Status Report: Automated Presentation Generation System using Design Documents

Author **Group 40**

Sneha Bandi snehaban@usc.edu
Kirthika Gurumurthy kgurumur@usc.edu
Tejas Kashinath tkashina@usc.edu
Surya Roshan Mugada mugada@usc.edu
Rhythm Girdhar rgirdhar@usc.edu

1 Tasks Performed

1.1 Formatting/Parsing of Dataset(s)

The Natural Questions Dataset [1] was parsed to obtain the context (parsed from the Wikipedia URL provided - Natural Questions corpus requires QA systems to read and comprehend an entire Wikipedia article that may or may not contain the answer to the question), question and answer pairs required as input for training models performing question/query driven summarization. The same parsing was performed for the Tech QA Dataset [2] to ensure a uniform format for the data. For the final results, two design documents have been parsed and formatted in the manner specified above for human evaluation to be performed.

1.2 Section Identification and abstractive/extractive labelling

For the final results, where the input provided is a corpus of design documents, the correct questions need to be posed for each section required in the context of software design: What is the goal/ What is the aim?-Extractive, What are the requirements?-Extractive, What is the description? / What is the overview of the system? / What is the system architecture? / What is the architecture of the software? - Abstractive, What is the scope? Extractive Who are the users?/ Who is the audience? Extractive What are the assumptions? Extractive What are the dependencies? Extractive

1.3 BERT model for extractive Q/A

We ran a simple pre-trained BERT model as an initial step for seeing how well it performs on the design document data. For this, we use the BERT model which is fine-tuned on the SQuAD benchmark (BertForQuestionAnswering class from the transformers library). Preprocessing is performed to provide the input to the model - tokenization of question and text as a pair (token embeddings), Segment embeddings and Position embeddings. We are currently working on building a model that is fine tuned to the dataset we are using.

2 Risks & Challenges To Address

2.1 Section identification

Identifying sections in our design documents dataset is an issue faced by our proposed model. We want to be able to present properly sectioned information in our final powerpoint presentation. Achieving this is not as straightforward as it seems. Most abstractive text summarization models expect the full text of the article to be compressed as an input to the model, and this makes sense, since context is an important feature used by such models to generate a summary. An alternative we considered is to section out the text in the preprocessing step, and then summarize the text section by section. However, as suggested above, this leads to a loss of context and hence provides poor quality summarization.

2.2 Abstractive part of bert summarization

Returning to the subject of context, abstractive text summarization takes text+context as input and produces text. Due to the highly specific nature of the problem we are trying to solve, we have to explore the application of transfer learning, which means our model does not “see” the actual context that will surround the text for our intended purpose but instead will learn from a parallel corpus. For abstractive summarization, the model judges which part is important and relevant; this is problematic as relevance and importance varies by domain, and target audience. So we have to consider a tradeoff between presenting a final output that is full of jargon versus one that is more concise and captivating. Adding to these, we have some of the

[1] https://ai.google.com/research/NaturalQuestions

[2] https://research.ibm.com/publications/the-techqa-dataset

regular issues that plague abstractive text summarization. Such as hallucination - where the summary is not supported by the input text, and input length limitations. Transformers - which are the most common component of text summarization models like BERT have a maximum number of tokens they can take as input. This leads to fragmentation, and ultimately generates poor quality summarizations.

2.3 Document text length - Semantic issue

Extra noise - in most long form documents, the central idea is encompassed by boilerplate, explanations and expansions. The ideas needing to be identified and summarized are distributed across the document. Higher dimensional vectors needed to represent larger documents add to model size and complexity in training.

2.4 Text summarization Q/A issue

Most existing Q/A models produce short fact based answers for questions. While this is desirable in most scenarios, we may want more verbose structured answers for our presentations.

2.5 Small size of dataset

While fairly large and diverse datasets exist for research article documents such datasets are extremely limited for software design documents. This also adds to problems during evaluation of the final output produced by the model. We will need to have enough samples to be able to employ transfer learning and then also to evaluate the performance of our model, using human and metrics based evaluation techniques. We will work to collect an adequate number of samples to make transfer learning possible.

2.6 Transfer Learning

Time and resources are expensive because we are first training the model on a rich dataset and are then fine tuning it to work for our dataset. When there is a mismatch in the domain between the dataset for pretext tasks and the downstream task, the transfer learning may not work. The pre-trained models may converge but it will be stuck in a local minimum. The above point can be elaborated as the need to understand the domain of the dataset on what we are going to be training our model on. Although we are using a well-known TechQnA dataset, we will know of its effectiveness in our problem statement, only once we finish training our model on it with some tweaks to match ours. Although it is a tech dataset, it might not be training well enough to work for software design documents. No surety if fine tuning of the pre-trained model will perform well on our dataset in comparison with general QnA systems such as Hugging Face

3 Steps to Mitigate Risks & Challenges To Address

3.1 Section identification

For now, we are summarizing the paper as a whole, we will explore techniques such as using hybrid models to identify relevant sections to summarize before actually performing the summarization.

3.2 Abstractive bert summarization

These are common issues faced with text summarization. We attempt to mitigate some of it by employing a question answering based approach to our summarization - this helps by targeting our model to answer extremely specific question and hence produce more relevant results

3.3 Document text length - Semantic issue

As mentioned in point 1, we are exploring hybrid models and the possibility of excluding some sections which we are sure add no value to the output we are trying to produce.

3.4 Text summarization Q/A issue

We are currently in the process of exploring possible solutions to this issue.

3.5 Data Creation/Generation

In order to solve the biggest issue i.e. Data Creation/Generation, we are attempting to use novel approaches to our implementation of the software design document summarization problem. We explored the usage of transfer learning technique by training a model on an existing rich corpus like techqa. We use that pre-trained model to fine tune it on manually-created small dataset.

4 Change in Project Direction

We are limiting the scope of the original proposal by focusing on building a Q/A system specific to the software design domain first before exploring the PPT generation aspect (which is more engineering focused than research) and the FAQ generation aspect.