**Documentation for running the baseline systems – Group 04**

Run the scripts for each model by downloading the data files from https://github.com/mbzuai-nlp/SemEval2024-task8#data\_format and running the models as shown below.

Important is to set two environment variables, exp\_name and seed\_value.

exp\_name is to set a different name to test/model you are running to compare later.

seed\_value is to set a static seed\_value to re-create random scenarios.

**Task A monolingual:**

python3 baseline\_task\_A.py --train\_file\_path subtaskA\_train\_monolingual.jsonl --test\_file\_path subtaskA\_dev\_monolingual.jsonl --prediction\_file\_path subtaskA\_distilbert\_predictions\_monolingual.jsonl --subtask A --model 'distilbert-base-uncased'

**Task A multilingual:**

python3 baseline\_task\_A.py --train\_file\_path subtaskA\_train\_multilingual.jsonl --test\_file\_path subtaskA\_dev\_multilingual.jsonl --prediction\_file\_path subtaskA\_distilbert\_predictions\_multilingual.jsonl --subtask A --model 'distilbert-base-multilingual-cased'

**Task B:**

python3 subtaskB/baseline/transformer\_baseline.py --train\_file\_path subtaskB\_train.jsonl --test\_file\_path subtaskB\_dev.jsonl --prediction\_file\_path subtaskB\_distilbert\_predictions\_monolingual.jsonl --subtask B --model 'distilbert-base-uncased'

**Task C:**

python transformer\_baseline.py --model\_path "severinsimmler/xlm-roberta-longformer-base-16384" --train\_file "../data/subtaskC\_train.jsonl" --load\_best\_model\_at\_end True --dev\_file "../data/subtaskC\_dev.jsonl" --test\_files ../data/subtaskC\_dev.jsonl --metric\_for\_best\_model "eval\_mean\_absolute\_diff" --greater\_is\_better False --do\_train True --do\_predict True --seed $seed\_value --output\_dir "./runs/$exp\_name" --logging\_dir "./runs/$exp\_name/logs" --num\_train\_epochs 10 --per\_device\_train\_batch\_size 32 --per\_device\_eval\_batch\_size 32 --auto\_find\_batch\_size True --logging\_steps 10 --load\_best\_model\_at\_end True --evaluation\_strategy "epoch" --save\_strategy "epoch" --save\_total\_limit 2

This results in a predictions file as specified under the ‘--prediction\_file\_path’ argument. This prediction file can then be evaluated using scorer.py. This python file also uses format\_checker.py so make sure you also have that file installed in the same directory.

**Task A monolingual:**

python3 subtaskA/scorer/scorer.py --gold\_file\_path=subtaskA\_dev\_monolingual.jsonl --pred\_file\_path=subtaskA\_distilbert\_predictions\_monolingual.jsonl

**Task A multilingual**

python3 subtaskA/scorer/scorer.py --gold\_file\_path=subtaskA\_dev\_multilingual.jsonl --pred\_file\_path=subtaskA\_distilbert\_predictions\_multilingual.jsonl

**Task B:**

python3 subtaskB/scorer/scorer.py --gold\_file\_path=subtaskB\_dev.jsonl --pred\_file\_path=subtaskB\_distilbert\_predictions\_monolingual.jsonl

**Task C:**

python ./scorer.py --pred\_file\_path ./runs/exp\_1/predictions/subtaskC\_dev.jsonl --gold\_file\_path ../data/subtaskC\_dev.jsonl