Course: ADVANCE NLP

QUERY-GUIDED MULTI-PERSPECTIVE ANSWER SUMMARIZATION

November 2, 2023

Abhinav Anand (2020101054)
Vivek Mathur (2020113002)
Kunal Kamalkishore Bhosikar (2022121005)

1 Introduction

Text summarization is the technique for transforming long documents into shortened versions, while focusing on the sections that convey useful information, and without losing the overall meaning. Query-focused summarization is a subtask within text text summarization that aims to generate a summary of given text conditioned upon a user-query that is passed alongside the source document as imput to the model. Query-focused summarization is even relevant for Community Question Answering, where a person poses a question and can get multiple answers to sift through. Work in this field has a notion of 'best answer' and make use of this best answer as the gold summary of all other answers[2]. However, best answer only presents one perspective and rarely captures the variety of perspective of other answers. Ideally, an answer summary should cover multiple perspectives found in answers. Answer summarization is a form of query-based, multi-document summarization, and creating answer summaries that reflect the underlying varying perspectives entails several subtasks:

- Selection of answer sentences relevant to the question (query sentence relevance
- Grouping these sentences based on perspectives (clustering)
- Summarizing each perspective (cluster summarization)
- Producing an overall fused summary (fusion)

2 Problem Statement

In this project, We focus on generating a fluent and concise answer summary that includes all perspectives of all the answers on a community question-answering forum. To break down the problem, query-guided refers to using the question as a guide for summarization, while multiperspective refers to using the perspective from each of the multiple available answers to produce a single summarized answer.

3 Dataset

AnswerSumm :[3] is a English-language dataset of questions and answers collected from a StackExchange data dump.The dataset was created to support the task of query-focused answer summarization with an emphasis on multi-perspective answers. The dataset consists of 4631 such question-answer threads annotated by professional linguists and includes over 8700 summaries. For each thread, the annotator writes two summaries. In First Summary, the annotator is asked to mark sentences that are included in the final summary and instructed to more closely use the words in these sentences rather than abstract. In Second

Question: I recently relocated to USA and have no Credit Score. Is Secure Credit Card is the only option for me to start building my credit score? Also please recommend which other credit cards are available for people like me to build credit score
Answer 1: If you have an AMEX from another country, you can get an AMEX in the US. American Express has a separate system that is not as strongly country-dependent as, say, VISA and MasterCard...
Answer 2: Secured credit cards are usually not very cost effective for building credit. Find a local credit union, of medium to large size. A credit union is like a bank, but operates under slightly different rules, and is non-profit...
Answer 3: If you have had an American Express card abroad, you can try and get a US Amex...
Answer 4: If the country you came from has an HSBC, you can ask HSBC to use your credit rating from that country to give you an HSBC Mastercard in the US...
Summary:
There are a range of options available to you, although your chance of success will depend on the bank that you apply with. However, if you have previously had a card with HSBC or American Express, the process may be simpler. Other options could include borowing from a credit union or asking a friend or family member to be an additional cardholder with you.

Figure 1: An example of the Query-guided multiple-perspective answer summarization. The goal is to generate a single summarized answer using multiple answers.

Summary the annotator was asked to paraphrase and condense the cluster summaries but was not asked to reduce abstraction.

4 Methodology and Evaluation:

4.1 BaseLine:

For baseline we used 2 model. **BART**[1] BART is a denoising autoencoder for pretraining sequence-to-sequence models. It is trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text. It uses a standard Transformer-based neural machine translation architecture. It uses a standard seq2seq/NMT architecture with a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT). **T5** or Text-to-Text Transfer Transformer, is a Transformer based architecture that uses a text-to-text approach. Every task – including translation, question answering, and classification – is cast as feeding the model text as input and training it to generate some target text. This allows for the use of the same model, loss function, hyperparameters, etc. across our diverse set of tasks.

4.2 Two Step Appoaches:

Two-step approaches consist of an Extract model, which ranks answers relevant to the input query, and an abstractor model, which synthesizes the ranked answers into a final summary.

Extractor: model scores each answer for relevance to the query and then rank answers which are then concatenated and passed to the abstractor. We implemented three Ranker models.

- **BM25:** is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document.
- **Cross-encode:** Single encoder concatenates query and source passage as input to produce the similarity score.
- **bi-encoder:** Encodes the query and source passage separately. Cosine similarity gives the relevance score. It provides computational benefits as passage embeddings can be precomputed and stored , reused for multiple queries. Fig2 depicts the difference between the two encoders.

Abstractor: We use BART and T5 model as abstractor to generate answer summary given top-k selected answers from extractor.

4.3 Evaluation:

We will automatically evaluate our models on ROUGE metrics.

- **ROUGE-N:** measures the number of matching 'n-grams' between our model-generated text and a 'reference'.

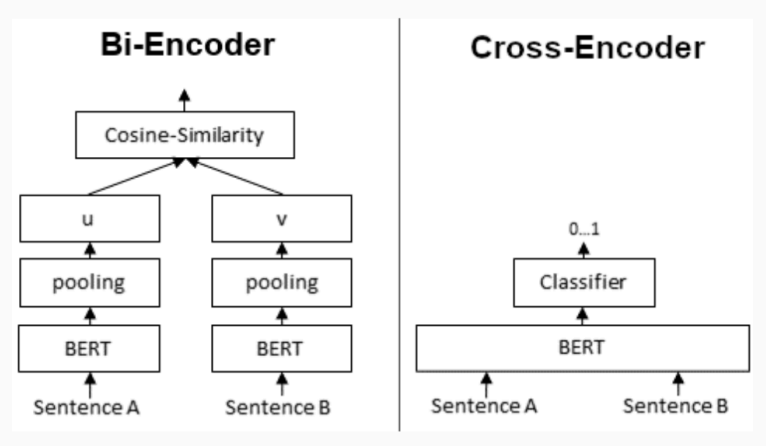

Figure 2: Bi-Encoder and Cross-Encoder

- **ROUGE-L:** measures the longest common subsequence (LCS) between our model output and reference. All this means is that we count the longest sequence of tokens that is shared between both.

5 Experiment

For evaluations on baseline model we have used the following approaches:

- **all-ans:** Concatenated all the answers of a query to form a source and the First Summary is used as target.
- **q+all-ans:**: Query concatenated with all the answers to form source and the First Summary is used as target.
- **clu-summ:** Concatenated all the cluster summaries of a query to form source and the Second Summary is used as target.
- **q+clu-summ:** Query concatenated with all the cluster summaries to form source and the Second Summary is used as target.
- **q+all-ans+cluster-id:** Question is concatinated with all the answers which are inclosed within there gold cluster-id to form source (Q + <id1>ans1<id1> + <id2>ans2<id2> +...) and the First Summary is used as a target. **Note:** This setup can't be used in practical scenarios as we don't have gold cluster-id for test data.

We limit the input/encoder sequence to 512 token and output/decoder sequence to 100 tokens and truncated the rest. We train the model for 5 epochs and used a batch size

of 2. We use Huggingface library for our experiment. We use distil-bart[4] pre-trained on Extreme Summarization (XSum) dataset as our Bart model and used 'simple' version of T5 . We first evaluate the performance on Pre-trained models and then fine-tune the models on AnswerSumm training dataset (Fine-tuned) and evaluate again.

6 Results:

In this section, we report all the experiment results based ROUGE Scores.

6.1 Baseline:

As evident from Table1 and Table2 BART trained on QSumm dataset outperforms T5 in all the experiments.

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Pre-trained (all-ans)	25.72	8.51	19.24
Pre-trained (q+all-ans)	23.26	6.04	17.64
Pre-trained (clu-summ)	59.84	46.23	52.75
Pre-trained (q+clu-summ)	57.2	45.3	50.4
Pre-trained(q+all-ans+cluster-id)	27.4	14.7	23.2
Fine-tuned (all-ans)	34.44	14.32	25.84
Fine-tuned (q+all-ans)	32.56	12.52	24.04
Fine-tuned (clu-summ)	64.35	50.98	57.42
Fine-tuned (q+clu-summ)	63.34	49.57	56.43
Fine-tuned(q+all-ans+cluster-id)	39.28	21.7	32.9

Table 1: Performance of pre-trained and fine-tuned DistilBART model on different approaches.

6.2 Two Step Approach:

Table3: reports the ROUGE score evaluated on BM25 ranked answers dataset.
Table4: reports the ROUGE score evaluated on Bi-Encoder ranked answers dataset.
Table5: reports the ROUGE score evaluated on Cross-encoder ranked answers dataset.

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Pre-trained (all-ans)	20.83	7.6	16.82
Pre-trained (q+all-ans)	15.25	3.88	12.18
Pre-trained (clu-summ)	44.74	33.13	41.05
Pre-trained (q+clu-summ)	22.42	9.57	18.50
Pre-trained(q+all-ans+cluster-id)	21.4	8.1	17.2
Fine-tuned (all-ans)	23.11	9.6	19.09
Fine-tuned (q+all-ans)	21.14	7.71	16.93
Fine-tuned (clu-summ)	46.51	34.80	42.31
Fine-tuned (q+clu-summ)	44.88	33.40	40.74
Fine-tuned(q+all-ans+cluster-id)	29.3	11.4	23.6

Table 2: Performance of pre-trained and fine-tuned T-5 model on different approaches.

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Pre-train-T5 (ranked-ans)	16.32	4.2	12.9
Pre-train-Bart(ranked-ans)	29.91	11.17	22.02
Pre-train-T5 (q+ranked-ans)	20.84	7.01	16.78
Pre-train-Bart (q+ranked-ans)	30.41	12.88	24.45
Fine-tune-T5 (ranked-ans)	20.83	7.6	16.82
Fine-tune-Bart(ranked-ans)	31.27	11.85	22.8
Fine-tune-T5 (q+ranked-ans)	21.94	7.58	17.63
Fine-tune-Bart (q+ranked-ans)	32.1	12.6	22.93

Table 3: Performance of T-5 model and Bart on BM-25 ranked answers.

7 Analysis and Observations:

7.1 Baseline:

BART trained on Qsumm dataset outperforms T5 model in all the experiments, especially when using cluster summaries by a huge margin. This shows that BART can transfer what it learned in pre-training to AnserSumm data as the pre-training task is similar to cluster summarization. We observe when we use questions and all answers as source performance drops on both the models. As evident from Table6, the average token length of question + all-answer is 737.5, while our source length is fixed to 512 tokens. So, the encoder is truncating the rest, resulting in a loss of important perspective for better summarization. We also noticed when we enclosed all answers within their cluster-id (q+all-ans+cluster-id), the performance increased by a huge margin. This is because answers with cluster-id [-1] don't belong to any cluster, and thus, these answer sentences are not used for writing

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Pre-train-T5 (ranked-ans)	6.78	0.22	5.89
Pre-train-Bart(ranked-ans)	10.06	0.6	8.9
Pre-train-T5 (q+ranked-ans)	7.16	0.26	6.19
Pre-train-Bart (q+ranked-ans)	10.26	0.59	12.59
Fine-tune-T5 (ranked-ans)	10.75	0.85	8.88
Fine-tune-Bart(ranked-ans)	13.48	1.13	10.17
Fine-tune-T5 (q+ranked-ans)	10.66	0.848	8.81
Fine-tune-Bart (q+ranked-ans)	14.12	1.06	12.59

Table 4: Performance of T-5 model and Bart on bi-encoder ranked answers.

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Pre-train-T5 (ranked-ans)	10.76	0.87	8.96
Pre-train-Bart(ranked-ans)	9.82	0.54	8.09
Pre-train-T5 (q+ranked-ans)	6.73	0.235	5.88
Pre-train-Bart (q+ranked-ans)	11.43	0.52	11.48
Fine-tune-T5 (ranked-ans)	10.82	0.87	8.94
Fine-tune-Bart(ranked-ans)	14.31	0.52	10.01
Fine-tune-T5 (q+ranked-ans)	10.59	0.874	8.84
Fine-tune-Bart (q+ranked-ans)	15.01	.71	10.14

Table 5: Performance of T-5 model and Bart on Cross-encoder ranked answers.

Text	Average length
Question	52.54
All-answers	685
First-summary	48
Second-summary	41
Cluster-summaries	50.11
Question+All-answers	737.5

Table 6: Average length of tokens.

summaries. The model learns to ignore these answers (enclosed within <-1> cluster-id) and only focuses on relevant answer sentences for summarization.

7.2 Two Step Approach:

We find that the performance of both models decreases dramatically on all three datasets when we use the top-20 ranked answers for our experiment. The AnswerSumm summaries are written to capture the entire perspective. The rankings are created based on the similarities between question and answers. The top-ranked answers are similar to each other, and these similar answers generally convey the same perspective. Ranking the answers and selecting the answers with the highest N-rank does not capture all perspectives, so there is a mismatch between source and target. As a result, performance drops dramatically. We also note that performance on both encoder rankings is quite low compared to BM25 rankings. We manually analyzed some examples of ranked answers from all rankers and found that the top 20 answers ranked by the encoder models come from very few clusters/perspectives. The BM25 ranked answers comes from more diverse clusters and thus capture more perspectives.

8 Conclusion:

In this work, we have tried to address the problem of query guided multi-perspective summarization. We used BART and T5 model as our summarizer. We observe that Bart pre-trained on summarization task outperforms T5 model especially when the task is similar to pre-traing task. We also observe that simply taking the top ranked answers as source doesn't helps in multi-perspective answer summarization as the top ranked answers fails to capture all the perspective.

References

[1] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

[2] Alexander R Fabbri, Xiaojian Wu, Srini Iyer, and Mona Diab. Multi-perspective abstractive answer summarization. *arXiv preprint arXiv:2104.08536*, 2021.

[3] Alexander R Fabbri, Xiaojian Wu, Srini Iyer, Haoran Li, and Mona Diab. Answersumm: A manually-curated dataset and pipeline for answer summarization. *arXiv preprint arXiv:2111.06474*, 2021.

[4] Sam Shleifer and Alexander M Rush. Pre-trained summarization distillation. *arXiv preprint arXiv:2010.13002*, 2020.