Query-guided Multi-perspective Answer Summarization

Likhith Asapu
2020114015

Rahothvarman P
2020114008

Vanshpreet S Kohli
2020114014

Team 4, Attention Seekers
{likhith.a, rahothvarman.p, vanshpreet.k}@research.iiit.ac.in

1. Abstract

The presented document describes a report of our project task: query-guided, multi-perspective answer summarization using the AnswerSumm dataset, carried out as the term project for Advanced NLP, Monsoon 2022.

2. Introduction

Question-answering, answer-ranking and text summarization are all active research problems in the field of Natural Language Processing. While significant work has been done to improve the accuracy of all these tasks, there are relevant tasks at the confluence of all three of them that have received relatively little attention in proportion to the scope of the problems. One such task is summarization of answers to a question from a community forum (such as SO, Reddit) into one short answer containing relevant information and reflecting the range of answer perspectives. Distilling the essence of multiple answers to a question asked in any of the several popular contemporary discussion fora into a concise and factually accurate summary is an essential task given the growing ubiquity of such fora. This task requires text summarization that is guided by the given query, and captures information from more than one perspective i.e. summarizes multiple answers.

For the purposes of this project, we use the AnswerSumm dataset (Fabbri et al.) to create summaries of all the relevant answers from question threads on StackExchange. Our task, therefore, is to create a summary of said answers (for each query) that is concise, coherent, consistent, fluent and accurate. For this we leverage previous work in multi-document summarization and query-focused summarization, and implement a model that generates a summarized answer to a given input query.

3. Relevant Work

Below we enlist previously published literature we found relevant to our scope and project, which we build upon or otherwise draw inspiration from in our work.

3.1 AnswerSumm (Fabbri et al.)

AnswerSumm is a dataset and pipeline for answer summarization presented by Fabbri et al. wherein over 4000 question-answer threads from StackExchange have been manually annotated to create gold-standard summaries of all the (non-negatively voted) answers in the threads. As this is the database we shall work with for the entirety of the project, we shall discuss it in further detail in the subsequent section.

3.2 Exploring Neural Models for Query-Focused Summarization (Vig et al.)

In this paper, the authors of AnswerSumm detail state-of-the-art neural models for query-based summarization, created to work for long single documents.

3.3 ROUGE: A Package for Automatic Evaluation of Summaries (Chin-Yew Lin)

ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is the most standard set of metrics used to evaluate machine-generated summaries against reference ones. We intend

to use the ROUGE-1, ROUGE-2 and ROUGE-L (R-1, R-2, R-L) metrics, referring to the measure of overlap of the unigrams, bigrams and longest-common-subsequences between the summaries. While searching for common subsequences is non-ideal as it incentivizes copying and penalizes fluent, accurate paraphrasing, it is a standard metric and we thus use it to maintain consistency with and allow comparisons against other models.

3.4 Improve Query Focused Abstractive Summarization by Incorporating Answer Relevance (Su et al.)

In this approach, the authors bring into consideration that the task of query focused summarization can be improved by viewing the selective extraction of text for the summary as a question-answering problem. Therefore, they use a state-of-the-art Q/A model to generate answer relevance scores for each word, which are then fed into the encoder-decoder attention for a more streamlined selection of answers.

3.5 Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (Raffel et al.)

This paper introduces the popular T5 Transformer. T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format. T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task. In our case, the prefix *summarize* for summarization tasks is of significance.

3.6 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al.)

Bidirectional Encoder Representations from Transformers, or BERT, is a popular model that generates embeddings by pre-training deep bidirectional representations from unlabeled text. As described later while explaining the model, we make use of BERT embeddings for words and sentences for downstream tasks.

4. Dataset

As mentioned above, AnswerSumm was created by Fabbri et al. to support the task of query-focused answer summarization with an emphasis on multi-perspective answers. The dataset is entirely in English, and consists of over 8700 summaries for over 4200 q/a threads on StackExchange.

Dataset Structure

- question: contains metadata about the question and forum
- answers: the sentence-tokenized answers
- summaries: list of list of summaries
- annotator_id
- mismatch_info: a dictionary of any issues in processing the annotation files

An example from the AnswerSumm dataset looks as follows:


```
{
  "example_id": 9_24,
  "annotator_id": [1],
  "question": {
    "author":
     "gaming.stackexchange.com/users/11/Jeffr
     ey",
    "forum": "gaming.stackexchange.com",
    "link":
     "gaming.stackexchange.com/questions/1",
    "question": "Now that the Engineer update
     has come, there will be lots of
     Engineers building up everywhere.  How
     should this best be handled?",
    "question_tags": "\<team-fortress-2\>",
    "title": "What is a good strategy to deal
     with lots of engineers turtling on the
     other team?"
  },
  "answers": [
    {
      "answer_details": {
        "author":
         "gaming.stackexchange.com/users/44/C
         orv1nus",
        "score": 49
      }
      "sents": [
```

```
        "text": "Lots of medics with lots of
         ubers on high-damage-dealing
         classes."
        "label": [0],
        "label_summ": [0],
        "cluster_id": [[-1]],
      ]
      ...
    },
    ...
  ]
  "summaries": [
     [
      "Demomen usually work best against a
       sentry farm. Heavies or pyros can
       also be effective. Medics should be
       in the frontline to absorb the shock.
       Build a teleporter to help your team
       through.",
      "Demomen are best against a sentry
       farm. Heavies or pyros can also be
       effective. The medic should lead the
       uber combo. ..."
     ]
  ]
  "cluster_summaries":[
    "Demomen are best against a sentry
farm.",
    "Heavies or pyros can also be
effective.",
    ...
  ]
}
```


5. Models

For the given dataset, there exist no published state-of-the-art models to use as baselines. In fact, apart from the RL-based model outlined in the paper (which we shall compare our models to), AnswerSumm has witnessed no published work done using it so far, making comparisons and scoring challenging to put in context. The only relevant comparison of AnswerSumm was made in passing in Vig et al. where fine-tuning the summarization model on the data from AnswerSumm yielded slightly poorer results than on AQuaMuSe and WikiSum, but better results than on WikiHowQA and CNNDM.

We now discuss our implementation of a summarization model for the AnswerSumm dataset, along with some simple baselines that we implemented, and the rationale behind the implementations.

For a baseline model, we simply concatenate the text from all the answers, and use a pretrained T5 summarizer to generate a summary of the concatenated text. We then repeat the process with a pretrained BART summarizer to form another baseline.

For our final model, we build upon and improve one of the baselines we presented in the interim submission document for this project, wherein we consider the creation of the summaries in the dataset (which were created manually) a proxy for a model that automates the summary generation. We thus follow the same steps as the authors of the dataset followed to create the summaries, except of course that our process is fully automated. We first use BERT's extractive summarizer to create extractive summaries for all the answers, and store the list of sentences created. We then use Sentence-BERT to generate embeddings for the sentences, so that we may then cluster similar sentences together. In the aforementioned baseline, this was done using a K-means clustering algorithm. In our final model, we accomplish the clustering using heirarchial clustering instead, with the cutoff distance set to 0.65. The clusters of size smaller than (mean cluster size/2) are removed to improve relevance of the final answer. In our baseline, we used a simple T5 summarizer to generate cluster-level summaries upto a hundred words long. For this model, we fine-tune a BART model pretrained on the multi-news dataset for generation, for a summarization pipeline on our dataset using a simple data collator for sequence-to-sequence generation upto 150 words long. The cluster summaries thus formed are concatenated to form the final summary (refer to figure 1). We further experiment with our model by trying to factor in the answer relevance, measured simply as the number of upvotes received by the answer on the forum. In one of the models, we make the extractive summarizer filter 40% of the input sentences while generating cluster summaries. Note that this number is deliberately small, in an attempt not to lose possibly relevant sentences too early in the

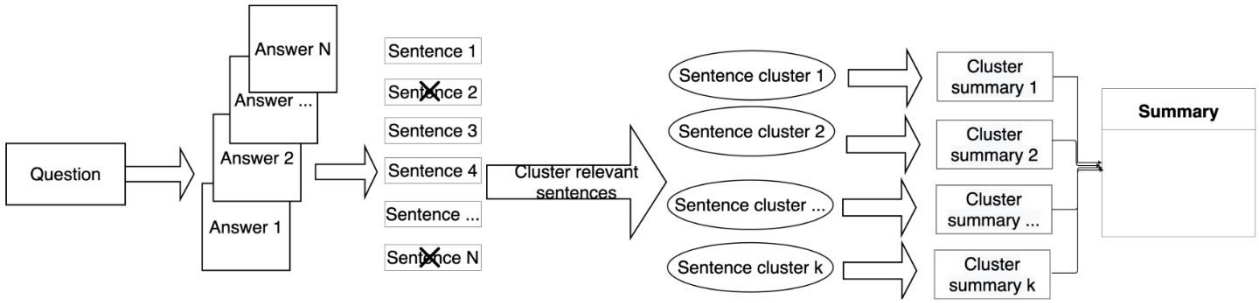

Figure 1: Implementation pipeline for Baseline 2. Given a question and answers to that question, our model selects relevant sentences, clusters them, summarizes each cluster's sentences, and fuses the clusters into a coherent summary.

pipeline, and in cognizance of the fact that the answers are themselves short. For the other model, the extractor selects sentences in a ratio proportional to the upvotes received by the answer, relative to the highest-upvoted answer. For instance, if the top answer gathered 100 upvotes, its summary would be a simple copy of the answer itself (the idea being to preserve data from the most helpful answer), and for an answer with 60 upvotes, the summary would be at most 60% as long as the answer in an attempt to remove noise.

It is worth noting here that neither of the proposed models incorporate the query into their pipelines while generating the final summary. This is not an oversight (although it considerably saves compute time as well as the time spent upon implementation), but a choice made after due deliberation. For a query-based multi-perspective summarizer, it is prima facie paradoxical to entirely ignore the query but we hypothesize that for the designated task, it may not be significantly productive (if at all) to factor in the query. This is in context of standard query-focused summarization and/or answering tasks; in such tasks it is understood that the documents scoured for the answer contain relevant as well as irrelevant pieces of information with regards to the original query, thereby requiring maintenance of attention of some kind with the query to extract only the relevant sentences for summarization. For this dataset, however, the task is different – in this case it is given to us that all the fora answers contain information relevant to the query (or at least, are phrased as though their information is relevant), and we are tasked with creating a multi-perspective summary thereof. Using the query to rank answer relevance (say, using ROUGE overlap or attention) could well be counterproductive, penalizing correct answers that are phrased in less direct ways and possibly rewarding misleading answers stated plainly and confidently. In this sense, it may even be argued that "query guided multi-perspective answer summarization" for the given dataset and the given objective is a misnomer, as this behaves differently from other query-focused models.

6. Results

We tested the accuracy of our models by comparing the summaries thus obtained against the ones supplied by the dataset by way of checking for the R-1, R-2 and R-L overlaps. We compare these models to the results from the RL-based model created and shared in the AnswerSumm paper itself. [see Table 1]

S.no.	Model/metric	R-1	R-2	R-L
1	**BART + RL (SOTA)**	28.81	8.96	24.72
2	**T5-100**	23.69	5.47	19.80
3	**BART-100**	3.52	0.78	3.07
4	**T5-256**	25.16	5.24	22.25
5	**Model-base**	16.97	2.95	15.27
6	**Model-with-upvotes**	22.01	3.89	19.58
7	**Model-upvote-agnostic**	25.29	5.89	21.11

Table 1: comparing scores of the model presented in AnswerSumm (BART + RL) with those created by us

7. Analysis

Immediately noticeable is the low score attained by the BART baseline model. At least one of the pretrained models performing poorly on this dataset was wholly expected, as not only is the dataset novel and relatively small (we tested our models on 3000 of the nearly 4000 threads in the dataset), but also human-created with multiple annotators. Unlike the summaries in datasets such as the CNN-Dailymail, the summaries in this dataset are therefore more complex to map as a function for a model to easily learn. Also significant is the introduction of possible idiosyncracies in gold-standard summaries, especially compared to general nature of a large pretrained LM. Human evaluation reveals that BART generates otherwise fluent summaries that seem to tend to be biased towards a small number of sentences, therefore not capturing enough perspectives when used on the large document that results from concatenating the answers.

T5 meanwhile receives much better scores, comparable to our best model inspite of naively creating a summary from the concatenated answers. This was also expectable, as T5-large is the second-best performing summarizer for the MultiNews dataset, behind the SOTA by only a small margin. A study of the dataset revealed that most summaries were around 100-150 words long, and very few were over 250 words long. Upon likewise increasing the output limit of T5 to 256 from 100, we get even better scores, only slightly behind our best model in R1 and R2 scores and beating it in RL score. A brief human evaluation reveals that the summaries from the T5 model seem as fluent as the ones generated by BART, but with shorter sentences and less repetition of information.

Our baseline model employing the full pipeline does moderately well, and significant gains are recorded upon improving the clustering method and using longer summaries (150 tokens instead of 100) generated by a fine-tuned BART generator. We notice here that our hypothesis about answer relevance to summary is not supported by our results, as the model that more aggressively filters out sentences from less popular answers does slightly poorly compared to the score-agnostic one. It is possible that the former loses too much of its multi-perspective nature in chasing the best answer, and likewise does worse.

The Reinforcement-Learning based method introduced in the AnswerSumm paper, meanwhile, beats our best model by around 3 rouge points for each of the evaluation metrics.

8. Summary and future work

Through this project we explore multi-perspective answer summarization in a Q/A thread using the recently introduced AnswerSumm datset.

One possible area of study that may be explored further is using conventional multi-document approaches for this dataset, which we discarded during ideation as they are aimed at much longer documents. Another approach that may be examined is treating it as a conventional query-focused summarization problem, which we also discarded for the reasons given above.

9. References

- *AnswerSumm: A Manually-Curated Dataset and Pipeline for Answer Summarization (Fabbri et al.)*
- *Exploring Neural Models for Query-Focused Summarization (Vig et al.)*
- *ROUGE: A Package for Automatic Evaluation of Summaries (Chin-Yew Lin)*
- *Improve Query Focused Abstractive Summarization by Incorporating Answer Relevance (Su et al.)*
- *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al.)*
- *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (Raffel et al.)*