Question 1:

1) **Quoref** - used for co-reference resolution.

   When using this dataset, the model is trained on data in the following format:

   a) A paragraph.
   b) Several questions about the paragraph.

   What distinguishes this dataset is the fact that for each question, the answer can be identified by its name (i.e. the entity - “book”) and also by its reference (e.g. "it").

   In this way, it is possible to test whether the model was able to perform co-reference resolution for any entity -

   If the model identified the correct answer and if it’s references in the text, it succeeded (and it means that the model showed understanding of the language), and if not, the model failed.

2) **SQuad** - used for NLI.

   In this article: https://www.arxiv-vanity.com/papers/1809.02922/, the authors took various QA datasets (including SQuAD), and trained a model to do the following:

   Given a passage and a question, and a possible answer, combined the question and the possible answer into one grammatically correct sentence.

   Then, it will look at the pair (text, article) and decide whether there is an inference, contradiction, or unknown relationship between the two.

3) **QA-SRL** - used for SRL.

   Presented in this article: https://arxiv.org/pdf/1805.05377v1.pdf

   The authors of the “Large-Scale QA-SRL Parsing” paper build this dataset and used it for their model - QA-SRL parser.

   The parser solves, among others, a QA-SRL subtask - detecting argument spans for a predicate.

   The authors state:

   “Given a sentence X = x0, . . . , xn, the goal of the QA-SRL parser is to produce a set of tuples (vi, Qi, Si), where v ∈ {0, . . . , n} is the index of a verbal predicate, Qi is a question, and Si ∈ {(i, j) | i, j ∈ [0, n], j ≥ i} is a set of spans which
   are valid answers.”

Question 2:

Interactive Summarization -

1) **Definition -** a process of taking a document and summarizing it in such a way that the presented information can be interactively explored by the user according to needs and interests.
2) **Benchmarks** - according to this paper: https://aclanthology.org/D17-2019.pdf
   It looks that this is a new task, which was presented in this paper for the first time. The authors didn't supply information about the dataset they used. It looks like they were inspired by a similar (yet) different setting - presented here: https://aclanthology.org/P03-2021.pdf
3) The difficulty is possessed by the fact that this is an interactive summary and not a static one. The model needs to be able to extract and present more information about any topic that it chose to present in it's summary.

Multi-document summarization-

1) **Definition** - a process of representing a set of documents, written about a similar topic, with a short piece of text by capturing the relevant information and filtering out the redundant information.
2) **Benchmarks** -
   a) PRIMER - domain: news articles, samples - around 56,000
   b) WCEP - domain: news events, obtained from the Wikipedia Current Events Portal. Samples - 10,200 clusters with a total of 2.39M articles.
   c) DUC 2004- designed and used for testing only. It consists of 500 news articles, each paired with four human-written summaries. Specifically, it consists of 50 clusters of Text REtrieval Conference (TREC) documents, from the following collections: AP newswire, 1998-2000; New York Times newswire, 1998-2000; Xinhua News Agency (English version), 1996-2000. Each cluster contained on average 10 documents.
   d) MS^2 - a dataset of over 470k documents and 20k summaries derived from the scientific literature

3) Multi-document summarization poses the following difficulties (among others):

- Different writing styles - it is possible that a single text may have a similar writing style, but there is no guarantee of this across multiple texts. It is important to remember that the summary should be coherent and readable, so in this task, the aspect of aligning writing styles toward a clear and readable summary is also added.
- Order - in comparison to a single text summarization, whereas the order of the chosen sentnces/main ideas is implied, when dealing with multiple documents the model has to order into considieration.
- Redundency - the probability to encounter redundancy is higer when doing MDS.

Question 3:

RNN -

When using RNN, we can’t parallelize train and inference steps. The reason for that lies in the architechre of RNN and the way it works: Each token from the input sentence is proccesed sequentually. The first token relies on nothing, the second token relies on the first, the third on the first and second, and so on. Each step can’t start before the step before it has finished.

Transformer- when using Transformer, we can parallelize both training and inference. The reason, is the architechrute. The transformer uses multi-headed attention (both in the encoder and in the decoder).

Multi-head Attention is a module for attention mechanisms which runs through an attention mechanism several times in parallel. By defining h attention heads, we divide Q,K and V to smaller dimension matrices, which are independent of each other. This way, it is possible to calculate the attention of all the matrices at the same time, and at the end to concat them all.

Question 4:

a) As I see it, I need to overcome two problems:
   - Limited resources
   - A small number of examples.

   To address the first problem, I checked the size of each model and found the following results:
   - ELECTRA-base - 110M
   - T5 XXL - 11B
   - InstructGPT - 175B (GPT3)

   Since ELECTRA-base has the smallest number of parameters, this is the more reasonable choice.

   To address the second problem, I read the article about ELECTRA which claimed the following:

   Instead of masking the input, ELECTRA’s approach corrupts it by replacing some input tokens with plausible alternatives sampled from a small generator network. Then, instead of training a model that predicts the original identities of the corrupted tokens, a discriminative model is trained that predicts whether each token in the corrupted input was replaced by a generator sample or not.

   This new pre-training task is more efficient than MLM because the model learns from all input tokens rather than just the small subset that was masked out.

   As a result, the contextual representations learned by our approach substantially outperform the ones learned by BERT given the same model size, data, and compute.

   This gave me confidence that even though I have a small number of examples, the way ELECTRA works will make the most out of it.

Moreover, I read that ELECTRA can be trained in a few days on a single GPU with better accuracy than GPT, a model that uses over 30x more computing.

b) I would fine-tune it for this task (as BERT is already pre-trained).
In detail:
- The input will be pairs of sentences
- Each input text will be split into tokens
- Adding special tokens: [CLS] at the beginning of the sentence, a [SEP] between the sentences, and [SEP] at the end of the sentence.
- If needed, add [PAD] tokens to get to the needed length.
- When the BERT model was trained, each token was given a unique ID. Thus, the next step is to convert each token in the input sentence into its corresponding unique IDs.
- Feeding the pre-processed input to BERT.
- Getting the embedding vector of each token as the output from BERT
- Take the embedding vector of the [CLS] token and pass it to a softmax (to be our classifier) - to predict 0/1 (same style/ not the same style).

c) Reasons to use ChatGPT as a baseline:
- It's interesting to check if our model outperforms the current state-of-the-art
- ChatGPT (3) is free and avilable for anyone to use. Also, it doen't require special resources at all. Thus, it's very convenient and easy to use.

Reasons to not use ChatGPT as a baseline:
- We don't know on what data ChatGPT is trained on. It's possible that the data we used to test our model is in the training data of ChatGPT.
- We cannot guarantee that an exact redo of the same comparison will be possible. This is because OpenAI has the ability to update the version of

ChatGPT that was used for comparison, rendering reproducibility impossible.