A Information for Dataset

A.1 Dataset Collection

Here we list the link to datasets used in our experiments.

- **CoNLL-03**: https://github.com/synalp/NER/tree/master/corpus/CoNLL-2003.
- **ACE05**: We are unable to provide the downloadable version due to it is not public. This corpus can be applied through the website of LDC: https://www.ldc.upenn.edu/collaborations/past-projects/ace.
- **Webpage**: Please refer the link in the paper (**?**).

A.2 Dataset Split

All the mentioned dataset has been split into **train/validate/test** set in the released version. We keep consistent with the validation set and the test set in our experiment. For the active learning paradigm, we split the training set as Table 1. The active learners are initialized on the seed set, then they implement 5 active learning rounds.

B Baseline Settings

For the baselines, we take random sampling and 3 active learning approaches – LC sampling, NTE sampling, and QBC sampling as Section **??**.

C Implementation Details of SeqMix

We implement bert-base-cased as the underlying model for the NER task and bert-base-multilingual-cased as the underlying model for the event detection task. We use the model from Huggingface Transformer codebase[1], and the repository[2] to fine-tune our model for sequence labeling task.

C.1 Number of Parameters

In our model, we use **bert-base-cased** and **bert-base-multilingual-cased** both of them occupy 12-layer, 768-hidden, 12-heads with 110M parameters.

[1]https://github.com/huggingface/transformers

[2]https://github.com/kamalkraj/BERT-NER

C.2 Adapting BERT for sequence labeling task

To fine-tune on sequence labeling tasks, a dropout layer ($p = 0.1$) and a linear (token-level) classification layer is built upon the pre-trained model.

C.3 SeqMix Details

In Section **??**, we construct a table of tokens $\mathcal{W}$ and their corresponding contextual embedding $\mathcal{E}$. For our underlying BERT model, we use the vocabulary provided by the tokenizer to build up $\mathcal{W}$, and the embedding initialized on the training set as $\mathcal{E}$.

We also need to construct a special token collection to exclude some generation in the process of sequence mixing. For example, BERT places token `[CLS]` and `[SEP]` at the starting position and the ending position for sentence, and pad the inputs with `[PAD]`. We exclude these disturbing tokens and the parent tokens.

C.4 Parameter Settings

The key parameters setting in our framework are stated here: (1) The number of active learning round is 5 for all the three datasets, but the size of seed set and the number of samples in each round differs from the dataset. We list the specific numbers as Table 1. (2) The sub-sequence window length s and the valid label density threshold η_0 vary from the datasets. For CoNLL-03, $s = 5$, $\eta_0 = 0.6$; for ACE05, $s = 5$, $\eta_0 = 0.2$; for WebPage, $s = 4$, $\eta_0 = 0.5$. (3) We set $\alpha = 8$ for the *Beta* distribution. (4) The discriminator score range is set as $(0, 500)$ for all the datasets. (5) For BERT configuration, we choose 5e-5 for learning rate, 128 for padding length, 32 for batch size, 0.1 for dropout rate, 1e-8 for ϵ in Adam. At each data usage point, we train the model for 10 Epochs. (6) We set $\mathcal{C} = 3$ for the QBC query policy.

D Details of Experiments

We take following criteria to evaluate the sequence labeling task. A named entity is correct only if it is an exact match of the corresponding entity in the data file. An event trigger is correct only if the span and type match with golden labels. Based on the above metric, we evaluate F_1 score in our experiments.

D.1 Performance on Development Set

Table 2 to Table 4 shows the model performance on the validation set. The data usage in these tables

Dataset	# of Entity Types	# of Seed Set	Sampling Rounds	# of Each Round Samples	# of Dev	# of Test
CoNLL-03	4	200	5	100	3250	3453
ACE05	29	1k	5	{1k, 2k, 2k, 4k, 4k}	873	711
Webpage	4	85	5	60	99	135

Table 1: The information for benchmarks in our experiments.

Data Usage	200	300	400	500	600	700
Random Sampling	69.03	83.28	84.93	85.50	85.79	86.62
LC Sampling	69.03	83.78	84.55	85.88	86.04	86.73
NTE Sampling	69.03	83.60	85.00	85.47	86.19	86.83
QBC Sampling	69.03	83.33	84.52	85.30	86.27	86.60
Sub-sequence mixup	81.69	85.28	85.95	86.52	87.07	87.44

Table 2: Validation F_1 of CoNLL-03

Data Usage	1000	2000	4000	6000	10000	14000
Random Sampling	48.16	59.10	63.13	64.95	66.23	67.12
LC Sampling	48.16	59.33	63.22	65.04	66.24	66.92
NTE Sampling	48.16	59.72	63.17	65.53	66.78	67.24
QBC Sampling	48.16	59.01	62.79	64.89	66.20	66.91
Sub-sequence mixup	56.51	61.62	63.65	65.83	67.54	67.98

Table 3: Validation F_1 of ACE05

Data Usage	85	145	205	265	325	385
Random Sampling	0	27.52	34.41	34.83	37.93	35.73
LC Sampling	0	28.84	32.88	34.22	38.78	38.11
NTE Sampling	0	22.44	34.81	33.74	36.59	38.27
QBC Sampling	0	23.88	32.18	34.17	36.56	35.66
Sub-sequence mixup	14.35	33.74	34.70	36.22	39.74	38.25

Table 4: Validation F_1 of WebPage

refers to the number of labeled data, excluding the augmentation data. Sub-sequence mixup is trained with $(1+\alpha)$ times data, where the α denotes the augment rate. Note that WebPage is a very limited dataset, there is a big difference between the performance on the validation set and the test set. We average each experiment by 5 times.

D.2 Computing Infrastructure

We implement our system on *Ubuntu 18.04.3 LTS* system. We run our experiments on an Intel(R) Xeon(R) CPU @ 2.30GHz and NVIDIA Tesla P100-PCIe with 16 GB HBM2 memory. The NVIDIA-SMI version is 418.67 and the CUDA version is 10.1.

D.3 Average Runtime

For the 5-round active learning with SeqMix augmentation, our program runs about 500 seconds for WebPage dataset, 1700 seconds for the CoNLL slicing dataset, and 3.5 hours for ACE 2005. If the QBC query policy used, all the runtime will be multiplied about 3 times.

D.4 Hyper parameter Search

For the discriminator score range, we first examine the perplexity score distribution of the CoNLL training set. Then determine an approximate score range $(0, 2000)$ first. We linearly split score ranges below 2000 to conduct parameter study and report the representative ranges in Section **??**. Given the consideration to the generation speed and the augment rate setting, we finally choose 500 as the upper limit rather than a too narrow score range setting.

For the mixing coefficient λ, we follow (**?**) to sample it from $Beta(\alpha, \alpha)$ and explore α ranging from $[0.5, 16]$. We present this parameter study in Section **??**. The result shows different α did not influence the augmentation performance much.

For the augment rate and the valid tag density, we also have introduced the parameter study in Section **??**.