Suman Acharya

Q1) Use the CoNLL-2003 Named Entity Recognition dataset which contains four entity types: PER (Person names) LOC (Locations) ORG (Organizations) MISC (Miscellaneous entities) Load the CoNLL-2003 dataset using HuggingFace datasets (https://huggingface.co/datasets/eriktks/conll2003) and initialize a Weights & Biases project called "Q1-weak-supervision-ner". Log the dataset statistics (number of samples, entity distribution) to W&B as summary metrics

```
(Assignment5_mlops) PS D:\Assignment5_mlops> python Q1.py
/ag4kog1d
Number of training samples: 14987
Number of validation samples: 3684
Number of test samples: 3466
Entity distribution: {'O': 170524, 'B-ORG': 6321, 'B-MISC': 3438, 'B-PER': 6600, 'I-PER': 4528, 'B-LOC': 7
140, 'I-ORG': 3704, 'I-MISC': 1155, 'I-LOC': 1157}
wandb:
wandb: Run summary:
wandb:        num_test_samples 3466
wandb:       num_train_samples 14987
wandb: num_validation_samples 3684
wandb:
wandb:  View run conll2003-local-stats at: https://wandb.ai/142402011-indian-institute-of-technology/Q1-we
ak-supervision-ner/runs/ag4kog1d
wandb:  View project at: https://wandb.ai/142402011-indian-institute-of-technology/Q1-weak-supervision-ner
wandb: Synced 5 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
wandb: Find logs at: .\wandb\run-20251013_125300-ag4kog1d\logs
(Assignment5_mlops) PS D:\Assignment5_mlops>
```

Summary

Summary metrics are your model's outputs. **Learn more**

🔍 Search keys with regex

▼ **Summary metrics:** {} 12 keys

entity_distribution.B-LOC: 7,140

entity_distribution.B-MISC: 3,438

entity_distribution.B-ORG: 6,321

entity_distribution.B-PER: 6,600

entity_distribution.I-LOC: 1,157

entity_distribution.I-MISC: 1,155

entity_distribution.I-ORG: 3,704

entity_distribution.I-PER: 4,528

entity_distribution.O: 170,524

num_test_samples: 3,466

num_train_samples: 14,987

num_validation_samples: 3,684

Q2) Implement two basic labeling functions using Snorkel AI: a. A heuristic function detecting years (1900-2099) as potential DATE/MISC entities b. A pattern-matching function identifying organizations by common suffixes ("Inc.", "Corp.", "Ltd.") Log each labeling function's coverage and accuracy to W&B using wandb.log()

```
OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                                    >_ powershell  + ∨

(Assignment5_mlops) PS D:\Assignment5_mlops> python Q2.py
wandb: Syncing run conll2003-lf-analysis-int
wandb:  View project at https://wandb.ai/142402011-indian-institute-of-technology/Q1-weak-supervision-ner
wandb:  View run at https://wandb.ai/142402011-indian-institute-of-technology/Q1-weak-supervision-ner/runs/30d70a3o
100%|                                                                 | 14987/14987 [00:01<00
Labeling Function Summary:
               j Polarity  Coverage  Overlaps  Conflicts
lf_year        0      [0]  0.095549  0.000133   0.000133
lf_org_suffix  1      [1]  0.001201  0.000133   0.000133
wandb:
wandb: Run history:
wandb: lf_org_suffix_coverage _
wandb:       lf_year_coverage _
wandb:
wandb: Run summary:
wandb: lf_org_suffix_coverage 0.0012
wandb:       lf_year_coverage 0.09555
wandb:
```

Q3) Implement Snorkell's Label aggregation (Majority Label Voter)

```
       agg_counts = df_train["aggregated_label"] value_counts()
OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                              >_ powershell  + ∨  ⬚ 🗑 ··

(Assignment5_mlops) PS D:\Assignment5_mlops> python Q3.py
100%|                                                            | 14987/14987 [00:00<00:00, 63489
Labeling Function Summary:
               j Polarity  Coverage  Overlaps  Conflicts
lf_year        0      [0]  0.095549  0.000133   0.000133
lf_org_suffix  1      [1]  0.001201  0.000133   0.000133

Aggregated Label Distribution:
aggregated_label
-1    13541
 0     1430
 1       16
Name: count, dtype: int64
wandb:
wandb: Run history:
wandb: lf_org_suffix_coverage _
wandb:       lf_year_coverage _
wandb:
wandb: Run summary:
wandb: lf_org_suffix_coverage 0.0012
wandb:       lf_year_coverage 0.09555
wandb:
wandb:  View run conll2003-lf-aggregation at: https://wandb.ai/142402011-indian-institute-of-technology/Q1-weak-supervision-ner/runs/sk32gd8i
wandb:  View project at: https://wandb.ai/142402011-indian-institute-of-technology/Q1-weak-supervision-ner
wandb: Synced 5 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
wandb: Find logs at: .\wandb\run-20251013_145703-sk32gd8i\logs
 (Assignment5_mlops) PS D:\Assignment5_mlops> |
                                                  Ln 92, Col 1   Spaces: 4   UTF-8   CRLF   {} Python   🔔
```

Summary

Summary metrics are your model's outputs. **Learn more**

Q Search keys with regex

▼ **Summary metrics:** {} 5 keys

aggregated_label_distribution.-1: 13,541

aggregated_label_distribution.0: 1,430

aggregated_label_distribution.1: 16
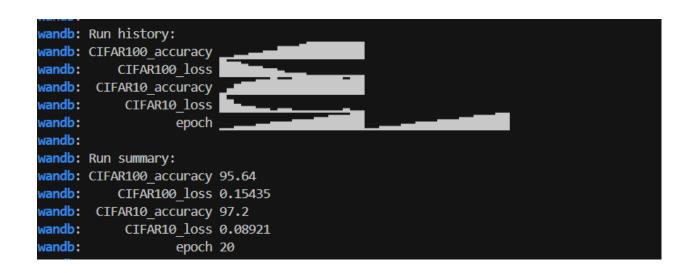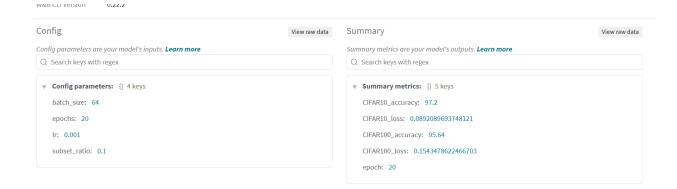
lf_org_suffix_coverage: 0.0012010409021151664

lf_year_coverage: 0.09554947621271768

Q4) Implement the following in Weights and Bias:
a. Train CIFAR 100 and CIFAR 10 sequentially for 100 epochs

```
OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

(Assignment5_mlops) PS D:\Assignment5_mlops> python Q4.py

Training on CIFAR-100...
[CIFAR100] Epoch 1/20  | Loss: 4.368 | Acc: 5.46%
[CIFAR100] Epoch 2/20  | Loss: 3.800 | Acc: 11.20%
[CIFAR100] Epoch 3/20  | Loss: 3.481 | Acc: 16.42%
[CIFAR100] Epoch 4/20  | Loss: 3.172 | Acc: 20.96%
[CIFAR100] Epoch 5/20  | Loss: 2.832 | Acc: 27.42%
[CIFAR100] Epoch 6/20  | Loss: 2.449 | Acc: 34.46%
[CIFAR100] Epoch 7/20  | Loss: 2.090 | Acc: 42.98%
[CIFAR100] Epoch 8/20  | Loss: 1.763 | Acc: 50.30%
[CIFAR100] Epoch 9/20  | Loss: 1.240 | Acc: 64.40%
[CIFAR100] Epoch 10/20 | Loss: 1.013 | Acc: 70.48%
[CIFAR100] Epoch 11/20 | Loss: 0.867 | Acc: 74.98%
[CIFAR100] Epoch 12/20 | Loss: 0.623 | Acc: 81.78%
[CIFAR100] Epoch 13/20 | Loss: 0.343 | Acc: 90.56%
[CIFAR100] Epoch 14/20 | Loss: 0.223 | Acc: 93.92%
[CIFAR100] Epoch 15/20 | Loss: 0.262 | Acc: 92.56%
[CIFAR100] Epoch 16/20 | Loss: 0.255 | Acc: 92.90%
[CIFAR100] Epoch 17/20 | Loss: 0.296 | Acc: 92.04%
[CIFAR100] Epoch 18/20 | Loss: 0.327 | Acc: 90.58%
[CIFAR100] Epoch 19/20 | Loss: 0.365 | Acc: 89.48%
[CIFAR100] Epoch 20/20 | Loss: 0.154 | Acc: 95.64%
```

```
(Assignment5_mlops) PS D:\Assignment5_mlops> python Q4.py
Fine-tuning on CIFAR-10...
[CIFAR10] Epoch 1/20  | Loss: 1.652 | Acc: 41.90%
[CIFAR10] Epoch 2/20  | Loss: 1.094 | Acc: 60.52%
[CIFAR10] Epoch 3/20  | Loss: 0.755 | Acc: 73.26%
[CIFAR10] Epoch 4/20  | Loss: 0.572 | Acc: 80.54%
[CIFAR10] Epoch 5/20  | Loss: 0.404 | Acc: 85.94%
[CIFAR10] Epoch 6/20  | Loss: 0.295 | Acc: 89.76%
[CIFAR10] Epoch 7/20  | Loss: 0.190 | Acc: 93.40%
[CIFAR10] Epoch 8/20  | Loss: 0.130 | Acc: 96.04%
[CIFAR10] Epoch 9/20  | Loss: 0.340 | Acc: 89.72%
[CIFAR10] Epoch 10/20 | Loss: 0.264 | Acc: 91.16%
[CIFAR10] Epoch 11/20 | Loss: 0.106 | Acc: 96.66%
[CIFAR10] Epoch 12/20 | Loss: 0.136 | Acc: 95.82%
[CIFAR10] Epoch 13/20 | Loss: 0.103 | Acc: 96.66%
[CIFAR10] Epoch 14/20 | Loss: 0.068 | Acc: 97.84%
[CIFAR10] Epoch 15/20 | Loss: 0.048 | Acc: 98.46%
[CIFAR10] Epoch 16/20 | Loss: 0.041 | Acc: 98.94%
[CIFAR10] Epoch 17/20 | Loss: 0.104 | Acc: 96.62%
[CIFAR10] Epoch 18/20 | Loss: 0.207 | Acc: 93.40%
[CIFAR10] Epoch 19/20 | Loss: 0.141 | Acc: 95.46%
[CIFAR10] Epoch 20/20 | Loss: 0.089 | Acc: 97.20%
```
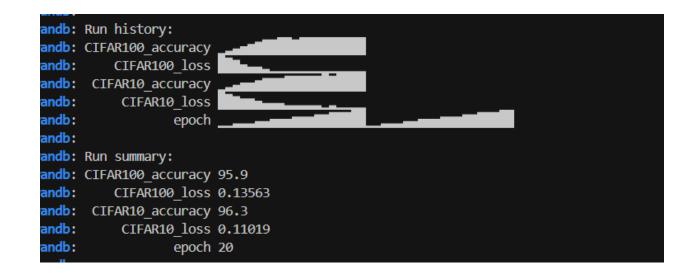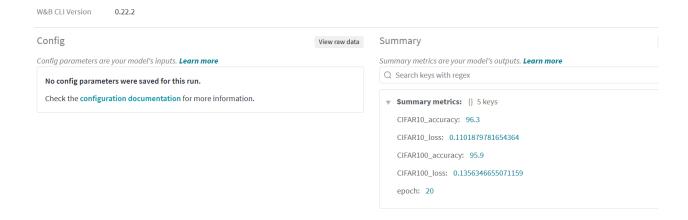
```
wandb: Run history:
wandb: CIFAR100_accuracy  ▁▂▃▄▅▆▇█
wandb:      CIFAR100_loss  █▇▆▅▄▃▂▁
wandb:   CIFAR10_accuracy  ▁▃▄▅▆▇██
wandb:       CIFAR10_loss  █▆▄▃▂▁▁▁
wandb:              epoch  ▁▂▃▄▅▆▇█
wandb:
wandb: Run summary:
wandb: CIFAR100_accuracy 95.64
wandb:      CIFAR100_loss 0.15435
wandb:   CIFAR10_accuracy 97.2
wandb:       CIFAR10_loss 0.08921
wandb:              epoch 20
```

Config   View raw data

Config parameters are your model's inputs. **Learn more**

Q Search keys with regex

▼ **Config parameters:** {} 4 keys

batch_size: 64

epochs: 20

lr: 0.001

subset_ratio: 0.1

Summary   View raw data

Summary metrics are your model's outputs. **Learn more**

Q Search keys with regex

▼ **Summary metrics:** {} 5 keys

CIFAR10_accuracy: 97.2

CIFAR10_loss: 0.0892089693748121

CIFAR100_accuracy: 95.64

CIFAR100_loss: 0.1543478622466703

epoch: 20

b. Train CIFAR 10 and CIFAR 100 sequentially for 100 epochs.

```
(Assignment5_mlops) PS D:\Assignment5_mlops> python Q4.py

Training on CIFAR-10...
[CIFAR10] Epoch 1/20 | Loss: 1.884 | Acc: 33.38%
[CIFAR10] Epoch 2/20 | Loss: 1.507 | Acc: 45.60%
[CIFAR10] Epoch 3/20 | Loss: 1.326 | Acc: 52.90%
[CIFAR10] Epoch 4/20 | Loss: 1.128 | Acc: 59.62%
[CIFAR10] Epoch 5/20 | Loss: 0.986 | Acc: 65.66%
[CIFAR10] Epoch 6/20 | Loss: 0.816 | Acc: 70.60%
[CIFAR10] Epoch 7/20 | Loss: 0.667 | Acc: 76.00%
[CIFAR10] Epoch 8/20 | Loss: 0.621 | Acc: 79.08%
[CIFAR10] Epoch 9/20 | Loss: 0.544 | Acc: 81.02%
[CIFAR10] Epoch 10/20 | Loss: 0.362 | Acc: 87.86%
[CIFAR10] Epoch 11/20 | Loss: 0.304 | Acc: 89.22%
[CIFAR10] Epoch 12/20 | Loss: 0.265 | Acc: 91.20%
[CIFAR10] Epoch 13/20 | Loss: 0.319 | Acc: 89.22%
[CIFAR10] Epoch 14/20 | Loss: 0.213 | Acc: 92.64%
[CIFAR10] Epoch 15/20 | Loss: 0.168 | Acc: 95.20%
[CIFAR10] Epoch 16/20 | Loss: 0.403 | Acc: 86.74%
[CIFAR10] Epoch 17/20 | Loss: 0.094 | Acc: 96.72%
[CIFAR10] Epoch 18/20 | Loss: 0.057 | Acc: 98.62%
[CIFAR10] Epoch 19/20 | Loss: 0.140 | Acc: 95.20%
[CIFAR10] Epoch 20/20 | Loss: 0.110 | Acc: 96.30%
```

```
Fine-tuning on CIFAR-100...
 [CIFAR100] Epoch 1/20  | Loss: 4.128 | Acc: 7.68%
 [CIFAR100] Epoch 2/20  | Loss: 3.299 | Acc: 20.12%
 [CIFAR100] Epoch 3/20  | Loss: 2.697 | Acc: 30.20%
 [CIFAR100] Epoch 4/20  | Loss: 2.081 | Acc: 43.58%
 [CIFAR100] Epoch 5/20  | Loss: 1.519 | Acc: 57.68%
 [CIFAR100] Epoch 6/20  | Loss: 1.046 | Acc: 69.20%
 [CIFAR100] Epoch 7/20  | Loss: 0.629 | Acc: 81.58%
 [CIFAR100] Epoch 8/20  | Loss: 0.395 | Acc: 89.08%
 [CIFAR100] Epoch 9/20  | Loss: 0.325 | Acc: 90.94%
 [CIFAR100] Epoch 10/20 | Loss: 0.227 | Acc: 93.64%
 [CIFAR100] Epoch 11/20 | Loss: 0.368 | Acc: 89.24%
 [CIFAR100] Epoch 12/20 | Loss: 0.307 | Acc: 90.84%
 [CIFAR100] Epoch 13/20 | Loss: 0.158 | Acc: 96.10%
 [CIFAR100] Epoch 14/20 | Loss: 0.176 | Acc: 94.86%
 [CIFAR100] Epoch 15/20 | Loss: 0.115 | Acc: 97.10%
 [CIFAR100] Epoch 16/20 | Loss: 0.145 | Acc: 96.16%
 [CIFAR100] Epoch 17/20 | Loss: 0.135 | Acc: 96.38%
 [CIFAR100] Epoch 18/20 | Loss: 0.159 | Acc: 95.98%
 [CIFAR100] Epoch 19/20 | Loss: 0.258 | Acc: 92.20%
 [CIFAR100] Epoch 20/20 | Loss: 0.136 | Acc: 95.90%
```

```
andb: Run history:
andb: CIFAR100_accuracy  ▁▅▆▆▇▇▇███
andb:       CIFAR100_loss  █▅▄▃▂▂▁▁▁▁
andb:   CIFAR10_accuracy  ▁▄▆▇▇████
andb:       CIFAR10_loss  █▄▃▂▂▁▁▁
andb:              epoch  ▁▂▂▃▃▄▄▅▆▇
andb:
andb: Run summary:
andb: CIFAR100_accuracy 95.9
andb:       CIFAR100_loss 0.13563
andb:   CIFAR10_accuracy 96.3
andb:       CIFAR10_loss 0.11019
andb:              epoch 20
```

Config   View raw data

Config parameters are your model's inputs. **Learn more**

> **No config parameters were saved for this run.**
>
> Check the **configuration documentation** for more information.

Summary

Summary metrics are your model's outputs. **Learn more**

🔍 Search keys with regex

▼ **Summary metrics:** {} 5 keys

   CIFAR10_accuracy: 96.3

   CIFAR10_loss: 0.1101879781654364

   CIFAR100_accuracy: 95.9

   CIFAR100_loss: 0.1356346655071159

   epoch: 20

Write your observations with experimental proof.

From the above two result we can see that when we train our model with CIFAR100 and then fine tuning it with CIFAR10 it gives better result and also faster convergence compare to we train with CIFAR10 and the fine tuning with CIFAR100 .

Reason

CIFAR-100 has more classes and complex
.features; pretraining learns diverse
representations that help CIFAR-10

CIFAR-10 features are simpler; transferring to
CIFAR-100 (more complex task) is harder