Actual Baseline

In [40]:
```
#  ref1 = "https://web.archive.org/web/20240530051418/" + \
#+        "https://stackoverflow.com/questions/73221277/" + \
#+        "python-hugging-face-warning"
#  ref2 = "https://web.archive.org/web/20240530051559/" + \
#+        "https://huggingface.co/docs/transformers/en/" + \
#+        "main_classes/logging"

##  Haven't tried this, because the logging seemed easier,
##+ and the logging worked
#os.environ("TRANSFORMERS_NO_ADVISORY_WARNINGS") = 1

logging.set_verbosity_error()

summarizer = pipeline('summarization',
                      model=model,
                      tokenizer=tokenizer)

#*p*#baseline_sample_dialog_list = []
baseline_prediction_list = []
baseline_reference_list = []

baseline_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_1:
        print(f"dialogue: \n{this_sample['dialogue']}\n---------------")
    ##endof:  if do_have_lotta_output_from_all_dialogs_summaries_1

    ground_summary = this_sample['summary']
    res = summarizer(this_sample['dialogue'])
    res_summary = res[0]['summary_text']

    if do_have_lotta_output_from_all_dialogs_summaries_1:
        print(f"human-genratd summary:\n{ground_summary}")
        print(f"flan-t5-small summary:\n{res_summary}")
    ##endof:  if do_have_lotta_output_from_all_dialogs_summaries_1

    #*p*#    baseline_sample_dialog_list.append(this_sample)
    baseline_reference_list.append(ground_summary)
    baseline_prediction_list.append(res_summary)
##endof:  for sample_num in range(len(dataset['test']))

baseline_toc = timeit.default_timer()

baseline_duration = baseline_toc - baseline_tic

print( "Getting things ready for scoring")
print(f"took {baseline_toc - baseline_tic:0.4f} seconds.")

#  It turns out that the deprecated one is preferable in
```

```
#+ output, at least until I can debug the aggregation of
#+ scores with another version
#+ That should come with the

rouge = load_metric('rouge', trust_remote_code=False)

baseline_results = rouge.compute(
                      predictions=baseline_prediction_list,
                      references=baseline_reference_list,
                      use_aggregator=True
)

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

#*p*# objects_to_pickle.append(baseline_sample_dialog_list)
#*p*# objects_to_pickle.append(baseline_prediction_list)
#*p*# objects_to_pickle.append(baseline_reference_list)
#*p*# objects_to_pickle.append(baseline_results)
```

```
Getting things ready for scoring
took 1113.8523 seconds.
```

In [43]:
```
print_rouge_scores(baseline_results)
```

```
---------- ROUGE SCORES ----------
  --------- dialogue ----------
ROUGE-1 results
AggregateScore(
     low=Score(
          precision=0.36320630445704477,
          recall=0.5391471908229872,
          fmeasure=0.41209971865595346),
     mid=Score(
          precision=0.37394711195774655,
          recall=0.5518956018541074,
          fmeasure=0.4216852406490635),
     high=Score(
          precision=0.3843089278286546,
          recall=0.5652673531194096,
          fmeasure=0.43106509690207256)
)
ROUGE-2 results
AggregateScore(
     low=Score(
          precision=0.15921598436893325,
          recall=0.24399260896723063,
          fmeasure=0.18098064580068599),
     mid=Score(
          precision=0.16751331807822,
          recall=0.25688418792453044,
          fmeasure=0.1901013569791662),
     high=Score(
          precision=0.17601669526453642,
          recall=0.26996925142296735,
          fmeasure=0.1988747178644448)
)
ROUGE-L results
AggregateScore(
     low=Score(
          precision=0.2798170544171966,
          recall=0.4220715282711129,
          fmeasure=0.31929134202126586),
     mid=Score(
          precision=0.28896822314514115,
          recall=0.43511544077895614,
          fmeasure=0.32786822093032963),
     high=Score(
          precision=0.29854357582265284,
          recall=0.44899752808600696,
          fmeasure=0.33655474992458917)
)
ROUGE-Lsum results
AggregateScore(
     low=Score(
          precision=0.2803262832798807,
          recall=0.4225291787351153,
          fmeasure=0.31968927668471403),
     mid=Score(
          precision=0.28924184457875435,
```

```
                    recall=0.4348222878968877,
                    fmeasure=0.3278854406001706),
               high=Score(
                    precision=0.2986060650799353,
                    recall=0.4471497194451444,
                    fmeasure=0.3366741267731763)
          )
```