Boosting Query Based Summarization by Exploiting Query Relations

Group 5: Hailiang Dong, Yangxiao Lu, Priyanshi Shah, Adit Shah

Introduction

Query-Based Summarization

- Aims to extract or generate a summary of a document which directly answers or is relevant to the search query. Ignore irrelevant contents in the input.
- Input consists of text source (T) and query (Q), the objective is to generate the summary/answer (S).

Existing Approach

- Combine and Encode the two inputs Q and T using separation token to form a single text input.
- Then any Seq2Seq model can be used to solve the task.

<s> Q </s> T </s>

Meeting Summarization Task

- An special case of query-based summarization. Each meeting is associate with **multiple** queries.
- Two type of queries involved:
  1. General Query

     A query that is relevant to all parts of meeting, the input T is all the meeting transcripts.

     E.g., Summarize the whole meeting.

  2. Specific Query

     A query that is about certain topic in the meeting, the input T is usually subset of meeting transcripts.

     E.g., What did the department header think about the petitions ?

Can we boost the summarization performance by exploiting related query and its summary?

Method

Idea

- Add two extra inputs related query (Qr) and its summary/answer (Qs), combine and encode all inputs as follows.

<s> Qr Qs </s> Q </s> T </s>

Challenges

- How to determine which query Qr is most related to current query Q ?
- During validation/testing time, Qs is not provided and must be generated by the model on the fly.
  1. Need to guarantee there is no future reference !

     If Qj is the most related query for Qi, then Qj must be evaluated (generate the answer of Qj) before Qi.

  2. What is the best evaluation order such that later queries can be benefited the most ?

Method (Continued)

How to determine which query Qr is most related to current query Q ?

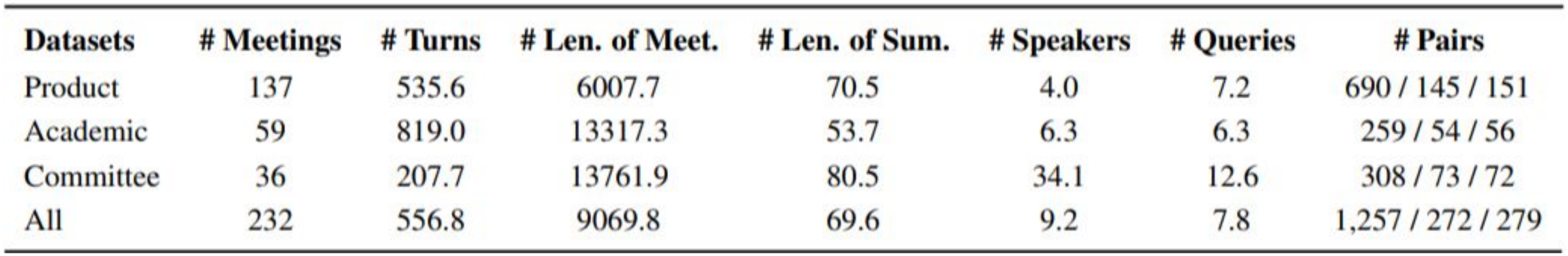
- Use BERT model to compute the embeddings for each query Qi (last 4 hidden states, averaged over tokens) .
- Employ cosine similarity to evaluate the relevance of Qj to Qi, and select the one with maximum similarity.

How to determine evaluation order and related query in testing/validation time.

- Evaluate the importance Wi of current query Qi as $\sum_{j \neq i} sim(Q_i, Q_j)$
- Queries with larger importance will be evaluated first.
- Determine related query by computing the similarity over only previous evaluated queries (Enforce there is no future reference).

Dataset

QMSum Dataset

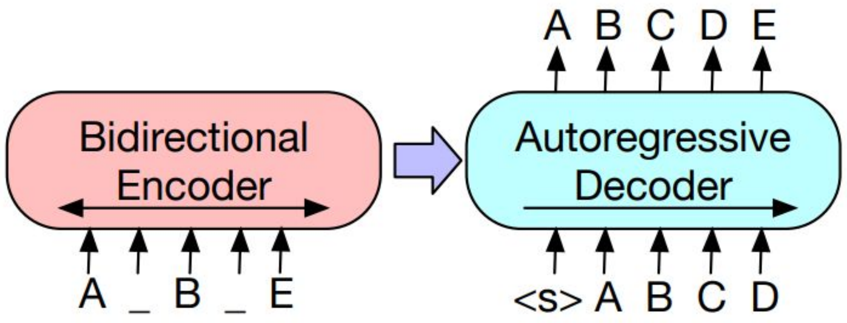


Datasets	# Meetings	# Turns	# Len. of Meet.	# Len. of Sum.	# Speakers	# Queries	# Pairs
Product	137	535.6	6007.7	70.5	4.0	7.2	690 / 145 / 151
Academic	59	819.0	13317.3	53.7	6.3	6.3	259 / 54 / 56
Committee	36	207.7	13761.9	80.5	34.1	12.6	308 / 73 / 72
All	232	556.8	9069.8	69.6	9.2	7.8	1,257 / 272 / 279

- 232 meetings in 3 different domains in total.
- Each meeting is associate with two sets of general and specific queries.
- For specific query, the related transcripts span is provided (set of intervals).

Model

BART Architecture



- A pre-trained model for sequence-to-sequence generation tasks (translation, summarization, etc).
- Supports up to 1024 input tokens at most.
- Pre-trained on CNN-DM dataset.

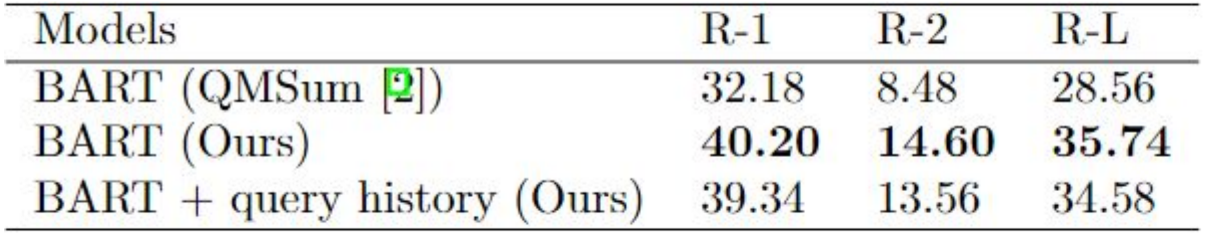
Modifications

- We extended the positional embedding layer to **support up to 2048 tokens**.
- The weights of position 1024 to 2047 (0 indexed) are initialized from position 0 to 1023.

Results

Training Details

- **80** epochs, with batch size **2**, learning rate **1e-5**, weight decay **1e-4**



Models	R-1	R-2	R-L
BART (QMSum [2])	32.18	8.48	28.56
BART (Ours)	**40.20**	**14.60**	**35.74**
BART + query history (Ours)	39.34	13.56	34.58

- The BART model we trained (2nd) use same input achieves significant better results compared to results reported in the original paper (1st) (both used 2048 maximum tokens).
- With extra related query and its summary/answer (3rd), the performance is slightly decreased (compared to 2nd) !

Analysis of Performance Decrease

- **General queries use all meeting transcripts as input, as we added extra input, it is easier to exceed token limit.**

  Solution: remove general queries from training, testing set (only lost like 15% of data).

- **The model might be overfitting.**

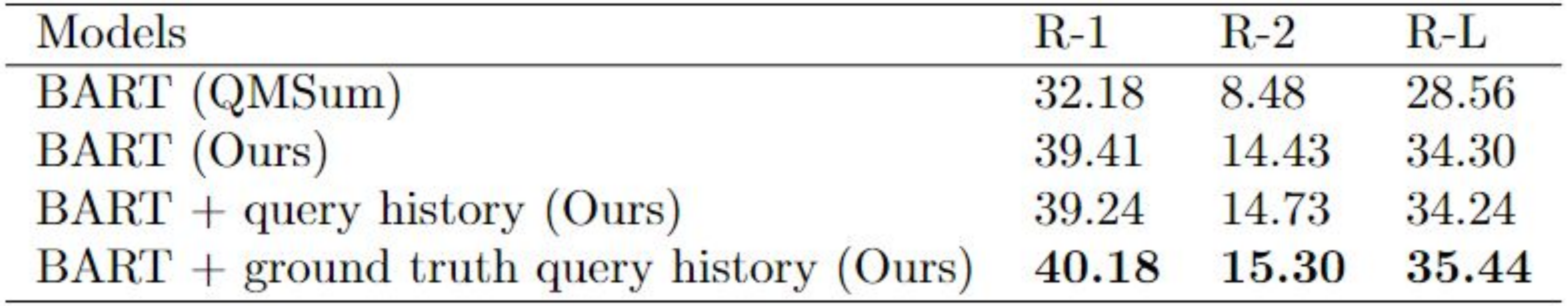
  Solution: increase weight decay (5e-3), reduce the number of training epochs (40 epoch).

- **During test time, the summary for related query is generated by model itself. Poor summarization/answer might hurt the model performance.**

  Solution: cannot be solved, at testing time, there is no ground truth summary for any meeting queries at beginning. However, in our case, we can at least test if this is the root cause.

Results (general query removed)

- With extra related query and its summary/answer (3rd), can achieve similar performance to BART with standard input (2nd).
- If ground truth summaries is provided for related query, can achieve a significant performance boost.



Models	R-1	R-2	R-L
BART (QMSum)	32.18	8.48	28.56
BART (Ours)	39.41	14.43	34.30
BART + query history (Ours)	39.24	14.73	34.24
BART + ground truth query history (Ours)	**40.18**	**15.30**	**35.44**

Related query & summary can significant boost the query-based summarization performance if we have high quality query histories.

Questions ?