1 Implementation of Decoding Algorithms

1.1 For a prompt like "Today I believe we can finally", you should show the output sequence(s) from the four decoding algorithms with the specific parameters you used (e.g., beam size, n-best, k, p). For this step, you can choose any random prompt. You must also calculate the likelihood of each output sequence by the log sum of every token logit.

I used the same prompt ("Today I believe we can finally") as in the question for generation, and the observations are noted down in the table 1.

Algorithm	Output	Parameters	Log Likelihood of each sequence
Greedy	get to the point where we can make a difference in the lives of the people of the United States of America.	max_length=30	-33.093
Beam Search	get to the point where we can make a difference in the lives of all of our children. I believe that	max_length=30, num_beams=3, early_stopping=True	-82.768
Top-K sampling	make good on our promise, and that we will continue to build on our progress, as the rest of the world does	do_sample=True, max_length=30, top_k=20	-40.484
Top-p Nucleas Sampling	bring the Bush administration back from the brink of chaos," former Bush White House chief of staff Cheryl Mills said. "And	do_sample=True, max_length=30, top_p=0.7, top_k=0	-48.569

Table 1: Columns from right to left, Algorithm, suggests algorithm used for generation, output corresponds to the generated text, parameters correspond to the parameters used for this process, and log likelihood corresponds to the loglikelihood of the output sequence.

Step 2: Decoding for downstream generation tasks

I chose summarization for task 2. The generated text from the four algorithms is saved in the spreadsheet named step2.csv.

Step 3: Automatic and Human Evaluation

Task 3.1: I used ROUGE score as the content overlap metric, and BERT score for the model-metric.

Table 2 below, shows the mean values of the metrics on different algorithms used.

	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	BERTScore-precision	BERTScore-precision	BERTScore-F1
Greedy	**0.202**	0.045	0.15	0.15	0.863	0.853	0.858
Beam	0.20	**0.046**	**0.16**	**0.16**	**0.864**	**0.859**	**0.861**
TopK	0.18	0.028	0.138	0.138	0.850	0.855	0.852
TopP	0.188	0.028	0.138	0.138	0.861	0.853	0.857

Table 2: Columns from right to left, mean ROUGE scores, and BERT score precision, recall and F1 scores of first 100 samples.

Given a reference sentence, {There is a "chronic" need for more housing for prison leavers in Wales, according to a charity}. The corresponding generated text by the algorithms mentioned above is given below.

Greedy: ,,,,,,,,, who has been in prison for 20 years, has found

Beam Search: in Wales, a Welsh charity says. "I think the key is connecting people with the services they need. It's a

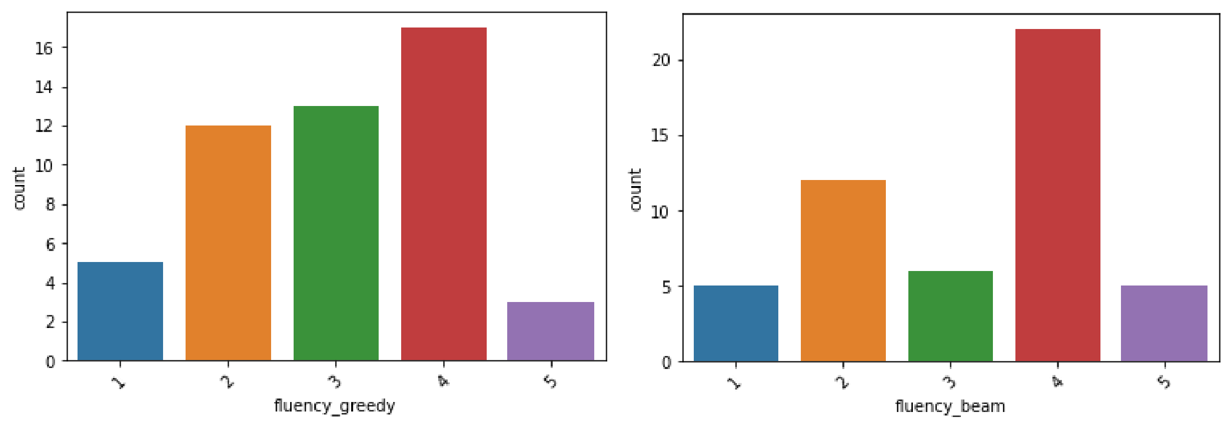
Top-K sampling: has been criticized for lack of help in the rented flat market.

Top-p sampling: the Welsh Government has warned that it is important to have a landlord, who can help other people find accommodation, to help make them

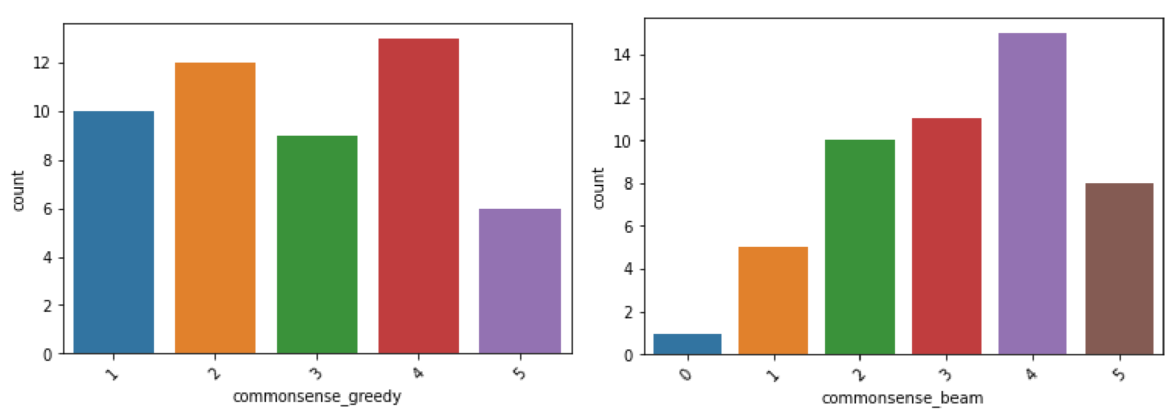
From these observations we can notice, overall Beam search showed a better performance than the rest.

3.2 Human Evaluation

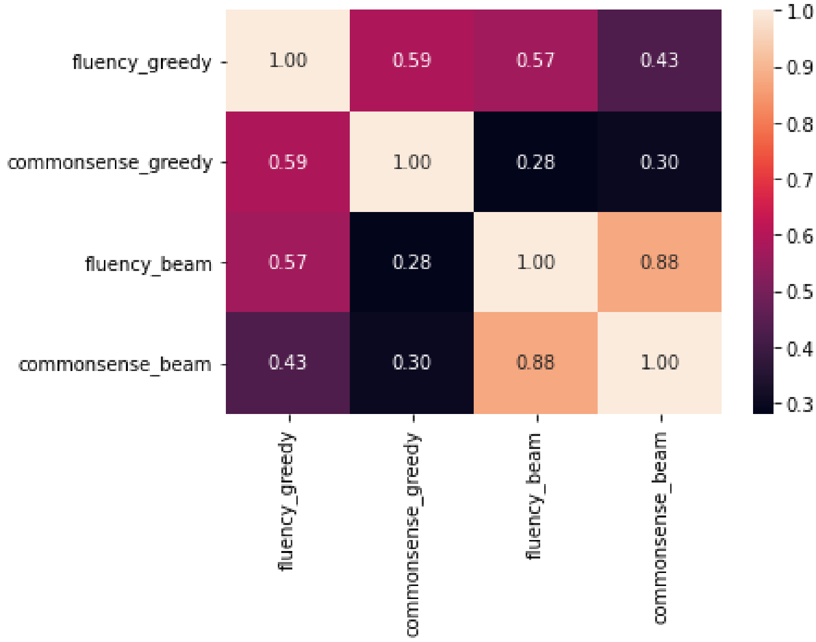
I picked Beam search and greedy search as the two algorithms for human evaluation as they were the best performing models. Even in human evaluation, Beam search is better than greedy search as per the count plots.



Fluency (Greedy vs Beam Search)



Commonsense (Greedy vs Beam Search)



Correlation map between two blind ratings