2. **Prompt-based finetuning:** Focusing more on the context surrounding the

Two fine-tuning strategies for pre-trained language model

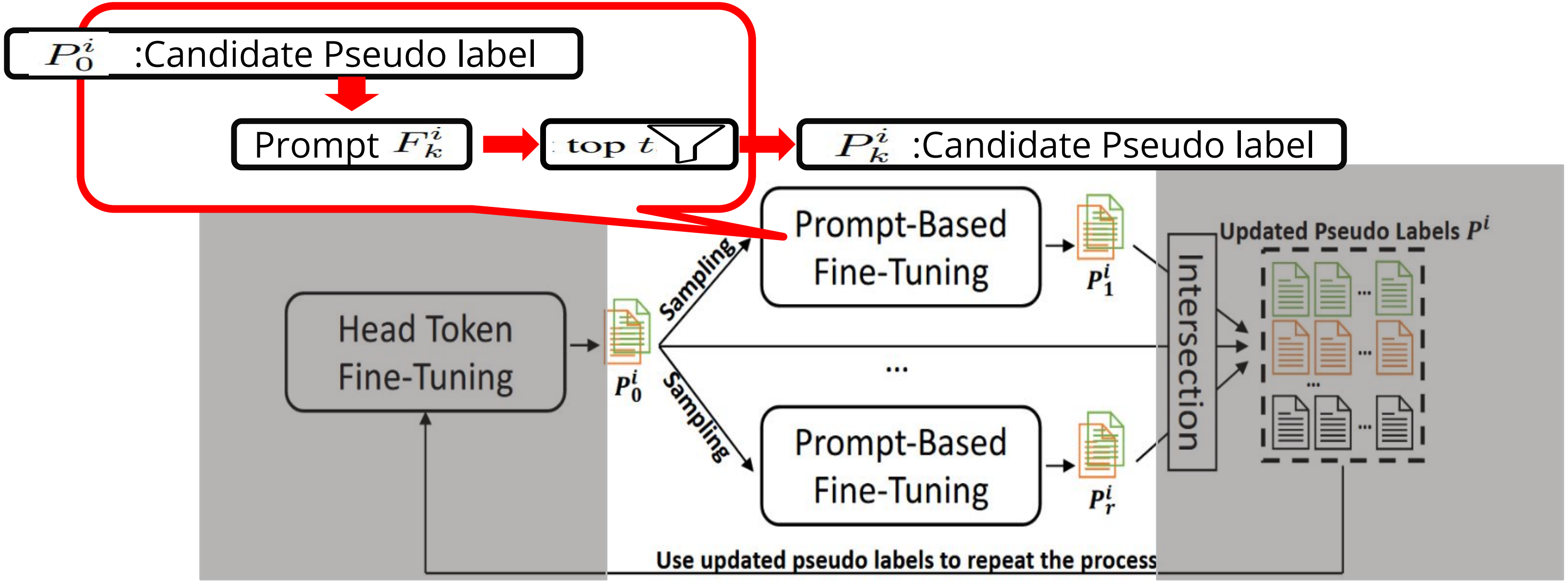


Noise-Robust Training with Iterative Ensemble



Noise-Robust Training with Iterative Ensemble

Prompt base only requires a small amount of data to achieve competitive performance with head token fine-tuning

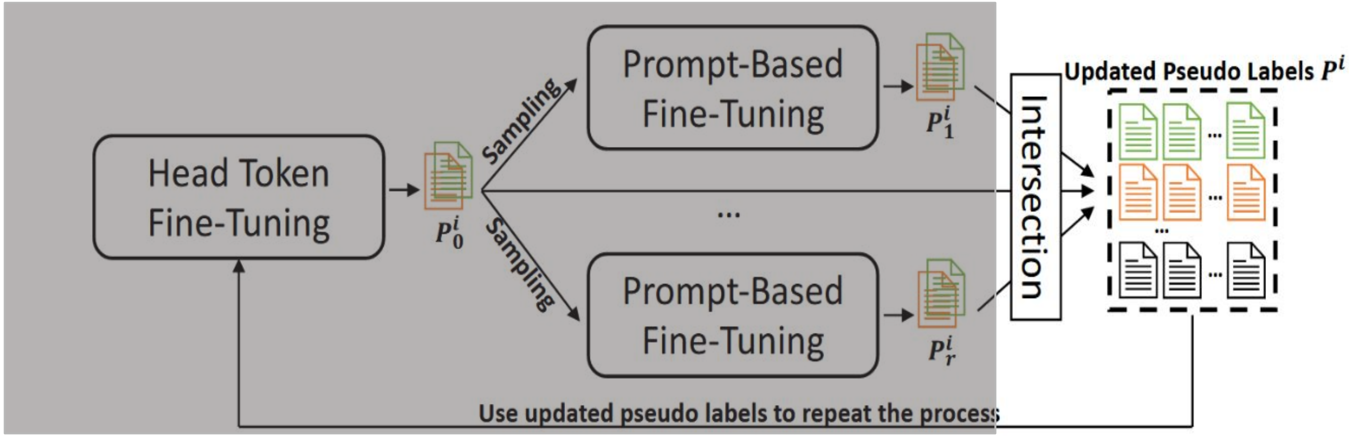


Noise-Robust Training with Iterative Ensemble

Only those most confident ones into the pseudo label pool to alleviate the error accumulation problem.

P_k^i :Candidate Pseudo label → Intersection →

$$\mathcal{P}^i = \bigcap_{k=0} \mathcal{P}^i_k. \tag{4}$$



Experiment

DataSet

- Topic
    - Ag_News(New topic with 4 class)
    - 20_News (New topic with 20 class)
    - NYT-Topics (New York Times context: imbalanced with 9 class)
    - NYT-Fine (New York Times context: imbalanced & fine-grained with 9 class)

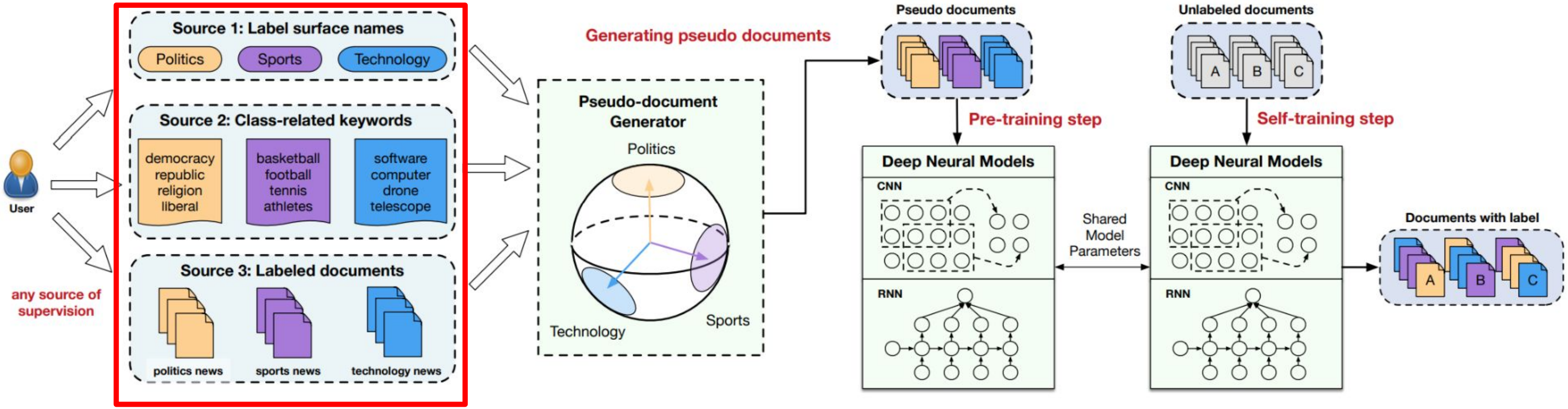
- Semantic(with 2 class)
    - Yelp(Review:Semantic analysis)
    - IMDB(Movie Review: semantic analysis)
    - Amazon(Amazon Review:semantic analysis)

Compared Methods

- Weakly method compare
    - WeSTClass
    - ConWea
    - LOTClass
    - XClass
    - ClassKG
- Pre-train model compare
    - RoBERTa (0-shot):Head Token
    - ELECTRA (0-shot):Head Token
    - Fully- Supervised BERT baseline

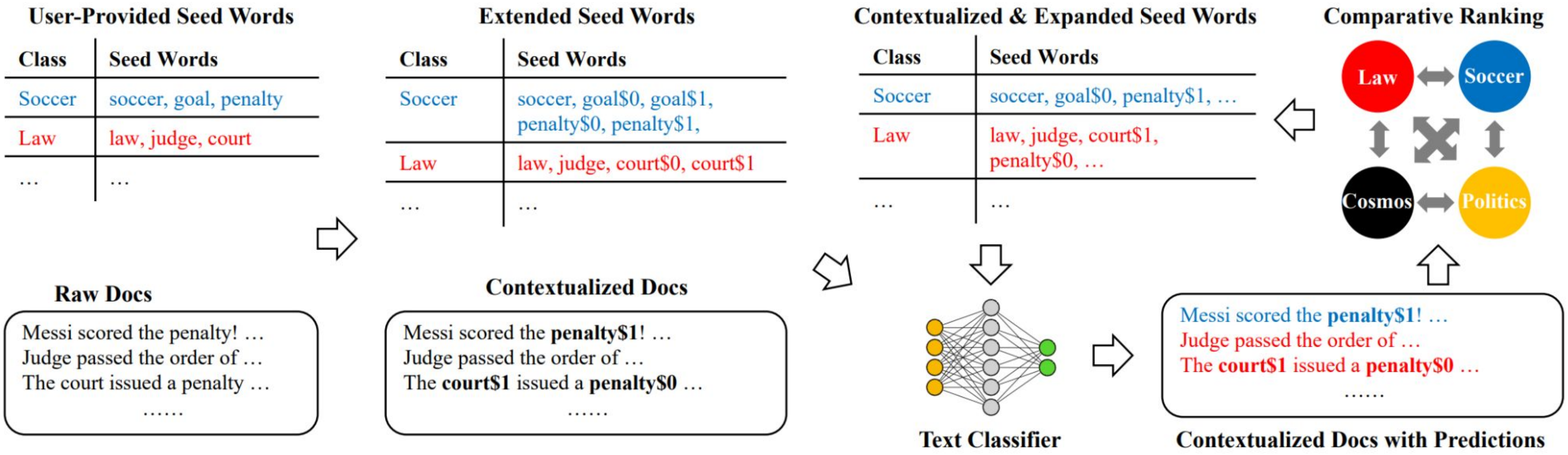
WeSTClass

Define the source of weakly supervision



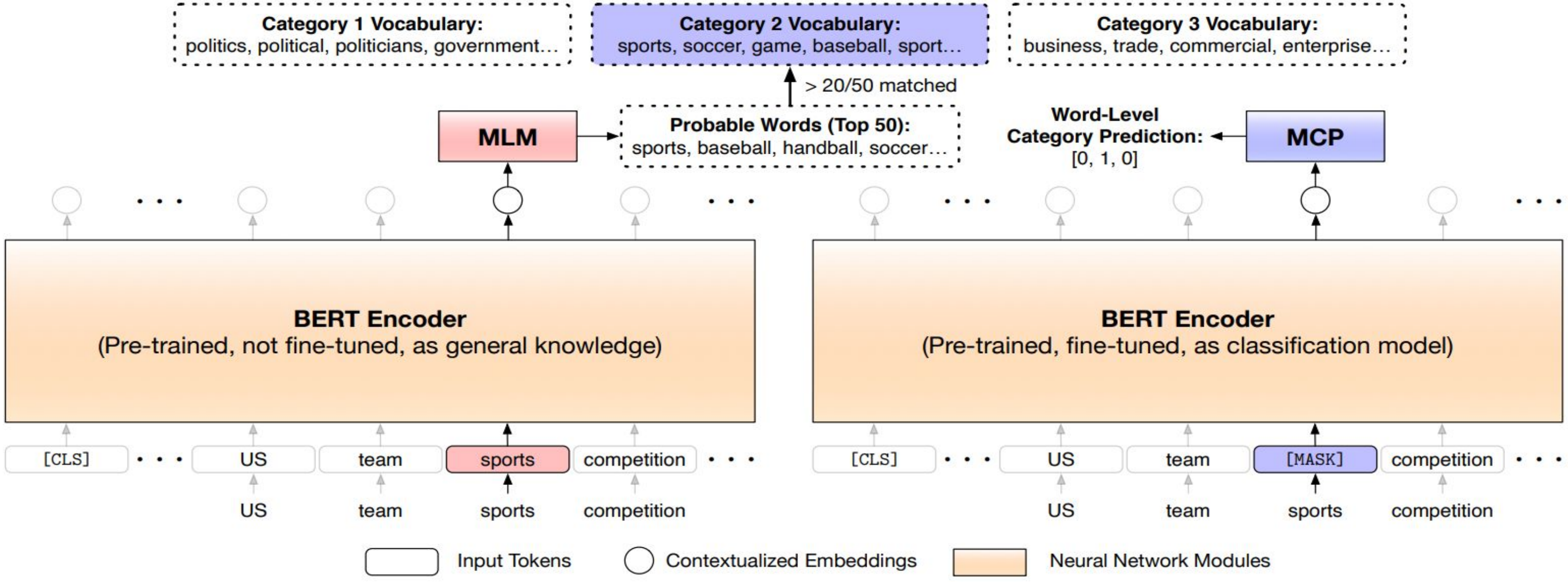
ConWea

Source.2



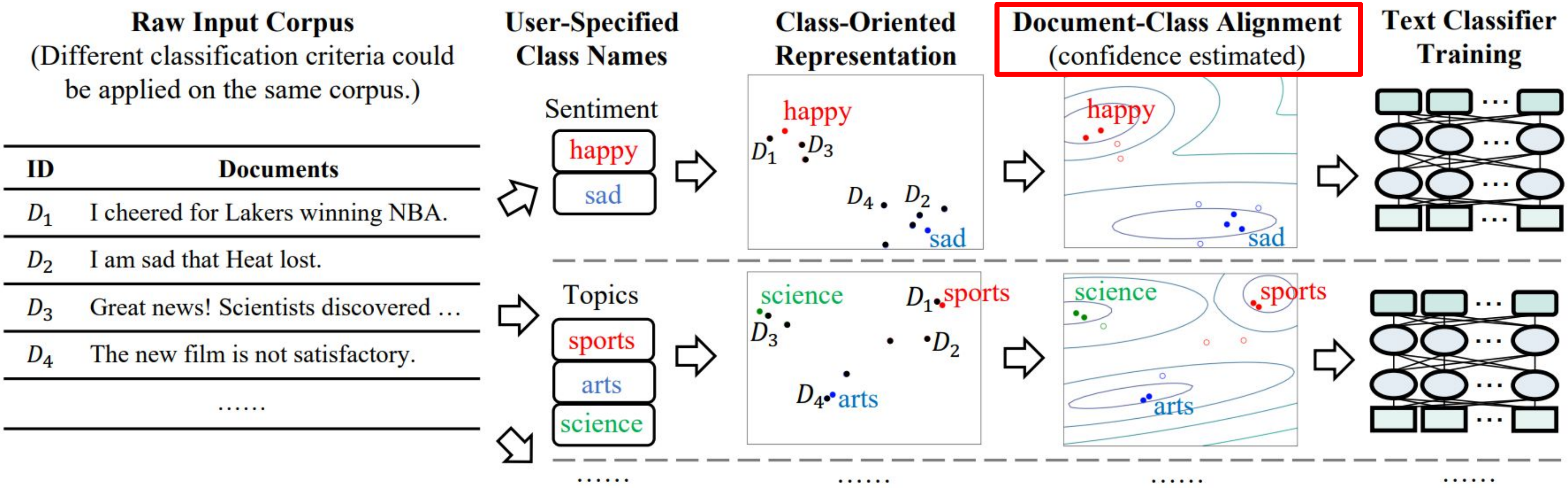
LOTClass

Source.1



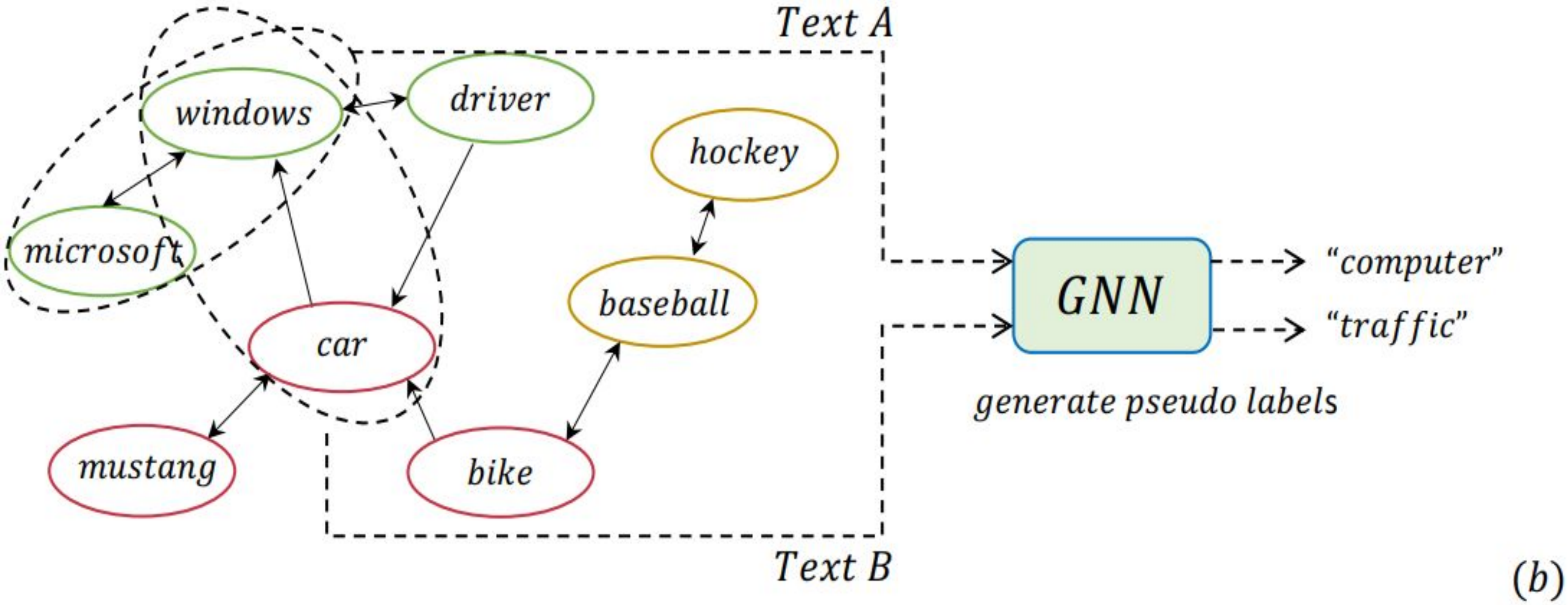
XClass

Source 1



ClassKG

Source1

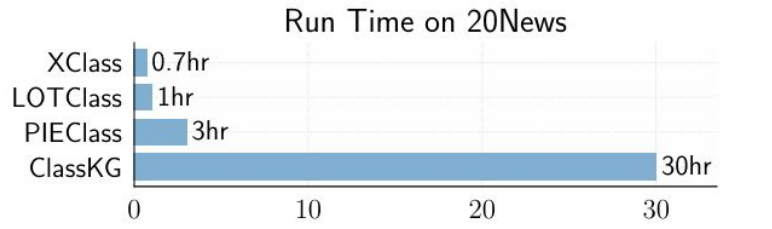


Compared Methods

Although ClassKG achieves the better results ClassKG uses more time

Methods	AGNews	20News	NYT-Topics	NYT-Fine	Yelp	IMDB	Amazon
WeSTClass	0.823/0.821	0.713/0.699	0.683/0.570	0.739/0.651	0.816/0.816	0.774/-	0.753/-
ConWea	0.746/0.742	0.757/0.733	0.817/0.715	0.762/0.698	0.714/0.712	-/-	-/-
LOTClass	0.869/0.868	0.738/0.725	0.671/0.436	0.150/0.202	0.878/0.877	0.865/-	0.916/-
XClass	0.857/0.857	0.786/0.778	0.790/0.686	0.857/0.674	0.900/0.900	-/-	-/-
ClassKG†	0.881/0.881	0.811/**0.820**	0.721/0.658	0.889/0.705	0.918/0.918	0.888/0.888	0.926/-
PIEClass ELECTRA+ELECTRA	0.884/0.884	**0.816**/0.817	**0.832/0.763**	**0.910/0.776**	**0.957/0.957**	**0.931/0.931**	**0.937/0.937**
Fully-Supervised	0.940/0.940	0.965/0.964	0.943/0.899	0.980/0.966	0.957/0.957	0.945/-	0.972/-

Micro-F1/Macro-F1



Compared Methods

Methods	AGNews	20News	NYT-Topics	NYT-Fine	Yelp	IMDB	Amazon
RoBERTa (0-shot)	0.581/0.529	0.507/0.445‡	0.544/0.382	-/-‡	0.812/0.808	0.784/0.780	0.788/0.783
ELECTRA (0-shot)	0.810/0.806	0.558/0.529	0.739/0.613	0.765/0.619	0.820/0.820	0.803/0.802	0.802/0.801
PIEClass							
ELECTRA+BERT	0.884/0.884	0.789/0.791	0.807/0.710	0.898/0.732	0.919/0.919	0.905/0.905	0.858/0.858
RoBERTa+RoBERTa	**0.895/0.895**	0.755/0.760‡	0.760/0.694	-/-‡	0.920/0.920	0.906/0.906	0.912/0.912
ELECTRA+ELECTRA	0.884/0.884	**0.816**/0.817	**0.832/0.763**	**0.910/0.776**	**0.957/0.957**	**0.931/0.931**	**0.937/0.937**
Fully-Supervised	0.940/0.940	0.965/0.964	0.943/0.899	0.980/0.966	0.957/0.957	0.945/-	0.972/-

Micro-F1/Macro-F1

Ablation Study

- **Two-Stage**:Directly trains classifier using pseudo labels from zero-shot prompting

- **Single-View ST:** Standard self-training method(only using zero-shot pseudo label)

- **Co-Training:** W/O Regularize in step Intersection

Ablation Study

- The single-view and two-stage method is not stable.
- Co-training ensures the consistency of model predictions, yielding great results.

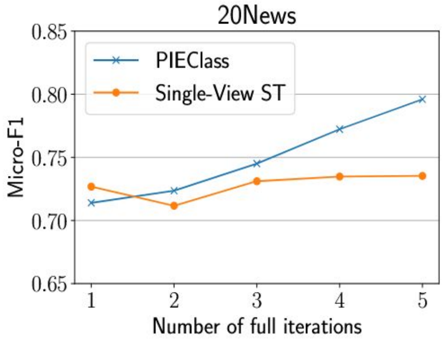
Methods	AGNews	20News	NYT-Topics	NYT-Fine	Yelp	IMDB	Amazon
Two-Stage	0.847/0.847	0.739/0.733	0.776/0.664	0.838/0.678	0.913/0.913	0.870/0.870	0.836/0.835
Single-View ST	0.871/0.871	0.736/0.737	0.757/0.668	0.853/0.695	0.912/0.912	0.846/0.846	0.892/0.892
Co-Training	0.877/0.877	0.795/0.791	0.818/0.715	0.877/0.744	0.948/0.948	0.925/0.925	0.930/0.930
PIEClass	**0.884/0.884**	**0.816/0.817**	**0.832/0.763**	**0.910/0.776**	**0.957/0.957**	**0.931/0.931**	**0.937/0.937**

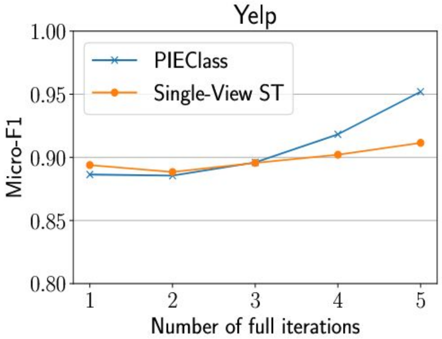
Micro-F1/Macro-F1

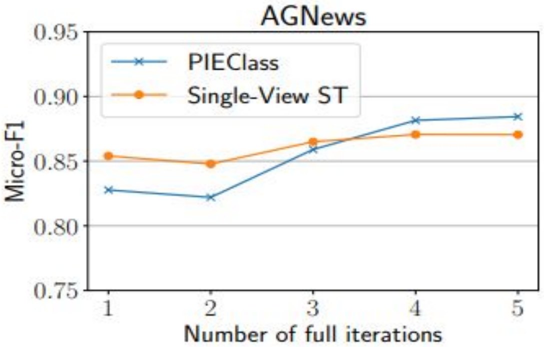
Ablation Study

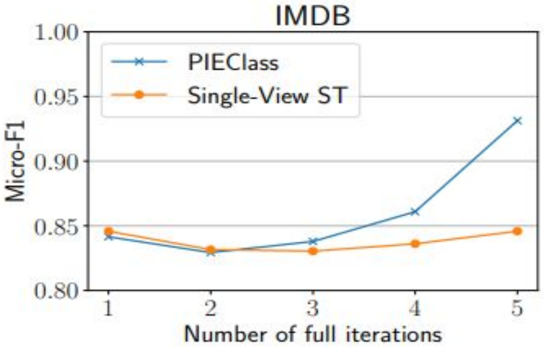
The PIEClass can surpass the bottleneck of traditional self-learning.

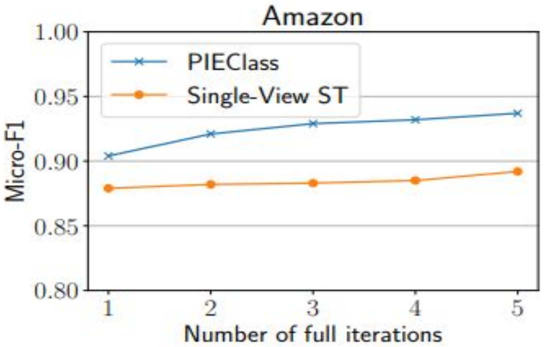
Traditional self-learning micor-f1 will be flattened after several iterations.





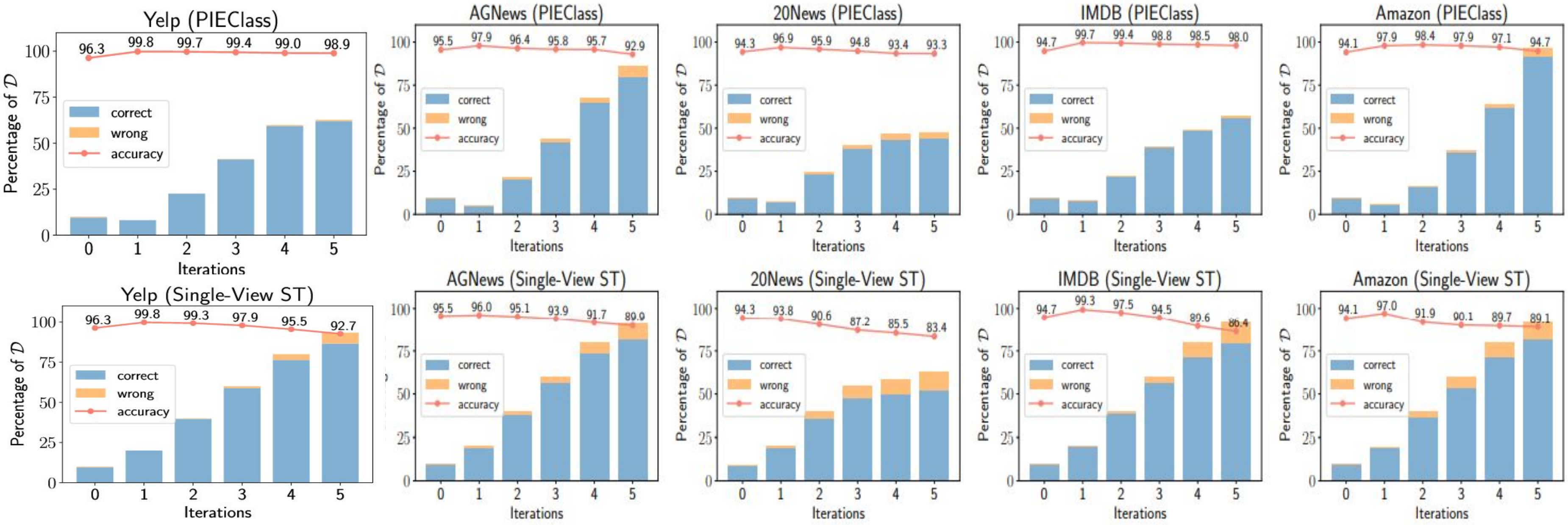






Quantities and qualities of the pseudo labels

We can see at the **first servals iteration** the pseudo label qualities in well.



Conclusion

Conclusion

1. Using zero-shot PLM prompting to assign pseudo labels based on contextualized text understanding.

2. Implementing a noise-robust iterative ensemble to expand pseudo labels while ensuring their quality.

Personal Comment

- In this paper, the noise-robust approach is crucial. Fully embracing it could significantly improve model adaptability in noisy environments.