Evaluation on the Test Set and Comparison to Baseline

In [ ]:
```
logging.set_verbosity_error()

tat_summarizer = pipeline('summarization',
                          model=new_model,
                          tokenizer=new_tokenizer)

#*p*#tat_sample_dialog_list = []
prediction_tat_list = []
reference_tat_list = []

tat_tic = timeit.default_timer()

for sample_num in range(len(dataset['test'])):
    this_sample = dataset['test'][sample_num]

    if do_have_lotta_output_from_all_dialogs_summaries:
        print("="*75)
        print(f"dialogue: \n{this_sample['dialogue']}\n---------------")
    ##endof:  if do_have_lotta_output_from_all_dialogs_summaries

    ground_tat_summary = this_sample['summary']
    res_tat = summarizer(this_sample['dialogue'])
    res_tat_summary = res_tat[0]['summary_text']

    if do_have_lotta_output_from_all_dialogs_summaries:
        print("-"*70)
        print(f"human-genratd summary:\n{ground_tat_summary}")
        print("-"*70)
        print( "dwb-flan-t5-small-lora-finetune summary:" + \
              f"\n{res_tat_summary}")
        print("-"*70)
    ##endof:  if do_have_lotta_output_from_all_dialogs_summaries

    #*p*#    tat_sample_dialog_list.append(this_sample)
    reference_tat_list.append(ground_tat_summary)
    prediction_tat_list.append(res_tat_summary)
##endof:  for sample_num in range(len(dataset['test']))

tat_toc = timeit.default_timer()

tat_duration = tat_toc - tat_tic

print( "Getting things ready for scoring (after training)")
print(f"took {tat_toc - tat_tic:0.4f} seconds.")

tat_time_str = format_timespan(tat_duration)

print(f"which equates to {tat_time_str}")


rouge = load_metric('rouge', trust_remote_code=True)
```

```
results_tat = rouge.compute(
                  predictions=prediction_tat_list,
                  references=reference_tat_list,
                  use_aggregator=True
)

# >>> print(list(results_tat.keys()))
# ['rouge1', 'rouge2', 'rougeL', 'rougeLsum']

#*p*# objects_to_pickle.append(tat_sample_dialog_list)
#*p*# objects_to_pickle.append(prediction_tat_list)
#*p*# objects_to_pickle.append(reference_tat_list)
#*p*# objects_to_pickle.append(results_tat)
```

In [65]:
```
print_rouge_scores(results_tat, "TEST AFTER TRAINING")
```

```
---------- ROUGE SCORES ----------
  --------- TEST AFTER TRAINING ----------
ROUGE-1 results
AggregateScore(
     low=Score(
          precision=0.18465960106995058,
          recall=0.5289884514472354,
          fmeasure=0.25686215590159345),
     mid=Score(
          precision=0.19191206582001252,
          recall=0.5419927875442789,
          fmeasure=0.26514311109911903),
     high=Score(
          precision=0.19892074709381968,
          recall=0.5560562002147722,
          fmeasure=0.273181138437335)
)
ROUGE-2 results
AggregateScore(
     low=Score(
          precision=0.05269906298279127,
          recall=0.15575094190620362,
          fmeasure=0.07409348910518994),
     mid=Score(
          precision=0.05716364568273007,
          recall=0.16594455254812504,
          fmeasure=0.0797410330836727),
     high=Score(
          precision=0.06147635605696184,
          recall=0.1761782280013389,
          fmeasure=0.08508543913464939)
)
ROUGE-L results
AggregateScore(
     low=Score(
          precision=0.1358391419777274,
          recall=0.38856823315589245,
          fmeasure=0.18868402958397204),
     mid=Score(
          precision=0.1413916125834373,
          recall=0.39950997023341217,
          fmeasure=0.19515605909360623),
     high=Score(
          precision=0.14730026614718988,
          recall=0.4117337256634965,
          fmeasure=0.20223694130284198)
)
ROUGE-Lsum results
AggregateScore(
     low=Score(
          precision=0.13581244201168188,
          recall=0.38871271829792664,
          fmeasure=0.1886832040731757),
     mid=Score(
          precision=0.14133285520379574,
```

```
                    recall=0.3998294223955289,
                    fmeasure=0.1950362847385175),
               high=Score(
                    precision=0.147008914964092,
                    recall=0.4109702578189206,
                    fmeasure=0.20206485129716942)
          )
```