A Token Overlap Study

This is the additional study we conducted to support Section 5. We computed n-gram precision scores (BLEU-4) for mentions and captions extracted from different settings. For mentions, we included *(i)* First Mention, the sentence that first mentions the figure in the paper; *(ii)* Random Mention, a randomly selected sentence among all mentions; and *(iii)* Random Sentence, a randomly selected sentence from the paper. We also included one or two following sentences, as surrounding context may contain relevant information for the figure. For captions, we examined: *(i)* First Caption, the first sentence of the caption; and *(ii)* Whole Caption, all the sentences in the caption. The results are shown in Table 10. The extremely low BLEU-4 score for the Random Sentence baseline (First: 0.01, Whole: 0.01) indicates that a randomly selected sentence has very limited information related to the caption. In contrast, the results for First Mention (First: 9.39, Whole: 10.54) and Random Mention (First: 9.15, Whole: 10.28) show the presence of significantly more information relevant to the caption. All scores in First Mention are slightly better than the corresponding ones in Random Mention, suggesting that writers tend to give more detail when they first introduce the figure.

B Data Preprocessing Details

We describe the detailed data preprocessing steps here as supplementary materials for Section 4.

Dataset Resplit. SCICAP was originally created for vision-to-language tasks, and we needed new train/val/test splits for our work. As figures do not overlap, different figures in the same paper can be assigned to different data splits in SCICAP. However, papers' texts overlap more easily and can be problematic for text-summarizing tasks. We resplit SCICAP to make sure no paper had figures from different data splits, and we excluded figures without any identified Mentions. As a result, the [train/val/test] sets used in this work had [86,825/10,833/10,763] figures sourced from [48,603/6,055/6,053] papers, respectively.[4]

OCR. The OCR texts were extracted from all figures using EasyOCR (JaidedAI, 2022). The output from EasyOCR included the OCR texts along with their bounding boxes. In order to incorporate the OCR texts into our models, we concatenated them with the sequence of coordinates obtained by traversing the bounding boxes from left to right and then from top to bottom.

[4]The Paragraph+OCR-Better model, which used captions with more than 30 tokens, was trained on only 27,224 samples.

Setting	Maximum Length	
	Source	Target
Mention	128	100
Mention+OCR	256	100
Paragraph	512	100
Paragraph+OCR	640	100
Paragraph+OCR-Better	640	140
OCR	128	100

Table 9: Maximum length configuration for the text summarization models.

Caption	FST Mention			RDM Mention			RDM Sentence		
Context	**+0**	**+1**	**+2**	**+0**	**+1**	**+2**	**+0**	**+1**	**+2**
First	9.39	6.25	4.91	9.15	6.09	4.78	0.01	0.59	0.54
Whole	**10.54**	8.08	6.96	10.28	7.92	6.83	0.01	0.80	0.76

Table 10: N-gram matching (BLEU-4) between captions and mentions of each figure. First and Whole refer to the first sentence of the caption and the whole caption, respectively. Context means the number of the following sentences included. First Mention was better than Random Mention in the corresponding settings, suggesting that writers may give more details when first introducing the figure.

Representative. It is worth noting that we manually verified 399 figures (the set used in Section 8) and found that 81.2% (324/399) were published at academic conferences, and 51.9% (207/399) were at ACL Anthology, IEEE, or ACM, suggesting that the data is representative.

C Training and Decoding Details

We describe the model training details and the decoding configuration used in Section 7.

Training Details for Text Summarization Models. We fine-tuned Pegasus[5] for the text-summarization task using HuggingFace's implementation (Wolf et al., 2020). All the models shared the same training hyper-parameters except maximum text length, as the data varies in all the examined settings. The maximum source length and target length were set to (*i*) fully cover at least 95% of text without truncation and (*ii*) be able to fit into the machine. We show the length configuration in Table 9. Other hyper-parameters used for

[5]We used `google/pegasus-arxiv`.

training were batch size = 32, learning rate = 5e-5 with a linear decay scheduler, and number of training epochs = 200. We evaluated the model every five epochs, and the one with the highest ROUGE-2 score was kept for testing (Liu and Liu, 2021; Zhong et al., 2020; Xu et al., 2020). All models were trained with an NVIDIA A100 GPU. Each model took one to three days to train.

Training Details for Vision-to-Language Models. Two vision-to-language models were fine-tuned using HuggingFace: (*i*) a sequence-to-sequence model using BEiT[6] and GPT-2[7] and (*ii*) TrOCR.[8] The hyperparameters used for training were maximum target length = 100, learning rate = 2e-5 with a linear warmup (one epoch), and linear decay scheduler. Batch sizes were 32 and 64, respectively. The models were trained using AdamW (Loshchilov and Hutter, 2019), with weight decay = 1e-4 for 100 epochs. We evaluated the model every epoch and kept the one with the highest ROUGE-2 score (Liu and Liu, 2021; Zhong et al., 2020; Xu et al., 2020). The model was trained with an NVIDIA A100 GPU for two days.

Decoding. For all generation models, captions were decoded using the beam sampling strategy, with beam size = 5, temperature = 0.8, top-k = 100, repetition penalty = 3.0, minimal length = 10, and maximum length = 100.

D Interfaces

Figure 5 shows the interface the human judges used to rank the captions (see Section 7.2). The paper's title (without linking to the paper's URL) and abstract are shown. The human judges can drag the captions (each displayed with the figure) on the left pane and drop them to the right pane to rank them. The initial display order of the captions is randomized on the interface. We did not display the paper's PDF or link to the paper's URL to prevent human judges from biasing toward the author-written captions.

Figure 6 shows the interface we used to rate the usefulness for captions (see Section 8.2.) The title (with a hyperlink to the paper's URL), abstract, and the PDF file of the paper were shown, alongside the target figure's image/caption and all the questions. We displayed the paper's PDF to help raters make more informed decisions on the caption quality.

[6]We used `microsoft/beit-large-patch16-384`.

[7]We used `gpt2-large`.

[8]We used `microsoft/trocr-large-printed`.

E Additional Experimental Results

In this section, we show all the additional experimental results mentioned in the experiment and analysis.

Normalization Scores. Figures 8 to 11 shows the relationship between generation text length and performance (ROUGE-1, ROUGE-L, MoverScore, and BERTScore). The random lines indicate that the text length and the performance are not independent, suggesting that normalization over text length is needed. Table 11 shows the corresponding random scores for each of the metrics used in Table 2.

Examples. Figure 7 shows two samples of generation output. The information generated by Pegasus$_{P+O+B}$ could be helpful (A), but it could also introduce factual errors (B).

Performance in Different Quality Beams. Figure 12 shows the ROUGE-1 and ROUGE-L changes in beams of different quality. We can see findings similar to Section 8.1 where among different generation models, only the one trained with data quality control (*i.e.*, Pegasus$_{P+O+B}$) performed better in the helpful beam, generating captions more similar to helpful captions.

Model	Feature	Length	Rouge-1 (F1)	Rouge-2 (F1)	Rouge-L (F1)	WMS	BERTScore
			Rand	Rand	Rand	Rand	Rand
Reuse	**M**	33.2	.216	.077	.171	.524	.590
	P	238.3	.164	.088	.130	.501	.563
	W[0, 1]	50.3	.231	.087	.176	.520	.592
	W[0, 2]	68.0	.230	.092	.173	.518	.591
	W[1, 1]	67.8	.230	.092	.173	.518	.591
	W[2, 2]	98.7	.217	.095	.162	.513	.588
Pegasus	**M**	12.2	.169	.053	.144	.519	.565
	M+O	12.8	.173	.055	.147	.520	.567
	P	14.0	.181	.058	.152	.521	.570
	P+O	14.0	.181	.058	.152	.521	.570
	P+O+B	38.3	.221	.080	.172	.523	.590
	O	12.1	.168	.052	.143	.519	.565
TrOCR	**Figure**	10.0	.150	.044	.130	.517	.557
BEiT+GPT2		15.8	.190	.062	.158	.522	.574

Table 11: Random scores corresponding to the length for each automatic evaluation metric.

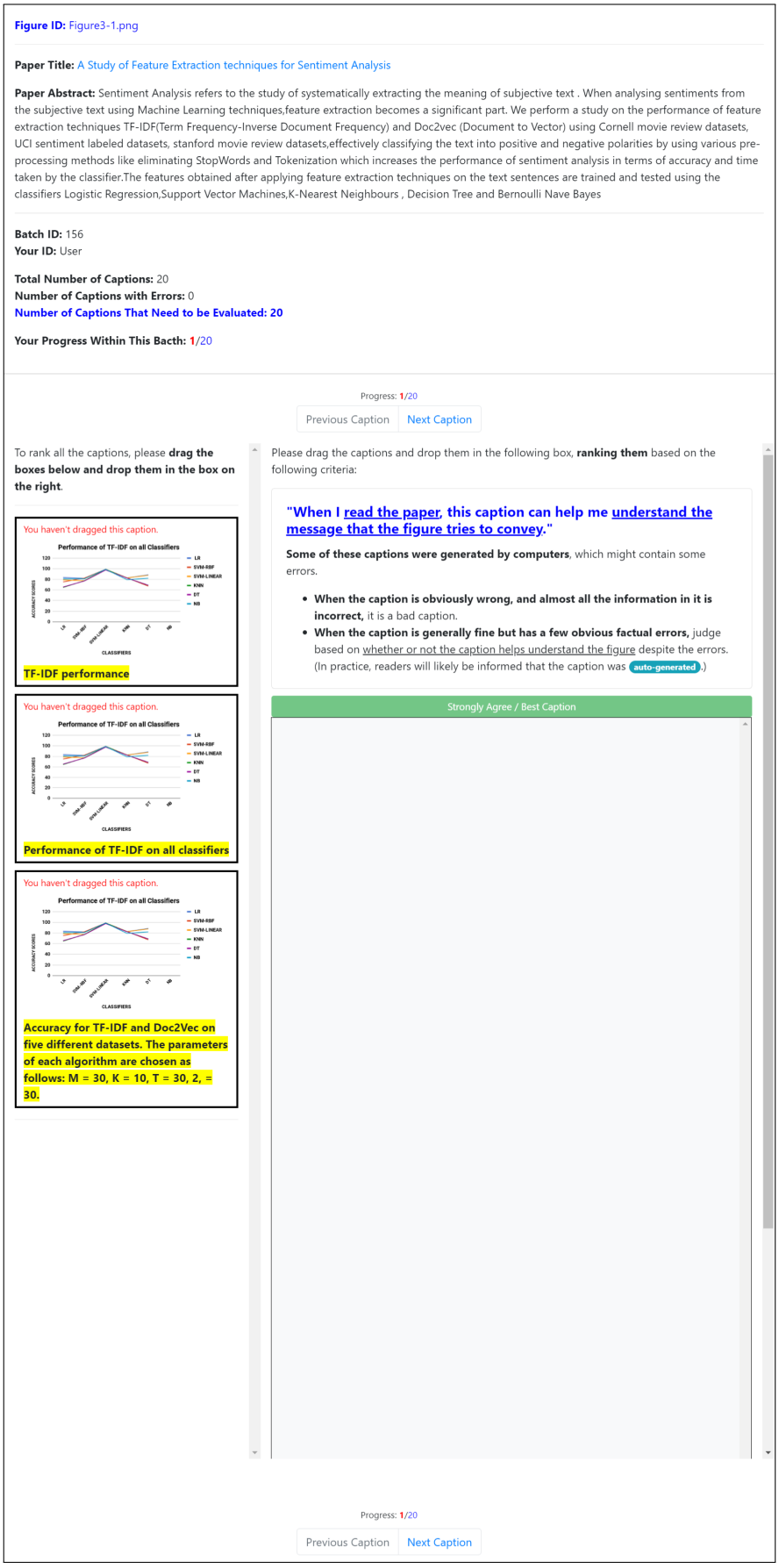


Figure 5: The interface the human judges used to rank the captions (see Section 7.2).

Figure ID: Figure2-1.png

Paper Title: Mitigating Gender Bias in Natural Language Processing: Literature Review

Paper Abstract: As Natural Language Processing (NLP) and Machine Learning (ML) tools rise in popularity, it becomes increasingly vital to recognize the role they play in shaping societal biases and stereotypes. Although NLP models have shown success in modeling various applications, they propagate and may even amplify gender bias found in text corpora. While the study of bias in artificial intelligence is not new, methods to mitigate gender bias in NLP are relatively nascent. In this paper, we review contemporary studies on recognizing and mitigating gender bias in NLP. We discuss gender bias based on four forms of representation bias and analyze methods recognizing gender bias. Furthermore, we discuss the advantages and drawbacks of existing gender debiasing methods. Finally, we discuss future studies for recognizing and mitigating gender bias in NLP.

Batch ID: 151

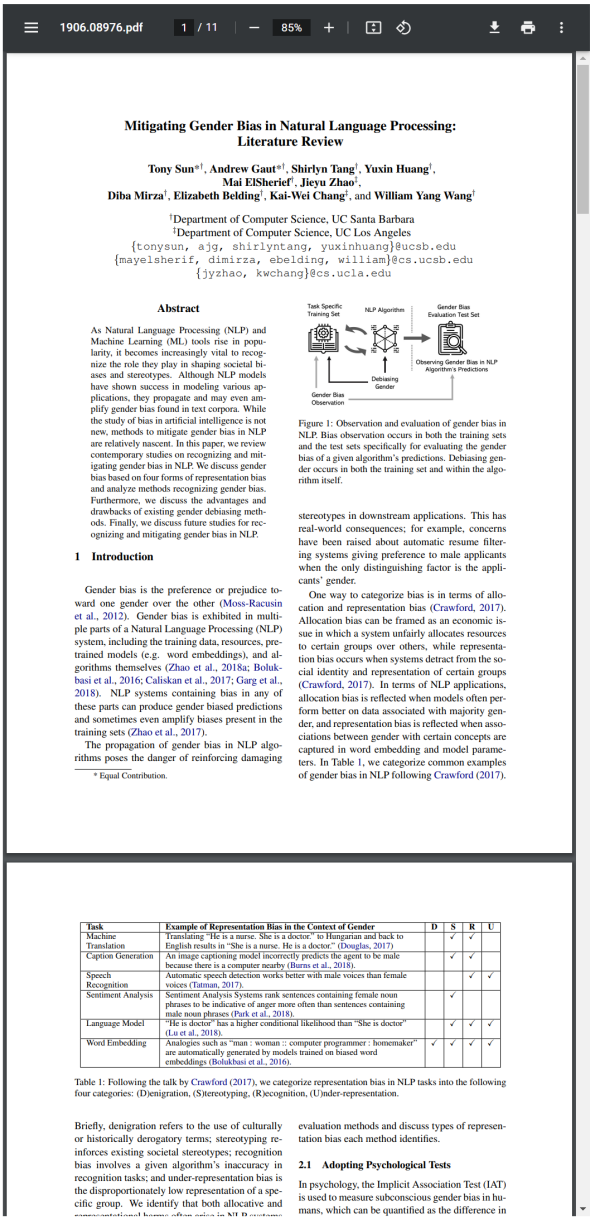
Your ID: User

Your Progress Within This Bacth: 1/20

Progress: 1/20

Previous Caption | Next Caption

Paper PDF (Please start after the PDF is completely loaded):



Please rate the caption of the following figure in this paper:

Figure 2: We project five word2vec embeddings onto the 'he' - 'she' direction before and after neutralizing the gender-neutral words maestro, instructor, and homemaker and equalizing the gender-specific pair businessman and businesswoman (Bolukbasi et al., 2018). For both x and y-axes, negative values represent male gender bias and positive values represent female gender bias.

The figure index displayed in the paper PDF is:

3 (Figure 3), 5.3 (Fig. 5.3)

Please locate the figure in the paper PDF.

Does the image or caption shown above have any issues?

- ☐ **Image Extraction Error:** The image we extracted from the PDF (shown above) has obvious errors (e.g., not containing the complete figure, containing parts that are not figures, damaged image, etc.)
- ☐ **Text Extraction Error:** The text we extracted from the PDF (shown above) has obvious errors (e.g., not containing the complete caption, containing extra text that is not the caption, incorrect text recognition, etc.)
- ☐ **Not a Line Chart:** This figure is not a line chart.
- ☐ **Compound Figure:** This figure is a compound figure that contains multiple subfigures.

1. This caption mentions some named entities or important words/numbers that appear in the figure (e.g., model names, title, x/y axis, legends, labels, etc.)

- ○ Strongly Agree
- ○ Agree
- ○ Neither Disagree Nor Agree
- ○ Disagree
- ○ Strongly Disagree

2. This caption mentions some visual characteristics of the figure (e.g., color, shape, size, trend, position, etc.)

- ○ Strongly Agree
- ○ Agree
- ○ Neither Disagree Nor Agree

Progress: 1/20

Previous Caption | Next Caption

Figure 6: The interface we used to rate the usefulness for captions (see Section 8.2).

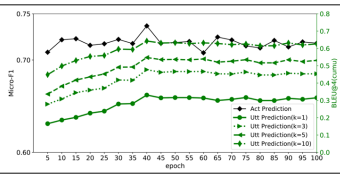
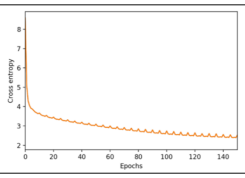
Figure	A 		B 	
	Avg. Rank	***Caption***	***Avg. Rank***	***Caption***
Original	3	Training curve of our model.	**1.33**	Evolution of training loss against epochs.
Pegasus$_{P+O}$	2	Training curve of our model for act prediction.	1.67	Convergence curve of the proposed thematicaware model.
Pegasus$_{P+O+B}$	**1**	Training curve of our method on Micro-F1 and BLEU@4(cumu). The model is trained on the act prediction task.	3	The learning process of the thematicaware model. The dashed line indicates the boundary of training set and the solid line indicates the epoch number. Note that y-axis is not linear.
Pegasus$_{M}$	-	Training curves of act prediction and utterance prediction.	-	Convergence curve of the proposed method for MNIST dataset
Pegasus$_{M+O}$	-	Training curve of act prediction on Micro-F1	-	Convergence graph of the proposed method for image classification
Pegasus$_{P}$	-	Training curve of our model on Micro-F1 and BLEU@4	-	Convergence history of thematicaware model on train dataset.
Pegasus$_{O}$	-	CDF of the number of iterations required to reach a given accuracy.	-	ROC curve of the proposed method with different number of samples.
TrOCR	-	Performance of the proposed model with respect to the number of iterations.	-	Cross entropy of the proposed model with different epochs.
BEiT+GPT2	-	Mean BLEU for different sentence lengths.	-	shows the convergence of model accuracy on training data.

Figure 7: Example output from text-summarization models. The top three rows were included in the human evaluation. Three human judges ranked the three captions for each figure (from 1 to 3 where lower is better). (A) Pegasus$_{P+O+B}$ was preferred, as it provided more details about the figure. [Image/caption source: (Jiang et al., 2019)] (B) Pegasus$_{P+O+B}$ generated a bad caption by introducing many obvious factual errors. [Image/caption source: (Wang and Zhao, 2019)]

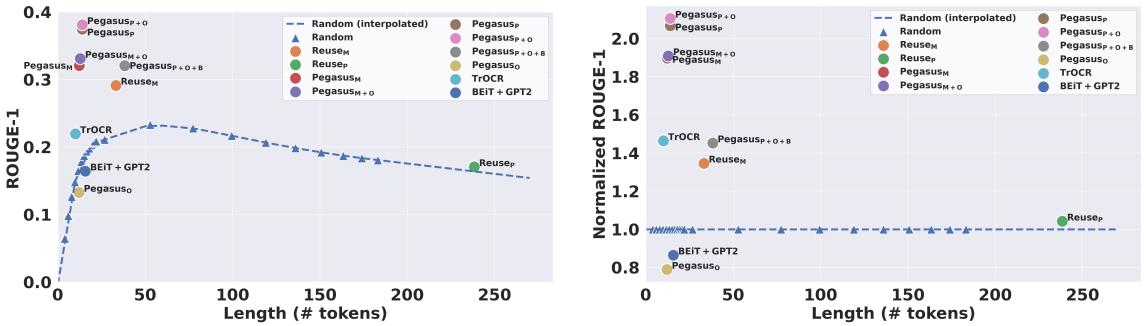


Figure 8: The relationship between average text length and ROUGE-1. When the generated text is shorter than 50 tokens, longer texts generally results in a higher ROUGE-1 score.

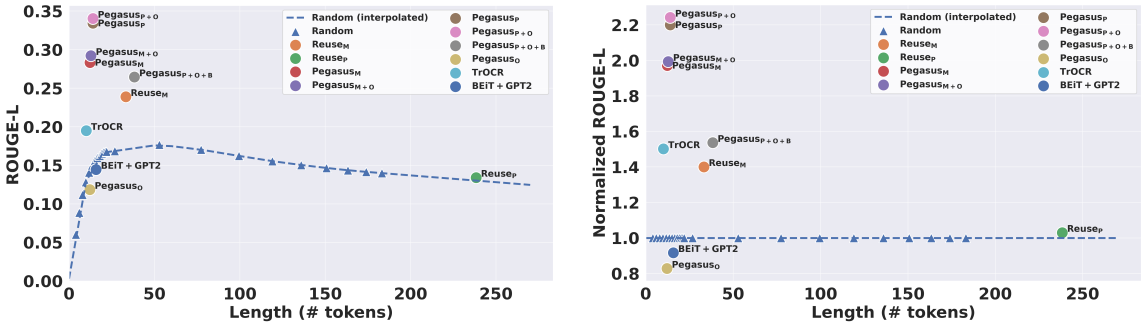


Figure 9: The relationship between average text length and ROUGE-L. When the generated text is shorter than 50 tokens, longer texts generally results in a higher ROUGE-L score.

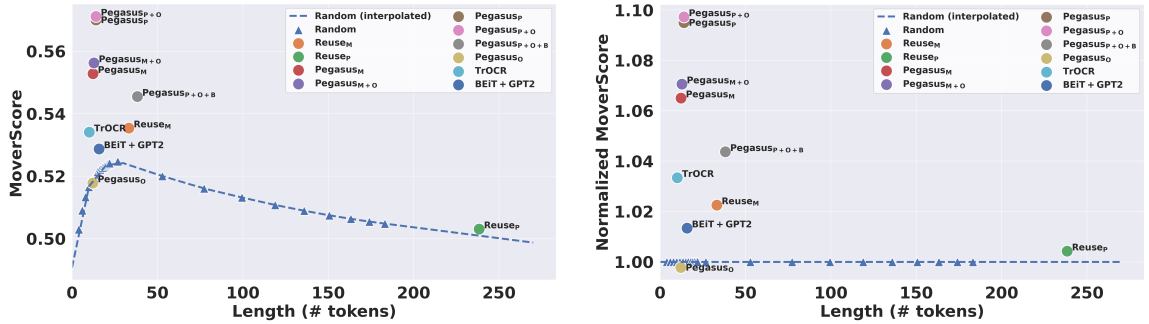


Figure 10: The relationship between average text length and MoverScore. When the generated text is shorter than 30 tokens, longer texts generally results in a higher MoverScore score.

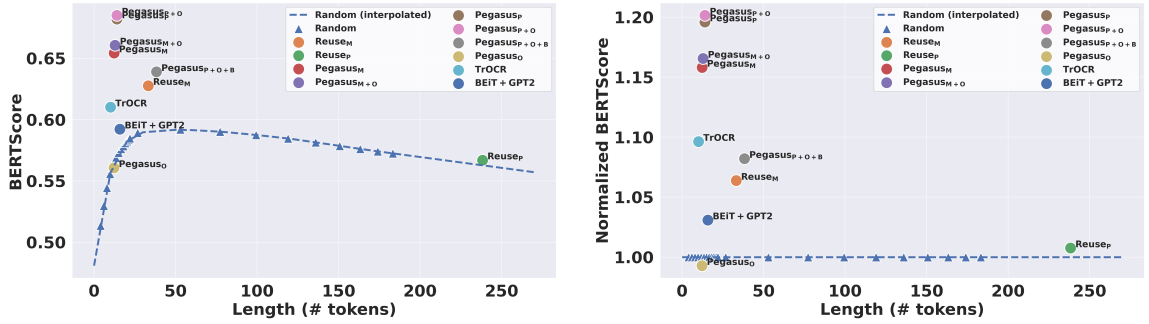


Figure 11: The relationship between average text length and BERTScore. When the generated text is shorter than 30 tokens, longer texts generally results in a higher BERTScore score.

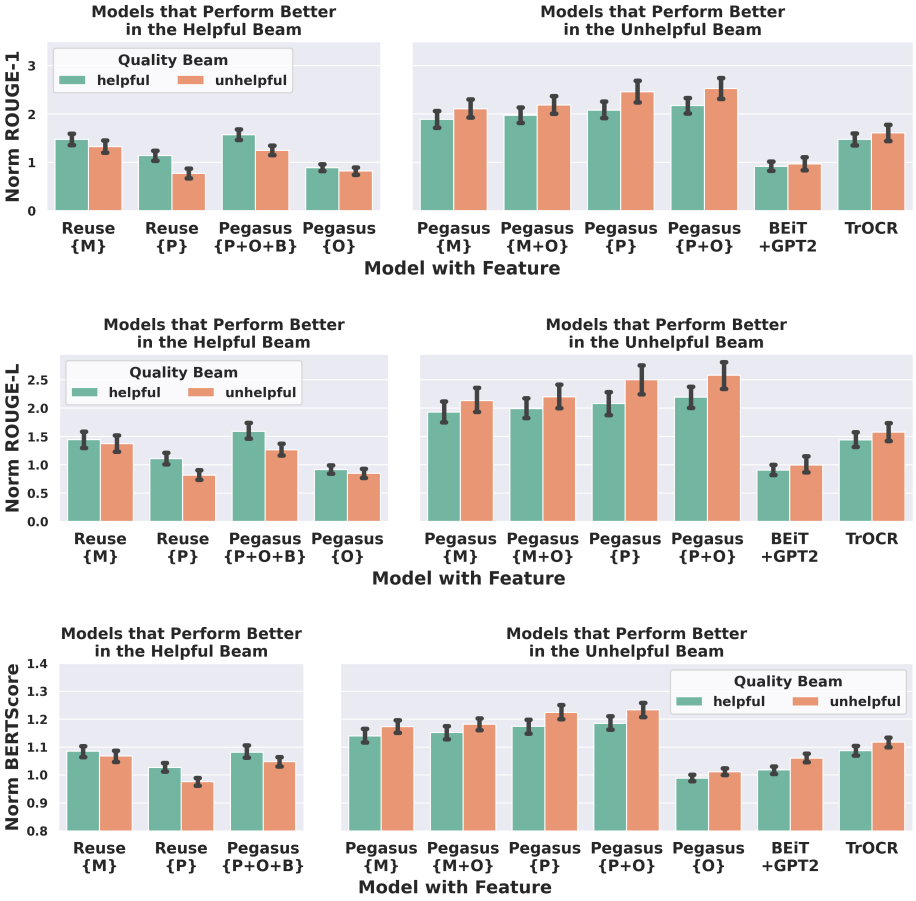


Figure 12: Normalized ROUGE-1, ROUGE-L, and BERTScore for beams of different quality. Most of the generative models (Pegasus, BEiT+GPT2, and TrOCR) performed better in the unhelpful beam, suggesting that they may be better at generating bad captions. Only the model trained with **better** captions (Pegasus$_{P+O+B}$) learned to generate good captions by showing a much better score in the helpful beam. Note that though Pegasus$_O$ also performs better in the helpful beam, the difference is subtle.