StoryBot: Interactive Story Generation For Kids

Ghiwa Lamah
ghiwa.lamah@berkeley.edu

Nicolas Loffreda
nicolof88@berkeley.edu

Ram Senthamarai
ram.senth@berkeley.edu

Abstract

Mastery of narrative skills by children is crucial for their long term success. However, modern day primary caregivers find it challenging to spend adequate time and effort on such activities. Our goal is to develop an NLP sentence continuation model that will enable an eventual interactive story development experience for children, in order to make the activity more accessible while reducing the burden on childcare providers. This system takes in the last few sentences of a story as an input, and generates the next sentence while staying on topic. To achieve this, we have utilized a corpus of children's books to fine-tune OPT, T5, and a custom BERT-to-BERT model. We evaluated the performance of these models using common NLP metrics (BLEU, ROUGE, and BLEURT) as well as human evaluation methods. Our results show inconsistent performance of fine-tuned models, where tuned models scored lower overall on human evaluation than the untuned OPT baseline. Our learnings for future research involve the need for thorough post-processing of our dataset to remove inappropriate vocabulary and improved human evaluation strategies.

1 Introduction And Motivation

The art of storytelling is an important skill to master, both in a business and social setting. It is vital that children are able to practice this early on to build adequate creative and critical thinking skills. As children are more exposed to computers and the internet than ever, it is important to create sufficient educational experiences that provide this type of skill-building. Inspired by the increasing prevalence of chatbots for customer service and question answering, this research aims to develop the base model for a children's educational storytelling chatbot experience, or StoryBot. The goal for our model is to support a story generation experience, in which a user provides a sentence and a model is able to provide the next sentence in the story.

We chose pre-trained encoder-decoder and decoder only models - T5, BERT-to-BERT, and OPT - as they lend themselves well to text generation tasks. In order to achieve the desired vocabulary, we fine-tuned the models on custom datasets we created by post-processing a corpus of children's books found on Kaggle. We tested fine-tuning with variations of one, two, or three sentences being passed as the variable, the label being the sentence to follow, and found that all models performed best with three sentences given as context. Automatic and manual evaluation showed inconsistent performance and lack of improvement over our baseline OPT model. While fine-tuning strengthened our models' performance for some metrics, it also highlighted shortcomings and potential improvement to both our chosen dataset and our evaluation strategy.

2 Background

The topic of NLP storytelling has seen increased attention and a reasonable amount of research of late. DeLucia et al explore using decoder models, specifically GPT-2, to utilize chatbot models for narrative text generation in Decoding Methods for Neural Narrative Generation, GEM 2021[1]. They used a Reddit Writing Prompts dataset to fine-tune their model, a dataset we investigated but does not provide the appropriate vocabulary for children's storytelling. Their research notes the discrepancy between metric-based and human-based evaluation results for their model, and the resulting current need for human-based evaluation for such tasks.

Research by Si, Waiman et al. (2021)[2] explored character-driven storytelling by training a multitask bi-encoder transformer on Dungeons and Dragons dialogue dataset. The goal here is to successfully be able to predict the character voicing the dialogue.

In their 2021 paper[3], Clark & Smith explore an educational game in which the user iteratively evaluates two NLP models outputs to generate a story,

essentially “choosing their own adventure”. Notably, this paper also uses a GPT-2 model fine-tuned on the Reddit Writing Prompts dataset as one of their two models, the second being the fusion model from Fan et al. (2018)[4].

3 Data

We used Children Stories Text Corpus[5] for fine-tuning our models. This dataset has one raw text file with over 200 children stories, such as The Happy Prince, The Nightingale and the Rose, The Selfish Giant, The Devoted Friend, The Remarkable Rocket, and The Ugly Duckling. We post-processed the corpus to the appropriate format for fine-tuning our models.

3.1 Dataset Processing

To post-process the data, we looped through the corpus sentences, assigning a certain number of sentences to a “variable” column, and the following sentence to a “label” column. Our initial method identified sentences by splitting the text whenever a period (“.”) is found, which led to obvious inaccuracies (for instance, phrases involving *‘Mr.’* or *‘...’* were split incorrectly). We improved our method by using the spaCy Sentencizer package, which greatly improved sentence detection.

3.2 Resulting Datasets

We initially had one fine-tuning dataset, which assigned only one sentence as the “variable”, or context. We wanted, however, to understand the impact of providing the model more context during fine-tuning. We ended up post-processing three resulting fine-tuning datasets:

- one sentence as context, labeled *S1*
- two sentences as context, *S2*
- three sentences as context, *S3*

We will use S1, S2, and S3 to refer to the datasets throughout the rest of this paper.

We used a 80/10/10 random split for training, validation, and test respectively on each dataset. We then compared the performance of the fine-tuned models to identify which tuning dataset yielded the best results.

Each resulting dataset contained over 200,000 samples (pairs of variable/label sentences). For all three datasets, even S3 which includes three sentences as context, we found that the examples are relatively short in length, with just 3% of the samples being above 150 tokens (see Appendix II for more details). This is well below the limit that modern models can handle.

4 Methodology

4.1 Model Selection

To define our baseline, we first experimented with four different base models without any finetuning: GPT2[6], OPT[7], XLNet[8] and T5[9]. The first three are decoder only models, while T5 follows an encoder-decoder architecture. Using a sample from our S1 dataset, we found that the BLEU and BLEURT scores were very low for all of them. However, generated outputs of several human chosen prompts showed OPT as the highest scoring model, so we chose to use it as our baseline.

During this phase, we also experimented with different parameters to use at inference time. Particularly, we noticed that the number of beams, the non repeated n-grams and early stopping had a significant impact on the quality of the sentences generated. Our human evaluation metrics showed that best results were obtained with 4 beams, a non-repeatable 2-gram, and early stopping. For consistency, we chose to use these parameters to generate text for all our fine-tuned models.

We then selected three models to fine tune using our datasets. OPT was the first natural choice as we wanted to understand if it could perform better after fine-tuning. OPT is a purely autoregressive model which tries to predict the next word (x_t) based on the words already predicted (x_1 to x_{t-1}):

$$max_\theta \log p_\theta(x_1, \ldots, x_T) \approx \sum_{t=1}^{T} \log p_\theta(x_t | x_1, \ldots, x_{t-1})$$

As our goal is to predict the next sentence given another one as context, we fine-tuned and evaluated OPT using S2 and S3 datasets only, excluding S1. This allowed that, at inference time, the model would've seen the whole previous sentence before trying to predict the next word.

The second model we chose is T5; while its performance was particularly poor during the baseline experimentation phase, we wanted to understand if we could leverage its Seq2Seq architecture to generate quality sentences.

Lastly, following the work from Sascha Rothe et al.[10], we implemented a custom encoder-decoder model using BERT as both encoder and decoder, BERT-to-BERT (B2B). All weights were initialized to those of BERT, and a cross attention layer was initialized randomly to bridge the 2 models. We also adjusted the decoder BERT model to only have a backward-masked attention. This made the model able to predict the next word based on the previous generated words and the context provided by the encoder:

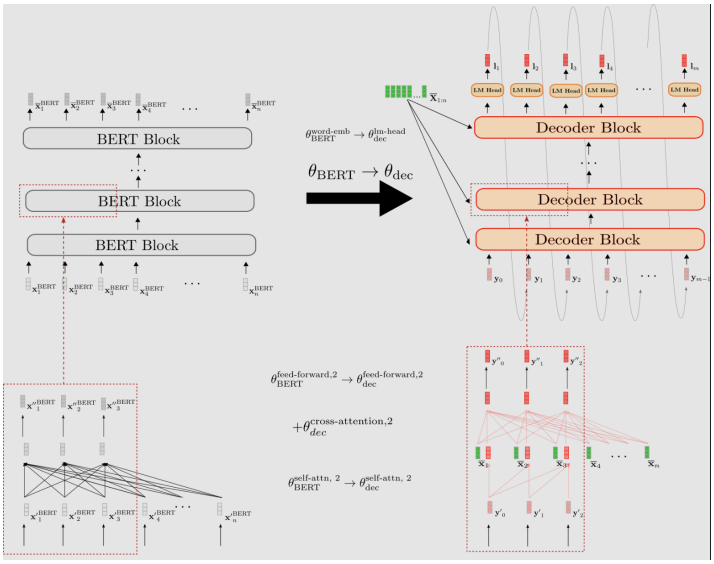


Fig 1: BERT 2 BERT Architecture

T5 and B2B Seq2Seq architecture allow the models to create a higher space representation of the input sentence from which it then generates the output. This gives the model more flexibility and is able to work properly with any given context. It would be interesting to explore as a next step how each model can begin the first sentence of a story without any context.

4.2 Hyperparameters

Given the long running times of our models, we chose to train each model only for three epochs. Even after these, we saw signs of overfitting in some models, B2B particularly, with training and validation loss beginning to diverge. As a following exercise, it could prove useful to try higher dropout levels on some layers to see if this can be mitigated.

We also experimented with the number of max tokens passed to each model. The sentences were short enough to be read by the model with about 150 tokens, but that increased training time significantly and added a lot of extra padding tokens to S1 and S2 context datasets. We decided to use a different number of max tokens for each context length: 65, 110 and 150 for S1, S2 and S3 respectively. The numbers were chosen such that 3% of the dataset was truncated when tokenized for each context length while yielding a reasonable training time of up to a couple of hours for each model.

4.3 Evaluation Strategy

Evaluating models on a creative text generation task, such as this, is a challenging task in general. For instance, an unexpected sentence continuation from the model may give the story a twist to make it more engaging and fun but will score low on metrics that compare generated text to the label like ROUGE, BLEU and BLEURT, which are the three most common automated metrics used in such NLP text generation tasks. Also, such automated metrics also do not do an adequate job on our other goals like generating kid-friendly language.

Still, we used these metrics as a first-level model and dataset selection criteria. We fine-tuned each model architecture on each of the S1, S2 and S3 datasets (except for OPT, which as mentioned earlier, we trained on S2 and S3 only). This resulted in a candidate list of eight fine-tuned models that we evaluated using BLEU, BLEURT, and ROUGE. We then selected the best performing version of each fine-tuned model. BLEU and BLEURT scores were low on all cases, but S3 consistently performed better across models, indicating that the more context the model had, the better it performed. So at the end of this process, we had a short list of four models to experiment with further using manual evaluation: Baseline OPT, Fine-tuned OPT, Fine-tuned T5, and Fine-tuned B2B, with S3 as the dataset for fine-tuning.

Since the automatic metrics were not reliable for actual model evaluation, we also devised a manual evaluation strategy following recommendations from Van der Lee et al., INLG 2019[11]. We created a list of thirty prompts, and generated five sequences per prompt from each of the four shortlisted models. Then, from each batch of five returned sequences, we selected two sequences to evaluate:

1. The sequence we selected manually as the best sentence continuation
2. The sequence that the model itself scored as the best sentence continuation

This resulted in 240 generated sequences to evaluate manually *(30 prompts x 4 models x 2 sequences = 240).* We each evaluated this final shortlist of 240 sequences using a 7 level Likert scale on five key metrics: Relevance, Readability, Grammar, Non-Redundancy, and Kid-Friendliness. We then summarized scores for each model for both the human-selected and the model-selected texts. Finally, we measured the Inter Annotator Agreement using the Krippendorff's Alpha metric, which measures a ratio of *disagreement observed* to *disagreement expected* in scores across annotators. As described by Antoine et al, EACL 2014[12], Krippendorff's Alpha is a good fit for our use case of more than two annotators evaluating criteria that is ordinal in nature.

It is important to mention that, in order to keep training and inference times reasonable, we didn't use the full models, but intermediate sizes of them as available from HuggingFace. In Particular, we used the facebook/opt-350M checkpoint for OPT, google/t5-v1_1-base for T5 and bert-base-cased for B2B. It will be an interesting exercise to try out larger models.

5 Results

5.1 Automated Metrics

All nine models, including our baseline, yielded low ROUGE, BLEU and BLEURT scores on our test dataset. This is expected, as the probabilities of the next words predicted by the model were not aligned with the words found in the original story. Figure 2 below shows the results of the baseline and top performing version of each fine-tuned model (S3 version for all three). The low scores highlight the need for manual evaluation. Full results for all nine models can be observed in Appendix III.

Model	BLEU	BLEURT	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum
B2B	0.0125	-1.0794	0.1305	0.0194	0.1114	0.1114
Baseline	0.0053	-0.9972	0.0853	0.0073	0.0656	0.0656
OPT	0.0091	-0.9241	0.1403	0.0153	0.1036	0.1036
T5	0.0131	-1.0888	0.1308	0.0204	0.1099	0.1100

Fig 2: Automated Metrics
(Higher is better)

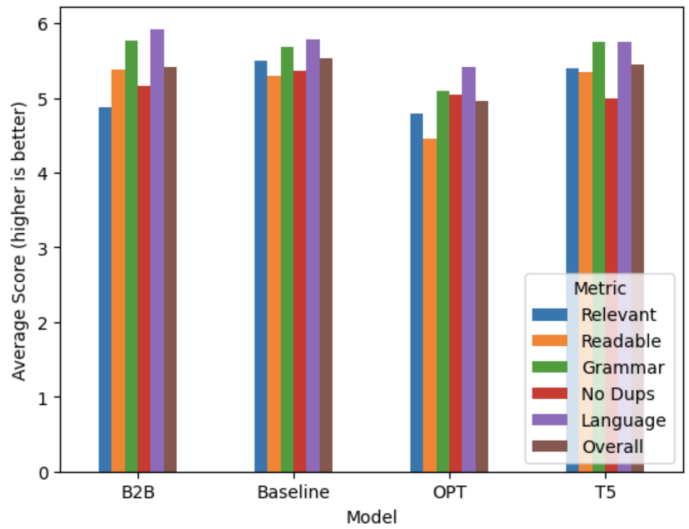
Inspection of the baseline output showed that it generated blank outputs for over 25% of the test set samples, whereas all eight fine-tuned models had at least some output text generated. In addition, despite the low scores, some samples of the generated text showed reasonable story continuation. Figure 3 below shows some of the generated text samples that are decent continuations but scored low. We also noticed that in many cases T5 and B2B generated text that is too short and generic like "Yes sir", "It is", "I don't know", and so on.

Prompt	Generated
Oh, I know! I'll make a cunning gingerbread man for the little boy who lives next door.	He'll be so pleased to have a little cake to eat, and a bit of butter and cheese to keep him warm, after he's been cold all day." "I'm sure I don't know how I'm to do it." (OPT)
The Fairy will be worried." "Poor Fairy! Is she afraid the bats will eat you up?"	"No," answered Pinocchio. (B2B)
It was not a quarter of an inch out of place. This was a great triumph for Mother Mitchel and her able assistant. But all was not over.	Suddenly a frightened voice cried out: "Mother Mitchel, what are you doing?" (T5)

Fig 3: Example Generated Texts for Fine-Tuned Models

5.2 Manual Evaluation

Figure 4 summarizes results of manual evaluation of 240 pieces of text generated on a 7 level Likert Scale: Outstanding (7), Very good (6), Good (5), Neutral (4), Bad (3), Very bad (2), Unusable (1) against 5 categories: Relevant, Readable, Grammar, Non-Redundancy (labeled as "No Dups" in Figure 4), and Language.



Model	Relevant	Readable	Grammar	No Dups	Language	Overall
B2B	4.8667	5.3778	5.7667	5.1667	5.9222	5.4200
Baseline	5.5000	5.2889	5.6889	5.3556	5.7778	5.5222
OPT	4.7889	4.4556	5.1000	5.0444	5.4111	4.9600
T5	5.4000	5.3444	5.7444	4.9889	5.7556	5.4467

Fig 4: Manual Evaluation Scores
(Higher is better)

All our models scored above average for all metrics, and all scores are reasonably close to each other, indicating that LLMs are generally strong models.

We also found that the untuned OPT model, which was our baseline, did the best overall when averaging across the scores from each category. However, compared to baseline, our custom B2B model did better in three of the five categories and came pretty close in fourth category, as shown in Figure 4. It fell very short in just one category - Relevance. Relevance was also the one category where our T5 model came very close to baseline.

Another interesting result is that while the untuned OPT model is the best overall, the same model tuned on our dataset scored the worst, especially in the relevance category. Figure 5 below shows some examples of texts generated by the two models for the same prompt.

Prompt	Tuned OPT	Untuned OPT
From all the mice in the kingdom, only one was able to save the princess, Kratos.	**He was a brave man, but he had no heart for fighting. One night he was playing with some other mice, and one of them said to him: "You are the only man in our world who can help us to get back our princess**	He saved her, not the world. And that princess was his mother.
Long ago, on the wild and windy isle of Berk, a smallish Viking with a longish name stood up to his ankles in snow.	He held out a large hand to show her the way; but the little mermaid, with her pretty hair waving loosely about her, pushed the stranger away with an air of contempt, and turned her back upon him, saying in a low voice,	**He had a flaming sword in his right hand, but his left arm was in chains. The sword had been tossed from the tower, and it looked like a cold wind had blown it over. The Viking looked at the snow and said,**

Fig 5: Untuned OPT vs Tuned OPT Outputs
Blue = Scored Higher

We also wanted to explore the impact of output length on our manual annotation results, since inspection of the test set outputs had revealed shorter lengths for T5

and B2B. Figure 6 below confirms that B2B and T5 had much shorter outputs than the baseline, with the fine-tuned OPT model having the longest outputs. We believe that this mostly explains why the tuned OPT model fared worse than the untuned OPT model. Longer outputs give more opportunity for the model to "mess up" when it comes to grammar, readability, and our other defined metrics. Further, even though B2B scored highest in some categories, we believe that is due to short outputs that would not in reality add much to continue the story. Future work would involve refinement of our manual evaluation categories to include ones that measure creativity and storytelling.

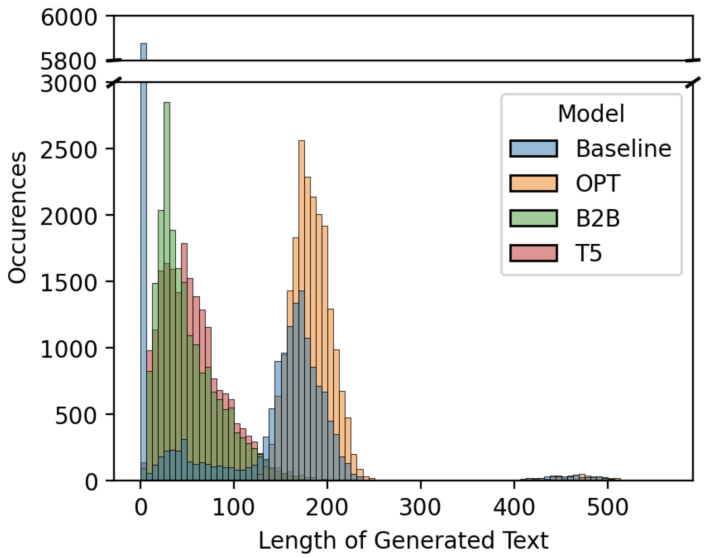


Fig 6: Generated Output Token Length Comparison

However, the manual evaluation results we obtained are colored by the fact that we do not have a good Krippendorff's alpha score, which was our inter annotator agreement metric. The Krippendorff's alpha can be a value between -1 and 0, with 1 meaning perfect agreement and -1 meaning perfect disagreement. A score closer to 0 means the results are random. We got an overall score of 0.176 which is too low. Due to time constraints we could not get to recalibrating and repeating manual evaluation. A more detailed discussion on the low IAA score is in Appendix I.

6 Conclusion

While our fine-tuned models did not show overall improvements over the baseline OPT model, our results highlight three important takeaways:

1. Dataset selection and processing has a substantial impact on the success of fine-tuning
2. Pre-trained models are susceptible to experience degraded performance during initial stages of fine-tuning
3. A manual evaluation strategy with sound criteria and clear cut guidelines is key for evaluating text generation models.

Future research involves deep-diving corpus post-processing strategies to diminish unwanted vocabulary and low quality outputs. We would like to also further explore human evaluation to build on the strategy developed for this research, by improving and supplementing the metrics chosen for annotation, and applying strict weights and guidelines to place greater focus on key success metrics.

7 Acknowledgements

We would like to thank the instructors of the MIDS W266: Natural Language Processing with Deep Learning, Spring 2023 course for their dedication, time, and constant guidance throughout the semester and project. We would especially like to acknowledge our section's instructor, Natalie Ahn, for her careful notes and feedback that helped guide this research in scope and methodology.

References

[1] Alexandra DeLucia, Aaron Mueller, Xiang Lisa Li, and João Sedoc. 2021. Decoding Methods for Neural Narrative Generation. In Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021), pages 166–185, Online. Association for Computational Linguistics.

[2] Si, W., Ammanabrolu, P., & Riedl, M.O. (2021). Telling Stories through Multi-User Dialogue by Modeling Character Relations. SIGDIAL Conferences.

[3] Elizabeth Clark and Noah A. Smith. 2021. Choose Your Own Adventure: Paired Suggestions in Collaborative Writing for Evaluating Story Generation Models. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3566–3575, Online. Association for Computational Linguistics.

[4] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

[5] Eden Bens (2021). Children Stories Text Corpus: Cleaned Gutenberg books. Retrieved Mar 1, 2023 from https://www.kaggle.com/datasets/edenbd/children-stories-text-corpus.

[6] Radford, Alec and Wu, Jeff and Child, Rewon and Luan, David and Amodei, Dario and Sutskever, Ilya (2019). Language Models are Unsupervised Multi-task Learners

[7] Susan Zhang, Stephen Roller, Naman Goyal (2022). OPT: Open Pre-trained Transformer Language Models

[8] Zhilin Yang and Zihang Dai and Yiming Yang and Jaime Carbonell and Ruslan Salakhutdinov and Quoc V. Le (2020). XLNet: Generalized Autoregressive Pretraining for Language Understanding

[9] Colin Raffel and Noam Shazeer and Adam Roberts and Katherine Lee and Sharan Narang and Michael Matena and Yanqi Zhou and Wei Li and Peter J. Liu (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

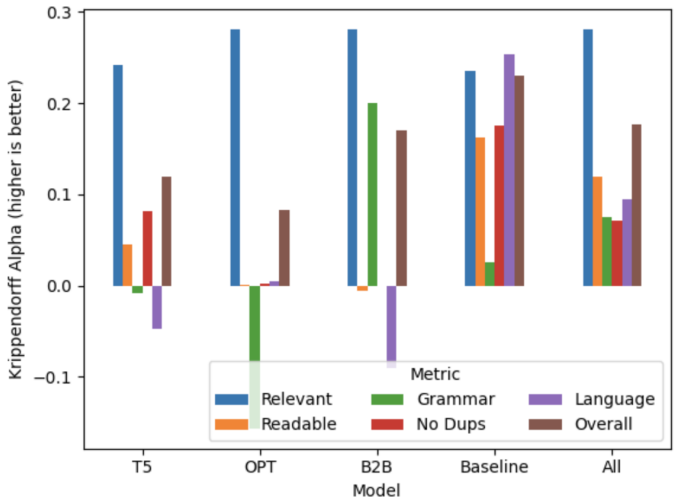
[10] Sascha Rothe, Shashi Narayan and Aliaksei Severyn (2020). Leveraging Pre-trained Checkpoints for Sequence Generation Tasks

[11] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In Proceedings of the 12th International Conference on Natural Language Generation, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

[12] Jean-Yves Antoine, Jeanne Villaneau, and Anaïs Lefeuvre. 2014. Weighted Krippendorff's alpha is a more reliable metrics for multi-coders ordinal annotations: experimental studies on emotion, opinion and coreference annotation. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 550–559, Gothenburg, Sweden. Association for Computational Linguistics.

Appendix I - IAA Scores

Krippendorff's alpha, our Inter Annotator Agreement metric of choice, measures data reliability as: $\alpha = \frac{D_e - D_o}{D_e}$, where D_o is the observed disagreement between annotators and D_e is an estimation of expected disagreement due to chance. Disagreement is measured as a distance metric that leverages the scale of our scores. K α ranges from -1 to 1. -1 represents perfect disagreement and 1 represents perfect agreement. As figure 7 below shows, we got Krippendorff's alpha, close to 0 indicating no correlation between annotators.

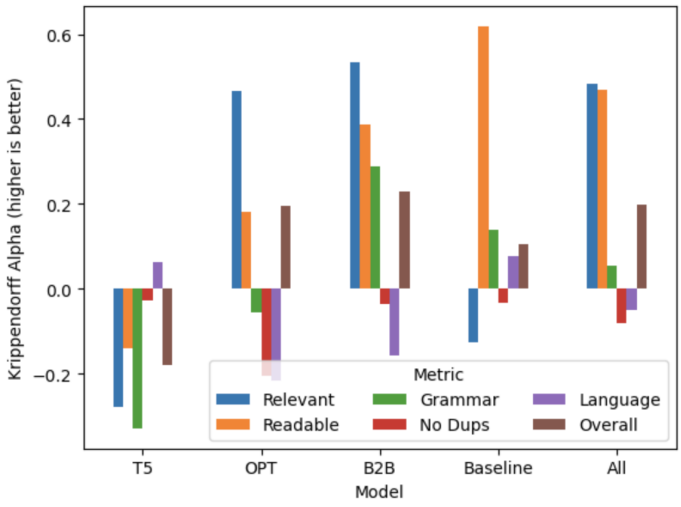


Model	Relevant	Readable	Grammar	No Dups	Language	Overall
T5	0.2418	0.0453	-0.0086	0.0821	-0.0479	0.1193
OPT	0.2811	0.0011	-0.1568	0.0016	0.0045	0.0827
B2B	0.2811	-0.0053	0.1999	-0.0011	-0.0903	0.1705
Baseline	0.2349	0.1619	0.0253	0.1751	0.2538	0.2303
All	0.2808	0.1195	0.0755	0.0711	0.0947	0.1766

Fig 7: Krippendorff Alpha (Human Selected Text)
(Closer to 1 is better)

Key observations:

- We are in better agreement on the quality of the baseline results than other three models.
- Across the board, we had better agreement on the Relevance.
- We had bigger disagreements on Grammar and Language.
- Interestingly, we had better agreement on the model-selected text, as can be seen in Figure 8 below.



Model	Relevant	Readable	Grammar	No Dups	Language	Overall
T5	-0.2778	-0.1389	-0.3288	-0.0273	0.0622	-0.1802
OPT	0.4665	0.1802	-0.0557	-0.2048	-0.2150	0.1955
B2B	0.5351	0.3886	0.2886	-0.0356	-0.1557	0.2283
Baseline	-0.1250	0.6188	0.1402	-0.0317	0.0778	0.1064
All	0.4833	0.4679	0.0540	-0.0805	-0.0486	0.1970

Fig 8: Krippendorff Alpha (Model Selected Text)
(Closer to 1 is better)

Below table in figure 9 has some examples where we disagreed on relevance.

Prompt	Generated	Relevance		
		A1	A2	A3
My stomach did a flip, then a flop, I couldn't believe what I was seeing.	"Come here, come here!" cried John.	4	4	6
My stomach did a flip, then a flop, I couldn't believe what I was seeing.	I knew that I had better not look back, and run down and call the dog, so I did. He came to me in a great hurry. inside of a minute he called to the man who was to be my guardian.	2	4	6

Fig 9: Examples Of Disagreement

Our key learnings from this valuable experience:

- When defining metrics, one should also define what each metric means and how to evaluate it and cover corner cases.

- Annotator calibration on the metrics is very important. A round of evaluation should be done on a much smaller throw-away dataset to calibrate.
- It is easy to underestimate time needed for manual annotation. Adequate time should be allocated to avoid fatigue.
- It is very hard to avoid bias creeping in. It is best to get neutral third parties as annotators.

Appendix II - Token analysis

Most datasets have sentences with less than 150 tokens, even on the S3 one as shown on Figure n. below. Only 3 examples on S3 have more than 512 tokens, the maximum number of tokens accepted by BERT. On the other hand, OPT can accept 1024 and T5 can accept an arbitrary number of tokens.

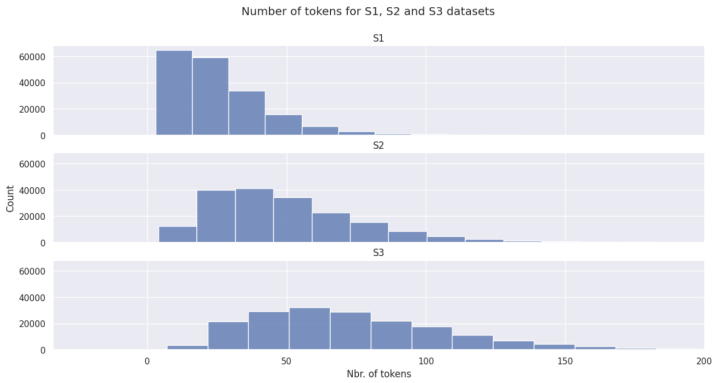


Fig 10: Tokens Per Dataset (S1, S2 and S3)

Appendix III - Experiment results

Models perform consistently better when using the S3 dataset and providing more context to generate the next sentence.

Model	BLEU	ROUGE1	ROUGE2	ROUGEL	ROUGELSUM	BLEURT
Untuned OPT	0.005272	0.085343	0.007274	0.065647	0.065646	-0.997209
OPT S2	0.008313	0.137038	0.013192	0.101051	0.101071	-0.949812
OPT S3	**0.009127**	**0.140313**	**0.015348**	**0.015348**	**0.103603**	**-0.924052**
T5 S1	0.007994	0.125251	0.010452	0.101473	0.101447	-1.108518
T5 S2	0.011892	0.132282	0.017951	0.110494	0.110561	-1.107098
T5 S3	**0.013134**	**0.130802**	**0.020352**	**0.109915**	**0.109953**	**-1.08875**
Bert2Bert S1	0.010602	0.119237	0.014406	0.101568	0.101565	-1.109143
Bert2Bert S2	0.011956	0.126579	0.017824	0.108175	0.108119	-1.098777
Bert2Bert S3	**0.012474**	**0.130512**	**0.019387**	**0.111407**	**0.111362**	**-1.079424**

Fig 11: Auto-evaluation experiment results