# Comp550 Final Project README

## There serveral step to run our code

### Size of the Data Store

| IT | Medical | koran | Law |
|---|---|---|---|
| 3613350 | 6903320 | 524400 | 19070000 |

### Generate the Data Store

```
%env DSTORE_SIZE=6903320 # Size of the data store
%env MODEL_PATH=Model/wmt19.de-en.ffn8192.pt # path to the pretrained model
%env DATA_PATH=medical # path to the pre processed data
%env DATASTORE_PATH=data_store # path to the stored data store
%env PROJECT_PATH=adaptive-knn-mt # path to the cloned project

! CUDA_VISIBLE_DEVICES=0 python $PROJECT_PATH/save_datastore.py $DATA_PATH \
    --dataset-impl mmap \
    --task translation \
    --valid-subset train \
    --path $MODEL_PATH \
    --max-tokens 4096 --skip-invalid-size-inputs-valid-test \
    --decoder-embed-dim 1024 --dstore-fp16 --dstore-size $DSTORE_SIZE --dstore-mmap
$DATASTORE_PATH

# 4096 and 1024 depend on your device and model separately
```

### Generate Faiss Index

```
%env PROJECT_PATH=adaptive-knn-mt # path to the project path
%env DSTORE_PATH=data_store # path to the stored data store
%env DSTORE_SIZE=6903320 # size of the data store

! CUDA_VISIBLE_DEVICES=0 python $PROJECT_PATH/train_datastore_gpu.py \
    --dstore_mmap $DSTORE_PATH \
    --dstore_size $DSTORE_SIZE \
    --dstore-fp16 \
    --faiss_index ${DSTORE_PATH}/knn_index \
    --ncentroids 4096 \
    --probe 32 \
    --dimension 1024
```

## Adjust the shell script

```
# Similar to the above
DSTORE_SIZE=6903320
DATA_PATH=medical
PROJECT_PATH=adaptive-knn-mt
MODEL_PATH=Model/wmt19.de-en.ffn8192.pt
DATASTORE_PATH=data_store
# We trained for k=1, k=2, k=4
max_k_grid=(1 2 4)
batch_size_grid=(32 32 32)
update_freq_grid=(1 1 1)
valid_batch_size_grid=(32 32 32)

for idx in ${!max_k_grid[*]}
do
  # Path to save the trained model
  MODEL_RECORD_PATH=model_record_medical/train-hid32-maxk${max_k_grid[$idx]}
  TRAINING_RECORD_PATH=training_record_medical/tensorboard/train-hid32-
maxk${max_k_grid[$idx]}
  mkdir -p "$TRAINING_RECORD_PATH"

  CUDA_VISIBLE_DEVICES=0 python \
  $PROJECT_PATH/fairseq_cli/train.py \
  $DATA_PATH \
  --log-interval 100 --log-format simple \
  --arch transformer_wmt19_de_en_with_datastore \
  --tensorboard-logdir "$TRAINING_RECORD_PATH" \
  --save-dir "$MODEL_RECORD_PATH" --restore-file "$MODEL_PATH" \
  --reset-dataloader --reset-lr-scheduler --reset-meters --reset-optimizer \
  --validate-interval-updates 100 --save-interval-updates 100 --keep-interval-updates 1 --
max-update 5000 --validate-after-updates 1000 \
```

```
  --save-interval 10000 --validate-interval 100 \
  --keep-best-checkpoints 1 --no-epoch-checkpoints --no-last-checkpoints --no-save-
optimizer-state \
  --train-subset valid --valid-subset valid --source-lang de --target-lang en \
  --criterion label_smoothed_cross_entropy --label-smoothing 0.001 \
  --max-source-positions 1024 --max-target-positions 1024 \
  --batch-size "${batch_size_grid[$idx]}" --update-freq "${update_freq_grid[$idx]}" --
batch-size-valid "${valid_batch_size_grid[$idx]}" \
  --task translation \
  --optimizer adam --adam-betas "(0.9, 0.98)" --adam-eps 1e-08 --min-lr 3e-05 --lr 0.0003
--clip-norm 1.0 \
  --lr-scheduler reduce_lr_on_plateau --lr-patience 5 --lr-shrink 0.5 \
  --patience 30 --max-epoch 50 \
  --load-knn-datastore --dstore-filename $DATASTORE_PATH --use-knn-datastore \
  --dstore-fp16 --dstore-size $DSTORE_SIZE --probe 32 \
  --knn-sim-func do_not_recomp_l2 \
  --use-gpu-to-search --move-dstore-to-mem --no-load-keys \
  --knn-lambda-type trainable --knn-temperature-type fix --knn-temperature-value 10 --
only-train-knn-parameter \
  --knn-k-type trainable --k-lambda-net-hid-size 32 --k-lambda-net-dropout-rate 0.0 --max-
k "${max_k_grid[$idx]}" --k "${max_k_grid[$idx]}" \
  --label-count-as-feature
done
```

## Train the model

```
! bash train.sh
```

## Inference for Adaptive model

```
%env DSTORE_SIZE=6903320 # size of the data store
%env MODEL_PATH=model_record_medical/train-hid32-maxk1/checkpoint.best_loss_1.53.pt # path
to the saved model

%env DATASTORE_PATH=data_store
%env DATA_PATH=medical
%env PROJECT_PATH=adaptive-knn-mt

%env OUTPUT_PATH=output_medical

! mkdir -p "$OUTPUT_PATH"

! CUDA_VISIBLE_DEVICES=0 python $PROJECT_PATH/experimental_generate.py $DATA_PATH \
--gen-subset test --path $MODEL_PATH \
```

```
--arch transformer_wmt19_de_en_with_datastore \
--beam 4 --lenpen 0.6 --max-len-a 1.2 --max-len-b 10 --source-lang de --target-lang en \
--scoring sacrebleu \
--batch-size 32 \
--tokenizer moses --remove-bpe \
--model-overrides "{'load_knn_datastore': True, 'use_knn_datastore': True, \
'dstore_filename': '$DATASTORE_PATH', 'dstore_size': $DSTORE_SIZE, \
'dstore_fp16': True, 'probe': 32,'knn_sim_func': 'do_not_recomp_l2', \
'use_gpu_to_search': True, 'move_dstore_to_mem': True, 'no_load_keys': True, \
'knn_temperature_type': 'fix', 'knn_temperature_value': 10,}" \
| tee "$OUTPUT_PATH"/generate.txt

!grep ^S "$OUTPUT_PATH"/generate.txt | cut -f2- > "$OUTPUT_PATH"/src
!grep ^T "$OUTPUT_PATH"/generate.txt | cut -f2- > "$OUTPUT_PATH"/ref
!grep ^H "$OUTPUT_PATH"/generate.txt | cut -f3- > "$OUTPUT_PATH"/hyp
!grep ^D "$OUTPUT_PATH"/generate.txt | cut -f3- > "$OUTPUT_PATH"/hyp.detok
```

## Vanilla

```
%env DSTORE_SIZE=6903320 # size of the data store
%env MODEL_PATH=Model/wmt19.de-en.ffn8192.pt # path to pretrained model

%env DATASTORE_PATH=data_store
%env DATA_PATH=medical
%env PROJECT_PATH=adaptive-knn-mt

%env OUTPUT_PATH=output_medical

!mkdir -p "$OUTPUT_PATH"

!CUDA_VISIBLE_DEVICES=0 python $PROJECT_PATH/experimental_generate.py $DATA_PATH\
    --gen-subset test\
    --path $MODEL_PATH --arch transformer_wmt19_de_en_with_datastore\
    --beam 4 --lenpen 0.6 --max-len-a 1.2 --max-len-b 10 --source-lang de --target-lang
en\
    --scoring sacrebleu\
    --batch-size 32\
    --tokenizer moses --remove-bpe\
    # Need to modify the value of k for a specific inference
    --model-overrides "{'load_knn_datastore': True, 'use_knn_datastore':
True,'dstore_filename': '$DATASTORE_PATH', 'dstore_size': $DSTORE_SIZE, 'dstore_fp16':
True, 'k': 1, 'probe': 32,'knn_sim_func': 'do_not_recomp_l2', 'use_gpu_to_search': True,
'move_dstore_to_mem': True, 'no_load_keys': True,'knn_lambda_type': 'fix',
'knn_lambda_value': 0.7, 'knn_temperature_type': 'fix', 'knn_temperature_value': 10,}" \
    | tee "$OUTPUT_PATH"/generate.txt
```

```
!grep ^S "$OUTPUT_PATH"/generate.txt | cut -f2- > "$OUTPUT_PATH"/src
!grep ^T "$OUTPUT_PATH"/generate.txt | cut -f2- > "$OUTPUT_PATH"/ref
!grep ^H "$OUTPUT_PATH"/generate.txt | cut -f3- > "$OUTPUT_PATH"/hyp
!grep ^D "$OUTPUT_PATH"/generate.txt | cut -f3- > "$OUTPUT_PATH"/hyp.detok
```

## Inference Based on the base model

```
MODEL_PATH=/path/to/pretrained_model_path/
DATA_PATH=/path/to/fairseq_preprocessed_path/
DATASTORE_PATH=/path/to/saved_datastore/
PROJECT_PATH=/path/to/knnmt/

mkdir -p $OUTPUT_PATH

CUDA_VISIBLE_DEVICES=0 python $PROJECT_PATH/fairseq_cli/generate.py $DATA_PATH\
    --gen-subset test \
    --path $MODEL_PATH \
    --beam 4 --lenpen 0.6 --max-len-a 1.2 --max-len-b 10 --source-lang de --target-lang en
\
    --scoring sacrebleu \
    --max-tokens 4096 \
    --tokenizer moses --remove-bpe | tee $OUTPUT_PATH/generate.txt

grep ^S "$OUTPUT_PATH"/generate.txt | cut -f2- > "$OUTPUT_PATH"/src
grep ^T "$OUTPUT_PATH"/generate.txt | cut -f2- > "$OUTPUT_PATH"/ref
grep ^H "$OUTPUT_PATH"/generate.txt | cut -f3- > "$OUTPUT_PATH"/hyp
grep ^D "$OUTPUT_PATH"/generate.txt | cut -f3- > "$OUTPUT_PATH"/hyp.detok
```

To reproduce the whole project, you only need to rerun all of the above shell scripts in order, and changed the parameters to the value you want to test.