



Master's Thesis

Data-to-Text Generation for Games

Factual Verbalization and Character-Conditioned Commentary

Yanjian ZHANG

Advisor: Mélanie LOPEZ MALET

August 2023

Acknowledgements

This work was carried out during an internship at Ubisoft. I thank my supervisor, Mélanie LOPEZ MALET, for her guidance and support. I thank Adrien ZHANG for assistance with database queries, and Eve Ben-Haïm and Anaïs Desfachelles for their evaluation criteria and feedback. I also thank Rémi LABORY, the Ubisoft paper-reading group, and the Player Analytics France team for their discussions and support. Finally, I thank my wife, Huihui CHI, for her support throughout the internship.

Abstract

Generating language from game data requires both preserving event information and expressing it in a voice appropriate to the speaker. This thesis investigates these requirements through two complementary tasks: factual verbalization of game-event records and tone-controlled or character-conditioned commentary. For factual verbalization, we implement a modular pipeline that serializes records, generates individual sentences, and combines them through sentence fusion. For commentary, we condition generation on tone descriptions and character profiles, then construct a synthetic corpus through API-generated seeds, human review, few-shot expansion with Llama 2, model-based filtering, and text cleaning. The resulting corpus contains 8,881 training, 426 validation, and 304 test examples covering seven characters. We use this corpus to train character-specific low-rank adapters for a FLAN-T5-based model. Evaluation of 30,000 verbalized records finds that 94.5% retain every input value under a value-matching criterion. Held-out commentary evaluation combines MAUVE, ROUGE-L, and BLEU-4 with qualitative analysis of shared-scene outputs. The examples exhibit character-associated expressions, while the different metric rankings distinguish distributional similarity from lexical overlap. Together, the study provides a documented workflow connecting structured game data, explicit style conditioning, and compact model adaptation, with an empirical account of the resulting descriptions and commentary.

Keywords: data-to-text generation; game commentary; character conditioning; synthetic data; parameter-efficient fine-tuning.

Contents

Abstract	ii
1 Introduction	1
1.1 Motivation and Research Context	1
1.2 Research Problem	1
1.3 Approach	2
1.4 Research Questions and Contributions	2
1.5 Thesis Organization	2
2 Industrial Context	3
3 Related Work	4
3.1 Factual Text Generation	4
3.2 Game Commentary	4
3.3 Logical Table-to-Text Generation	4
3.4 Language Models and Instruction Following	5
3.5 Supervised Instruction Tuning and Human Feedback	5
3.6 Parameter-Efficient Fine-Tuning	5
3.7 Prompt Design	5
4 Task Formulation	6
4.1 Data Serialization	6
4.2 Factual Text Generation	6
4.3 Conditioned Commentary Generation	6
4.4 Parameter-Efficient Adaptation	7
5 Data and Generation Framework	8
5.1 Data Sources and Representation	8
5.1.1 Game-Event Records	8
5.1.2 Character-Scene Inputs	9
5.2 Factual Text Generation	10
5.3 Event-Conditioned Commentary	11
5.3.1 Data Preprocessing	11
5.3.2 General Prompt	12
5.3.3 Hosted Instruction-Following Models	12
5.3.4 Local Instruction-Following Models	13
5.4 Tone-controlled Comment Generation	13
5.5 Character-Conditioned Commentary	14
5.5.1 Dataset Construction Pipeline	14
5.5.2 Seed Comment Generation	15
5.5.3 Few-shot Generation with Llama 2	15

5.5.4	Model-based Quality Verification	16
5.5.5	Corpus Cleaning	16
5.6	Chapter Summary	17
6	Experiments and Evaluation	18
6.1	Experimental Setup	18
6.2	Preliminary Adaptation Experiments	18
6.3	Evaluation Metrics	18
6.4	Learning Dynamics	19
6.5	Held-Out Commentary Results	20
6.6	Qualitative Analysis	21
6.7	Discussion	23
6.8	Limitations and Reproducibility	23
6.8.1	Data Integrity and Reproducibility	23
6.8.2	Experimental Scope	23
7	Conclusions and Future Work	25
7.1	Conclusions	25
7.2	Future Work	26
A	Prompt Templates and Experimental Listings	27
B	Complete Qualitative Examples	31

List of Figures

1.1	Data-to-text generation in a game setting.	1
1.2	Examples of factual descriptions and commentary.	2
5.1	Action	8
5.2	New Status	8
5.3	Damage	8
5.4	Factual generation through serialization, verbalization, concatenation, and sentence fusion.	11
5.5	Synthetic commentary construction through seed generation, review, expansion, and filtering.	15
6.1	Training-loss trajectories for the preliminary full-model and character-specific LoRA runs.	19
6.2	Validation MAUVE trajectories for the preliminary adaptation runs.	19
6.3	Training loss by optimization step for character-specific adapters.	20
6.4	Validation MAUVE by training epoch.	20
6.5	Validation ROUGE-L by training epoch.	21
6.6	Validation BLEU-4 by training epoch.	21

List of Tables

5.1	Game-event records sharing a single action identifier.	9
5.2	Example character profile used to construct character–scene inputs	9
5.3	Design considerations for factual generation.	10
5.4	Design considerations for commentary generation.	11
5.5	Aggregated representation of the event in Table 5.1.	12
5.6	Input event used for the qualitative model comparison.	12
5.7	Hosted-model outputs for the event in Table 5.6.	12
5.8	Local-model outputs and recorded CPU generation times for the same event. .	13
5.9	Tone-conditioned outputs and reviewer acceptance counts out of five examples per tone.	14
5.10	Human review outcomes for seed comments, before revision of partially accepted outputs.	15
5.11	Retained synthetic commentary examples by character and split	17
6.1	Character-specific evaluation on held-out synthetic references. n is the number of test examples. Scores are reported on their original scale; trailing zeros standardize presentation without adding measurement precision.	22
6.2	Selected excerpts from adapted FLAN-T5 outputs for a shared floating-island scene. Excerpts are taken from the full outputs in Appendix B; the final column describes an observable feature rather than a quality score.	22
B.1	Few-shot Llama 2 and adapted FLAN-T5 outputs for one scene and three character profiles.	32

Chapter 1

Introduction

1.1 Motivation and Research Context

Data-to-text generation converts structured records into natural language. In video games, event logs and observations provide a basis for descriptions of player actions, changes in the environment, and interactions with non-player characters (NPCs). Such applications require both factual fidelity and control over the voice in which information is presented.

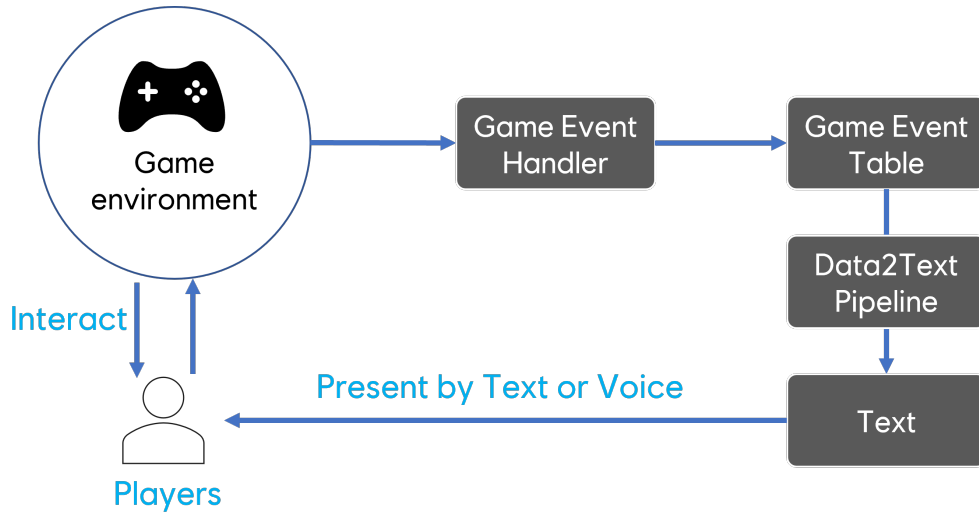
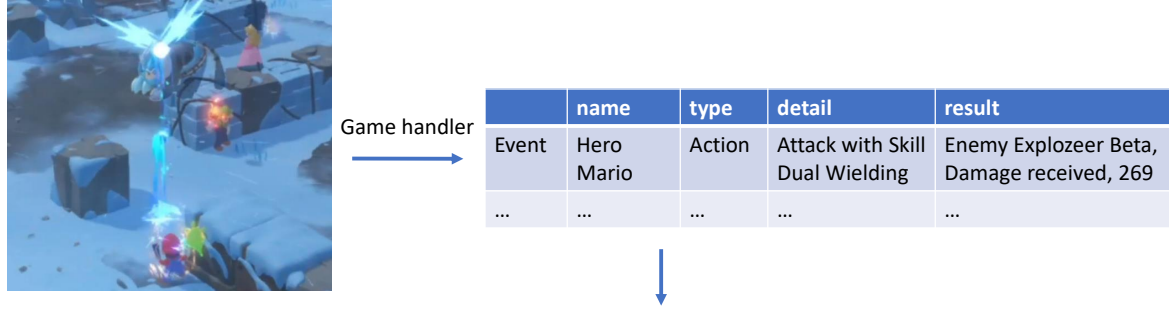


Figure 1.1: Data-to-text generation in a game setting.

1.2 Research Problem

This thesis studies two complementary tasks: factual text generation and commentary generation. The former verbalizes the information contained in game records; the latter interprets an event or observation in a specified tone or character voice. Figure 1.2 illustrates this distinction.



Factual Text: Mario attack Explozeer Beta with Dual Wielding, causing 269 damage.

Comment: It's-a me, Mario! I'll-a show you that-a you don't have-a a chance against my-a skill!

Figure 1.2: Examples of factual descriptions and commentary.

1.3 Approach

For factual generation, we investigate a modular pipeline that separates row-level verbalization from sentence fusion. This design follows ASDOT[38] and supports the use of relatively small models. For commentary, we investigate explicit tone descriptions and character profiles as conditioning information. We then construct synthetic training data through seed generation, human review, few-shot expansion, and model-based filtering, and adapt a FLAN-T5-based model using low-rank adaptation (LoRA).

1.4 Research Questions and Contributions

The study addresses three questions. First, can a modular pipeline preserve the values in game-event records? Second, can explicit instructions elicit commentary with different tones and character styles? Third, can a compact, character-specific model learn from filtered synthetic examples? The contributions are an application pipeline, a documented data-construction procedure, and an exploratory evaluation across seven characters.

1.5 Thesis Organization

Chapter 2 describes the industrial setting. Chapter 3 reviews related research. Chapter 4 defines the tasks and notation. Chapter 5 presents the datasets and generation pipelines. Chapter 6 reports the experiments and their limitations. Chapter 7 relates the findings to the research questions and outlines future work. Appendix A records the full experimental prompts.

Chapter 2

Industrial Context

This work was conducted during an internship in Ubisoft’s Player Analytics France (PAF) department. The team brings together expertise in data analysis, data science, software development, and visualization to support game production and the analysis of player behavior.

The project investigates how structured game data can support natural-language interactions. Its practical requirements include factual fidelity, stylistic control, deployment cost, and inference latency. These requirements motivate the separation of factual descriptions from character-conditioned commentary and the investigation of compact models.

The industrial setting provides structured event records and writer-authored character profiles. Together, these inputs motivate a workflow that preserves event information while conditioning its expression on the intended speaker.

Chapter 3

Related Work

This chapter situates the project within data-to-text generation, game commentary, instruction following, and parameter-efficient adaptation. The discussion connects these research directions to the two generation tasks studied in this thesis.

3.1 Factual Text Generation

Data-to-text benchmarks differ in their input representations and generation objectives. E2E[22] uses meaning representations, WebNLG[8] uses sets of triples, and DART[21] brings together structured inputs from several sources. ToTTo[24] conditions generation on highlighted table cells. The game records studied here are tabular and require domain-specific interpretation of actions, status changes, and damage.

Template-based approaches provide explicit control over output structure but require manual design and maintenance. Blabbeur[14] illustrates this approach in a game-development setting. Neural pipelines instead decompose generation into intermediate operations. Kasner and Dušek[13] study a modular pipeline, while ASDOT[38] separates data disambiguation from sentence fusion. This thesis adopts the latter decomposition for factual descriptions.

Other approaches investigate graph representations[28], coverage-oriented search[12], verbalization for question answering[20], multitask generation[31], and contrastive learning[2]. These studies motivate attention to information coverage and factual fidelity, although the present experiments do not compare all of these methods.

3.2 Game Commentary

Esports commentary datasets[35] and MOBA-E2C[39] investigate the generation of commentary from competitive game events. The present project focuses on an additional application requirement: conditioning the output on explicit tone descriptions or NPC profiles.

3.3 Logical Table-to-Text Generation

LogicNLG[5] studies generation requiring logical operations over tables. Logic2Text[6] provides explicit logical forms, MURMUR[29] investigates intermediate reasoning, and PLOG[17] explores pretraining for table-to-logic generation. These works highlight the difference between mentioning values and reasoning correctly about them. Accordingly, the value-coverage measure used in this thesis should not be interpreted as a comprehensive factuality metric.

3.4 Language Models and Instruction Following

Contextual representation learning, including ELMo[25], and Transformer-based pretraining, including BERT[10], contributed to the development of pretrained language models. These are distinct architectures and training approaches rather than a direct sequence of model upgrades. T5[27] expresses tasks in a text-to-text format. FLAN[36] and the FLAN Collection[19] investigate instruction tuning and transfer across tasks. Chain-of-thought prompting[37] provides another way to structure model inputs, although this thesis does not systematically evaluate it.

3.5 Supervised Instruction Tuning and Human Feedback

Instruction tuning and reinforcement learning from human feedback (RLHF) are related but distinct procedures. The use of instruction-response examples does not by itself imply that a model was trained with RLHF. Alpaca[32] and Dolly[9] illustrate instruction-tuning workflows, while Llama 2[34] describes supervised and preference-based alignment for its chat models. The experiments here use particular historical checkpoints; their behavior should not be attributed to one training component without controlled evidence. Licensing conditions are model-specific and should be checked separately before redistribution or deployment.

3.6 Parameter-Efficient Fine-Tuning

Parameter-efficient methods adapt pretrained models while updating a restricted set of parameters. Prefix-tuning[15] learns continuous conditioning representations; control prefixes[7] extend this idea for controlled generation. LoRA[11] learns low-rank weight updates. This thesis uses LoRA to construct separate character-specific adapters and reports the resulting automatic evaluation scores.

3.7 Prompt Design

Prompting specifies a task through instructions, context, and optional demonstrations[18]. Few-shot examples can guide generation without updating model parameters[4]. In this project, prompts convey event schemas, tone definitions, character profiles, and quality criteria. Their effectiveness is examined through exploratory examples and synthetic-data filtering rather than a systematic optimization of prompt variants.

Chapter 4

Task Formulation

4.1 Data Serialization

Let T be a table with n rows and m columns. Column j has key k_j , and its value in row i is v_{ij} . A row is represented as an ordered sequence of key–value pairs:

$$r_i = ((k_1, v_{i1}), \dots, (k_m, v_{im})), \quad 1 \leq i \leq n. \quad (4.1)$$

A serialization function S converts this representation into a textual model input:

$$x_i = S(r_i) = \text{concat}_{j=1}^m (\text{str}(k_j) \parallel : \parallel \text{str}(v_{ij}) \parallel .). \quad (4.2)$$

Here str converts a key or value to text, $\parallel$ denotes string concatenation, and concat joins the field strings with separating spaces. This representation follows the key–value format used in the project, drawing on instruction-formatted inputs[19].

4.2 Factual Text Generation

Following the decomposition in ASDOT[38], row-level verbalization produces a sentence f_i , and sentence fusion combines the resulting sequence:

$$f_i = F_{\text{verb}}(x_i), \quad i = 1, \dots, n, \quad (4.3)$$

$$y_{\text{fact}} = F_{\text{fusion}}(f_1, \dots, f_n). \quad (4.4)$$

The intended output preserves the information in the input while reducing repetition.

4.3 Conditioned Commentary Generation

Let x denote a serialized event or observation and I an instruction. Let C_{tone} describe a requested tone and C_{char} describe a character’s identity and speaking style. The generation settings are

$$c = F_{\text{comment}}(x, I), \quad (4.5)$$

$$c_{\text{tone}} = F_{\text{comment}}(x, I, C_{\text{tone}}), \quad (4.6)$$

$$c_{\text{char}} = F_{\text{comment}}(x, I, C_{\text{char}}). \quad (4.7)$$

For event commentary, x may encode several records aggregated by action identifier. For observation commentary, it encodes a location and visible entities. Commentary permits interpretation and expressive language, but should remain consistent with the supplied context.

4.4 Parameter-Efficient Adaptation

LoRA[11] freezes a pretrained weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and learns a low-rank update:

$$W = W_0 + \frac{\alpha}{r}BA, \quad A \in \mathbb{R}^{r \times d_{\text{in}}}, \quad B \in \mathbb{R}^{d_{\text{out}} \times r}. \quad (4.8)$$

Here r is the adaptation rank and α the scaling factor. The experiment uses $r = 32$ and $\alpha = 32$. This equation describes the adaptation method; it does not imply that the number of parameters of the base model increases by the trainable-parameter percentage.

Chapter 5

Data and Generation Framework

This chapter develops the generation framework from its inputs to the final adaptation corpus. We first describe the two data sources, then explain factual verbalization and event commentary, and finally present the construction of character-conditioned training examples. Full prompt templates and model-generated descriptor lists are collected in Appendix A.

5.1 Data Sources and Representation

The study uses two data sources: game-event records extracted from Mario + Rabbids Sparks of Hope, and character-scene inputs assembled from writer-authored profiles and scene observations. We refer to these sources descriptively as *game-event records* and *character-scene inputs*. The *synthetic commentary corpus* consists of the generated, filtered responses used for adaptation.

5.1.1 Game-Event Records

Game-event records are extracted by an event handler for Mario + Rabbids Sparks of Hope. They cover three event categories:

1. **Actions:** attacks, techniques, item use, movement, and empowerment (Figure 5.1).
2. **Status changes:** conditions such as airborne, burning, or dead (Figure 5.2).
3. **Damage:** damage associated with an action (Figure 5.3).



Figure 5.1: Action



Figure 5.2: New Status



Figure 5.3: Damage

An action may produce multiple status changes and damage records. Table 5.1 groups records with the same action identifier. In this example, Rabbid Mario uses Counter Fist; Stooge Alpha and Stooge Beta acquire the burning status, and Stooge Alpha also acquires the dead status.

Table 5.1: Game-event records sharing a single action identifier.

Action ID	Character	Faction	Event Type	Action Type	Details
0001	Rabbit Mario	Hero	Action	Attack	Skill Counter Fist
0001	Stooge Alpha	Enemy	New Status	Burning	-
0001	Stooge Alpha	Enemy	New Status	Dead	-
0001	Stooge Beta	Enemy	New Status	Burning	-

5.1.2 Character–Scene Inputs

Character–scene inputs combine character context with observations of the game environment.

Character Context

Each character profile specifies a name, identity, dialogue style, conversational habits, colloquial expressions, and example utterances. Table 5.2 presents an excerpt from Liza’s profile; the supplied profile text and utterances are reproduced as experimental inputs.

Table 5.2: Example character profile used to construct character–scene inputs

Character	LIZA
Identity	Liza is optimistic, cheerful, radiant and very sociable. She makes friends wherever she goes. She’s already traveled all over the world, and what she loves most of all is meeting people. She’s a real party girl, a night owl. she wants to be light and free.
Dialogue style	Liza is very curious and asks lots of questions. She talks very fast, and her boundless enthusiasm sometimes prevents her from breathing (missing periods, commas... air...). Her enthusiasm translates into words in capital letters and a frenzy of exclamation marks. She contracts a lot of words (ex : that’s hilar’ = that’s hilarious)
Habits	Liza is very sociable and her language is friendly and unformal. She speaks in the superlative. Everything is brilliant splendid or boujee (slang gen z for fancy). She often announces that she’s telling the truth by saying no cap. And she talks to everyone the same way from the President of the United States to the waiter at the airport bar.
Colloquialism	She uses a lot of Generation Z slang, and as soon as someone says something sexist or retrograde, she calls them a boomer. She often starts her sentences with Jiz. Sheuses swear words. She talks about sex freely and also knows queer slang.
Example 1	I fuckin’ love your energy dude, no cap. Did I already met you ? Maybe in Cancun ? Well I was colocado back there, but who is sober at the full moon party righ’ ?
Example 2	Did you know that cave men and women slept on top of each other with the whole family! ! Today we cherish our living space in a weird MY PRECIOUS way!!! I don’t miss canibalism but I do miss that closeness in this day and age.
Example 3	OMG we were at this overrated restaurant with a date. There was a band playing and we decided to dance like no one was there. My kind of boujee date.
Example 4	Sorry, I know I use a lot of bad words. It’s my family’s bad habits, especially my brother’s. We don’t know how to talk without saying shit around here.
...	...
Example 10	Sorry I messed up big time is perhaps the phrase I heard most often with my ex. And I alway reply him Bitch pleaz

To obtain a concise style description, we prompt ChatGPT to extract seven adjectives

from the character profile (Listing A.1).

The resulting descriptors and explanations appear in Listing A.2:

The explanations are inspected when reviewing the extracted descriptors. Only the resulting adjectives are subsequently used to condition commentary generation.

Observation data

An observation describes the location of an NPC and the entities visible from that location.

A writer supplies the following seed observation: "In the central place of the city of Glimdale: 56 of Wood on the ground, A trash can"

Additional observations are generated with ChatGPT using the following prompt:

Each request produces five observations. Previously generated observations can be reused as demonstrations in later requests.

5.2 Factual Text Generation

Factual generation prioritizes fluent descriptions that preserve the input information. Stylistic diversity is secondary to fidelity in this task.

Table 5.3 summarizes practical considerations for the approaches considered in the original project.

Table 5.3: Design considerations for factual generation.

Approach	Main design benefit	Main requirement
Templates	Explicit output structure	Manual coverage and grammatical rules
Hosted models	Access through an API	Service access and usage-dependent costs
Local models	Control over inference and adaptation	Compute resources and model configuration

Templates offer direct control but require manually designed rules and grammatical handling[14]. Hosted APIs provide an alternative at a usage-dependent cost. Here we investigate local generation with FLAN-T5[19] for verbalization and the ASDOT sentence-fusion model[38], drawing on WikiFluent[13].

Figure 5.4 summarizes the four-stage factual-generation pipeline:

1. **Serialization.** Convert each record into the key-value string defined in Chapter 4.
2. **Verbalization.** Generate one factual sentence for each serialized record.
3. **Concatenation.** Join the sentences and prepend the task prefix `summarize:.`
4. **Sentence fusion.** Apply the ASDOT fusion model to combine related sentences and reduce repetition.

We evaluate verbalization on 30,000 records by checking whether every input value appears in the generated sentence. Complete value coverage is observed for 94.5% of records.

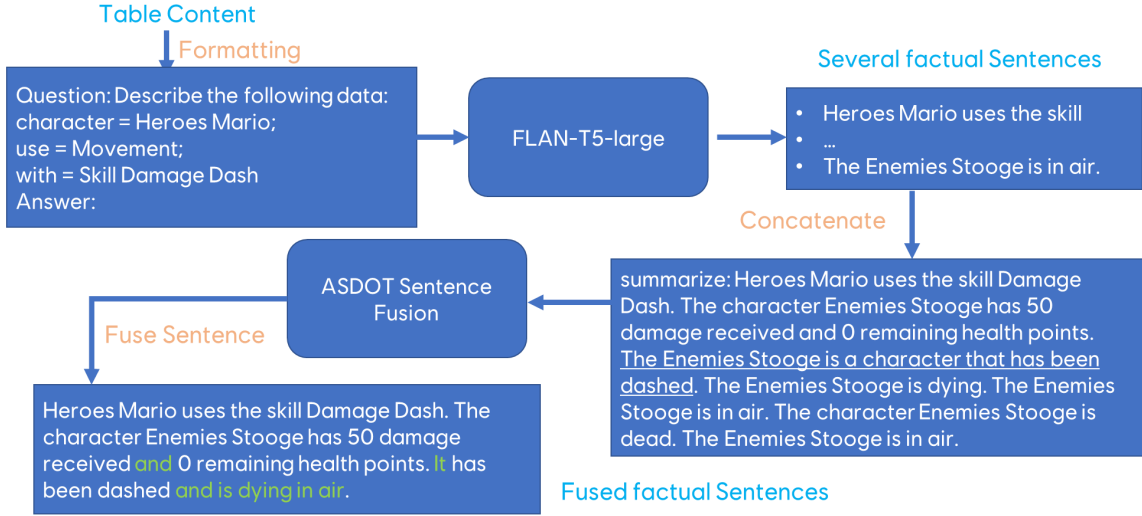


Figure 5.4: Factual generation through serialization, verbalization, concatenation, and sentence fusion.

5.3 Event-Conditioned Commentary

Commentary generation is guided by three application objectives:

1. **Diversity.** Produce varied expressions for the events being described.
2. **Controllability.** Condition the output on a tone specified by the game designer.
3. **Character consistency.** Generate plausible utterances that reflect the intended NPC profile.

Table 5.4: Design considerations for commentary generation.

Approach	Main design benefit	Main requirement
Templates	Explicit output structure	Manual coverage and grammatical rules
Hosted models	Access through an API	Service access and usage-dependent costs
Local models	Control over inference and adaptation	Compute resources and model configuration

5.3.1 Data Preprocessing

For commentary, related records are aggregated before generation, rather than being verbalized separately and fused afterward.

Records sharing an action identifier are merged into the representation shown in Table 5.5.

The actor name and event type identify the action; the detail field combines its type and description. Associated status changes and damage records populate the result field. The study selects actions with recorded consequences for commentary generation.

Table 5.5: Aggregated representation of the event in Table 5.1.

Key	Context	Value
Name	the name of the character	Rabbid Mario
Type	the type of the event	Action
Detail	the detail of the event	Attack with Skill Counter Fist
Result	the result of the event	Stooge Alpha, Status, Burning, Status, Dead ; Stooge Beta, Status, Burning

The following exploratory comparisons use the game-event example in Table 5.6. They examine hosted APIs and locally run instruction-following models; Table 5.4 summarizes the practical considerations.

Table 5.6: Input event used for the qualitative model comparison.

Key	Value
name	Hero Bowser
type	Action
detail	Attack with Skill Bazooka
result	"Enemy Goomba Gamma, Status, Dead; Enemy Goomba Delta, Status, Dead"

5.3.2 General Prompt

Listing A gives the prompt used for the API and local-model comparisons.

Braced placeholders are replaced with event values. The prompt combines a task instruction, explanations of the event fields, and the serialized event.

5.3.3 Hosted Instruction-Following Models

The API examples below use the same event to illustrate differences in generated commentary.

Instruction-following variants are selected where available.

For text-davinci-003 and Jurassic-2[30], we use the prompt in Listing A. Lamini[1] represents the input and output schemas using the Python classes reproduced in Listing A.3.

Table 5.7: Hosted-model outputs for the event in Table 5.6.

Model	Response
lamini	fact: Hero Bowser used the skill Bazooka to defeat two enemies, Goomba Gamma and Goomba Delta. title: Hero Bowsers Heroic Victory! content: Wow, Hero Bowser just defeated two enemies with one attack! His skillful use of the Bazooka was truly impressive! We can all learn from his example and strive to become better warriors.
J2-instruct-Grande	Hero Bowser sure knows how to get the job done! With his trusty Skill Bazooka, he was able to take down not one, but two Goombas! Talk about making a statement!
text-davinci-003	Bowser always gets the job done! Goombas Gamma and Delta never stood a chance against his skillful bazooka!

All three API outputs describe the example event in a commentary-like form. This illustrative comparison does not establish their performance on a representative test set.

5.3.4 Local Instruction-Following Models

Table 5.3.4 presents local-model outputs for the same input and the CPU latencies recorded in the original experiment.

Table 5.8: Local-model outputs and recorded CPU generation times for the same event.

Model	Response	CPU time (s)
FLAN-T5-large(0.8b)[19]	Bowser is a hero	1.79
FLAN-T5-xl(3b)[19]	Bowser killed two Goombas with his bazooka.	5.81
FLAN-T5-xxl(11b)[19]	Hero Bowser used his bazooka to kill two Goombas.	20.78
Dolly-v2-7b[9]	So, that brave hero, Bowser, went on a very daring and risky attack against the village, armed with a Skill Bazooka. Despite all the danger and risk, the attack worked and the Hero destroyed both enemies. Of course, it wasn't all smooth sailing, with Hero going down with enemy statuses. But, well done Bowser!	57.48
Falcon-7b-instruct[33]	Hero Bowser's got some serious skill with that Bazooka, huh?	266.32
Llama-2-7b-chat-hf[34]	Looks like Hero Bowser just bazooka-ed those Goombas into oblivion. I guess when you're the king of the Koopas you get to call the shots literally!	35.30

Model-size annotations denote billions of parameters; CPU times are reported in seconds.

In this example, FLAN-T5-xl and FLAN-T5-xxl produce factual descriptions rather than witty commentary. Some larger models produce commentary, but the Dolly output also introduces an unsupported status change.

5.4 Tone-controlled Comment Generation

Initial examples tended toward enthusiastic commentary. To request other tones explicitly, the prompts include definitions and demonstrations.

A Ubisoft writer selects ten target tones: worried, joyful, sardonic, naive, violent, wise, optimistic, facetious, resentful, and flirty.

The writer supplies definitions and demonstrations for violent, flirty, and sardonic tones. Listing A.4 illustrates the violent-tone prompt.

ChatGPT drafts definitions and examples for the remaining tones using the three writer-provided examples. The content advisor reviews the drafts. Listing A.5 shows the request.

Table 5.9 reports examples generated by text-davinci-003 under the tone-conditioned prompts.

Five examples per tone are inspected. Tone correctness is the number judged consistent with the requested tone out of those five examples.

The assessment identifies both successful examples and tone-control failures. One rejected facetious output reads: “Wow, Hero Rabbid Peach sure knows how to make a statement with Tri Shooter - they just took out their enemy in one shot! Talk about making an impact! The reviewer considered this output insufficiently facetious despite its fluency.

Table 5.9: Tone-conditioned outputs and reviewer acceptance counts out of five examples per tone.

Tone	Tone correctness	Good example
facetious	3/5	Looks like Hero Mario got an early start on handing out presents to the enemy. Their holiday spirit has been duly rewarded!
flirty	5/5	Your movements are as graceful as they are powerful, Stoooge Dashmaster Iota! I'm impressed!
joyful	5/5	Wow! Mario just got hit hard by Enemy Sniperino Kappa's Sharp Shooter! That was an impressive attack! Looks like the battle is heating up!
naive	5/5	Wow! That was so cool! I can't believe I took down all those enemies with my Skill Gauntlets. I'm so powerful!
optimistic	5/5	Hero Rabbid Rosalina is unstoppable! She's dealt some serious damage to our enemies, and we're sure to come out victorious with her on our side!
resentful	5/5	How could our heroes be so easily disarmed?! They should have been more prepared for the enemy's attack!
sardonic	4/5	Wow, you think you can take me on with that tiny spark you call a Super Attack? Good luck!
violent	2/5	I'll smash your heads together with my Counter Fist and send you both to the grave!
wise	4/5	It looks like we have a powerful hero on our side! With this Skill Immunity Charm, we can be sure that our hero is well-protected from any attacks and can fight with confidence.
worried	5/5	Oh no, Mario took a huge hit! We have to find a way to counterattack quickly or this could be the end of us.

5.5 Character-Conditioned Commentary

NPC commentary should be consistent with the speaker profile. Character conditioning therefore includes identity, dialogue style, and conversational habits, in addition to the observed scene.

This experiment uses the character profiles described in Section 5.1.2.

5.5.1 Dataset Construction Pipeline

The workflow in Figure 5.5 constructs training data for a compact model, with the aim of limiting repeated API calls during subsequent use.

The workflow starts with a small set of API-generated seed comments. Human reviewers assess these seeds, which then serve as demonstrations for few-shot expansion with Llama 2. Model-based criteria filter the expanded outputs. The subsequent sections describe each stage.

Human review is concentrated on the seed set. The reviewed seeds then support local few-shot expansion and criterion-based filtering, yielding the synthetic commentary corpus used for character-specific adaptation.

Llama 2[34] expands the reviewed seeds. Writer-supplied criteria guide automated filtering, reducing the number of outputs that require individual manual review; the retained outputs form the adaptation corpus.

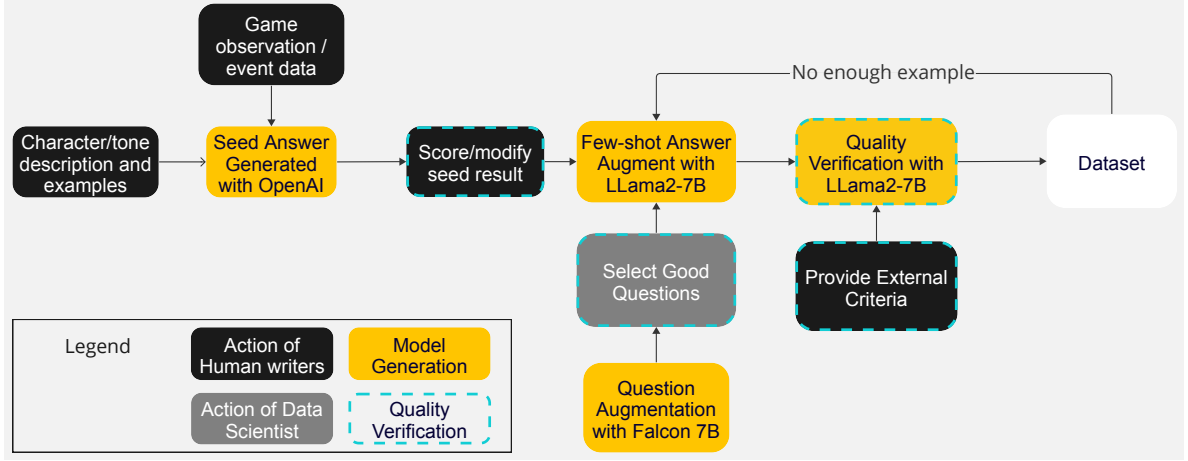


Figure 5.5: Synthetic commentary construction through seed generation, review, expansion, and filtering.

5.5.2 Seed Comment Generation

Writers provide the tone definitions and character descriptions used to generate seed comments with text-davinci-003 on the character-scene inputs.

Listing A specifies the seed-generation prompt.

The instruction is instantiated from the nested template in Listing A.

Human writers classify the reviewed seed outputs as acceptable (139), partially acceptable (49), or unacceptable (160), as reported in Table 5.10.

We retain the 139 accepted outputs and the 49 partially accepted outputs after writer revision, yielding 188 seed comments. Table 5.10 reports the review outcomes for each character.

Table 5.10: Human review outcomes for seed comments, before revision of partially accepted outputs.

	Margie	Karima	Birdie	Liza	James	Noah	Salma	Total
unacceptable	19	27	13	22	29	22	28	160
partially acceptable	9	4	3	11	10	5	7	49
acceptable	21	19	33	17	11	23	15	139

5.5.3 Few-shot Generation with Llama 2

The reviewed seeds provide two-shot demonstrations for Llama-2-7b-chat-hf[34]. The nested prompt contains a system message, character context, two demonstrations, and the target observation. Listing A.6 preserves the system-prompt representation in the original source.

The full few-shot prompt appears in Listing A.7.

Listing A.8 preserves the exemplar-template record supplied with the original experiment.

The generation procedure uses seven characters, eleven instruction variants, and 188 reviewed seeds. The observation split contains 145 training locations, 22 validation locations, and 22 test locations. The original report states that locations and visible-entity sets do not overlap across splits. It records 25,000 training sampling attempts and 1,000 attempts each for validation and testing, before filtering and cleaning.

5.5.4 Model-based Quality Verification

Llama 2 evaluates generated outputs along four dimensions:

1. Congruence to the speaker’s characteristics
2. Fidelity to the context
3. Semantic correctness
4. Human-like credibility

For the first dimension, we provide the contextual information of the character shown in Listing A.9 and ask Llama 2 to evaluate the quality.

For the other dimensions, the writer supplies the following speaker-independent evaluation criteria:

Criteria about the meaning and semantics:

1. All the words in the comment make sense.
2. The slang used in the comment is understandable thanks to an explicit explanation or a contextualization within the comment.
3. The comment looks like something that could be said, and does not use abbreviation.

Criteria about the credibility as a human imitation:

1. The comment feels like something someone would say.
2. The comment is true to the personality of the character without using obvious clichés.
3. The comment is respectful and takes into consideration the required civility concerning vulnerable people.

Criteria about the context and elements provided:

1. The comment does not feature an exhaustive list of the elements provided.
2. The comment uses the elements provided to assess the surrounding of the character.
3. The comment relates to the character’s surrounding like a human would.

The evaluation prompts share a binary-decision format and request a brief justification. Listing A.10 gives the semantic-quality prompt.

With the initial sample of 1000, we estimate the passing rate of 74.8%, 97.6%, 84.1% and 89.8% for characteristics congruence, context fidelity, semantic correctness, and human-like credibility respectively. The overall acceptance rate on this initial sample is 64.5%.

The checks are ordered to evaluate higher-rejection criteria first. Processing stops at the first failed check to avoid unnecessary subsequent evaluations.

5.5.5 Corpus Cleaning

After model-based filtering, we remove character-name prefixes and stage directions such as **giggles**. We discard comments shorter than 10 words or longer than 100 words, collapse consecutive spaces, and trim leading and trailing whitespace. The retained comments form the synthetic commentary corpus.

Random character sampling, different acceptance rates, and cleaning yield unequal character counts. Table 5.11 reports the retained samples.

Table 5.11: Retained synthetic commentary examples by character and split

	train	eval	test
Karima	1596	68	61
Margie	1487	90	56
Birdie	182	5	9
Liza	865	55	35
James	1394	61	49
Noah	2113	97	49
Salma	1244	50	45
Total	8881	426	304

5.6 Chapter Summary

The framework connects structured event descriptions with controlled commentary through explicit input representations and conditioning information. Human-reviewed seeds provide examples for synthetic expansion, while model-based filtering and cleaning produce the character-specific adaptation corpus. The following chapter evaluates the resulting models.

Chapter 6

Experiments and Evaluation

This chapter evaluates character-specific adaptation using the synthetic commentary corpus. We describe the training configuration and preliminary runs, define the evaluation metrics, and examine learning curves, test scores, and generated examples. The discussion then relates these observations to the research questions and identifies limitations.

6.1 Experimental Setup

We use `pszemraj/flan-t5-large-instruct-dolly_hhrlhf`, a FLAN-T5-large-based checkpoint further instruction-tuned on Dolly and a filtered subset of Anthropic HH-RLHF data[9, 3]. The motivation is to study an instruction-following model with fewer than one billion base parameters.

Training uses one NVIDIA RTX 3090, AdamW, a learning rate of 3×10^{-5} , batch size 16, and 200 epochs. LoRA uses rank 32, scaling factor 32, and dropout 0.05. Approximately 1.1% of parameters are reported as trainable. A separate adapter is trained for each character using the corresponding subset, and the checkpoint with the highest validation ROUGE-L is selected for test evaluation.

6.2 Preliminary Adaptation Experiments

We examine two preliminary configurations: full-model fine-tuning on examples from all characters, and LoRA adaptation on the Karima subset. Figures 6.1 and 6.2 show training loss and validation MAUVE for these runs.

The plots retain the original training logs. The displayed loss curves use a smoothing factor of 0.6; legend entries record the final logged values, steps, and run timestamps.

The original runs report approximately 15 minutes for the character-specific LoRA experiment and approximately three hours for full-model training over all characters. These preliminary runs motivate the subsequent character-specific adaptation experiments.

6.3 Evaluation Metrics

We evaluate generated commentary against the retained synthetic test references using three complementary metrics:

- **MAUVE** compares distributions of generated and reference text through divergence frontiers[26].
- **ROUGE-L** measures sequence overlap based on the longest common subsequence[16].

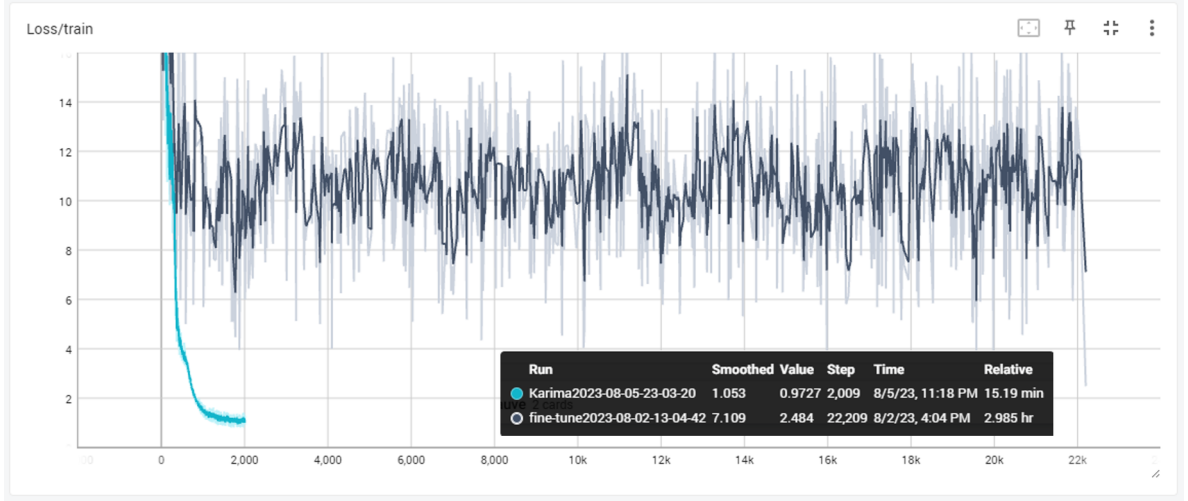


Figure 6.1: Training-loss trajectories for the preliminary full-model and character-specific LoRA runs.

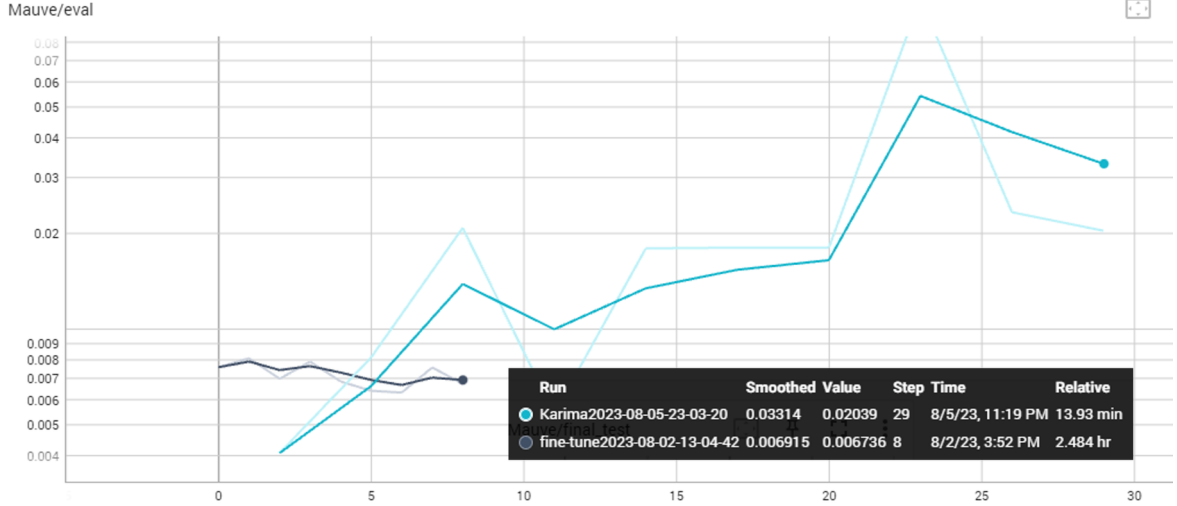


Figure 6.2: Validation MAUVE trajectories for the preliminary adaptation runs.

- **BLEU-4** measures modified n-gram precision up to order four with a brevity penalty[23].

Qualitative analysis complements these metrics by examining character-specific expressions and the relationship between generated comments and scene inputs.

6.4 Learning Dynamics

We first examine whether the evaluation metrics follow the same trajectory during adaptation. This analysis connects optimization behavior to the checkpoint-selection procedure described above. Figure 6.3 reports the training-loss trajectories. The plots retain the original training logs. The displayed loss curves use a smoothing factor of 0.6; legend entries record the final logged values, steps, and run timestamps.

Figures 6.4, 6.5, and 6.6 show validation trajectories. MAUVE initially increases and subsequently decreases, whereas ROUGE-L and BLEU-4 remain comparatively stable after their initial rise.

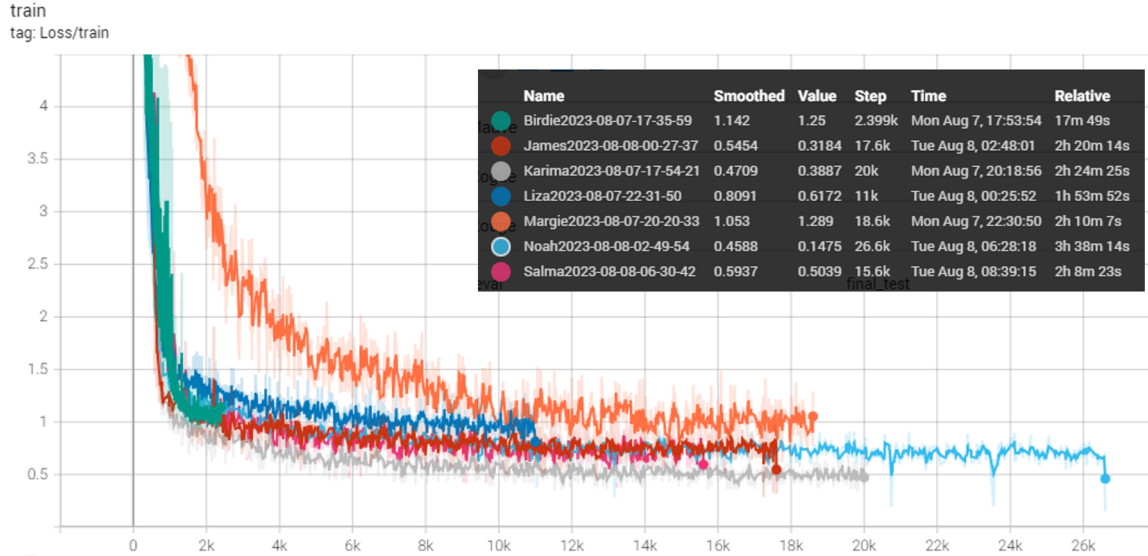


Figure 6.3: Training loss by optimization step for character-specific adapters.

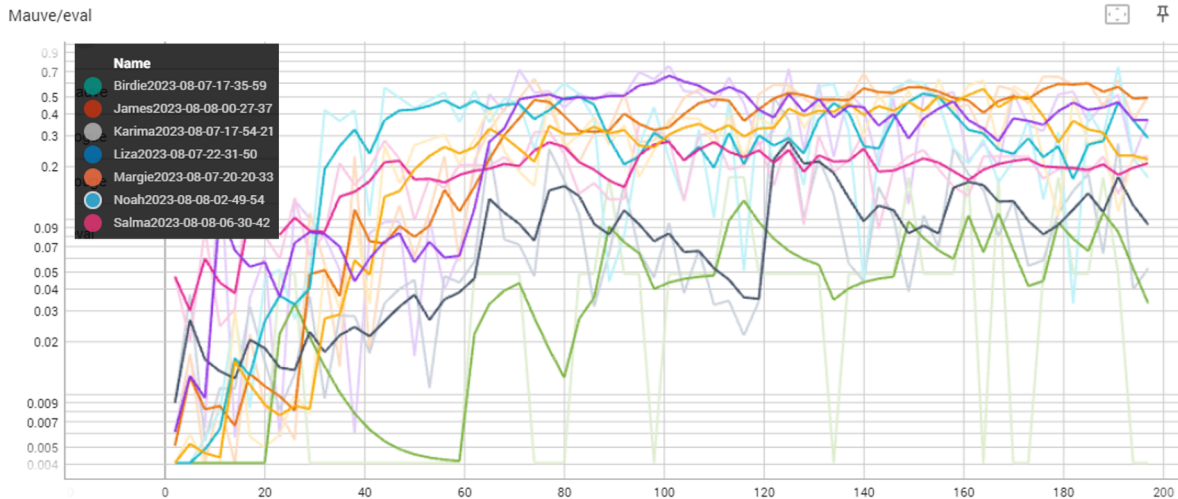


Figure 6.4: Validation MAUVE by training epoch.

6.5 Held-Out Commentary Results

The held-out evaluation asks how closely each character-specific model reproduces the filtered reference distribution. Table 6.1 reports the three metrics alongside the number of test examples for each character. The sample counts make the evaluation population explicit and connect the scores to the corpus statistics in Chapter 5.

The metrics yield different orderings across character subsets. Salma has the highest observed MAUVE score, 0.537, whereas Noah has the highest ROUGE-L and BLEU-4 scores, 0.403 and 0.220. Noah’s MAUVE score is 0.258, illustrating that high lexical overlap can coexist with a lower distributional-similarity score in the recorded evaluation. This contrast motivates examining the generated language directly rather than describing performance through a single metric.

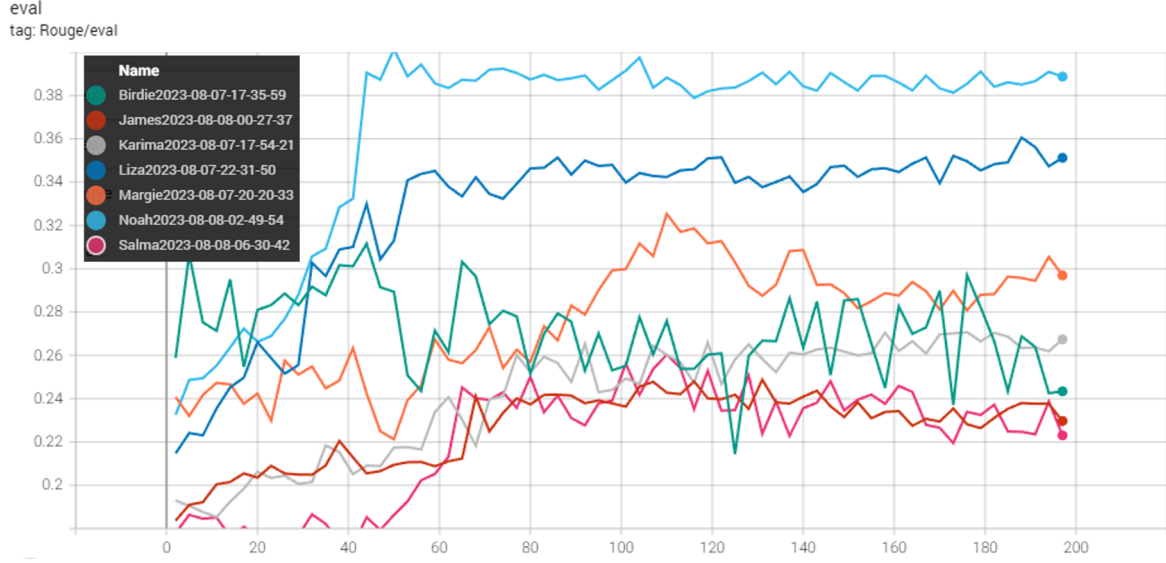


Figure 6.5: Validation ROUGE-L by training epoch.

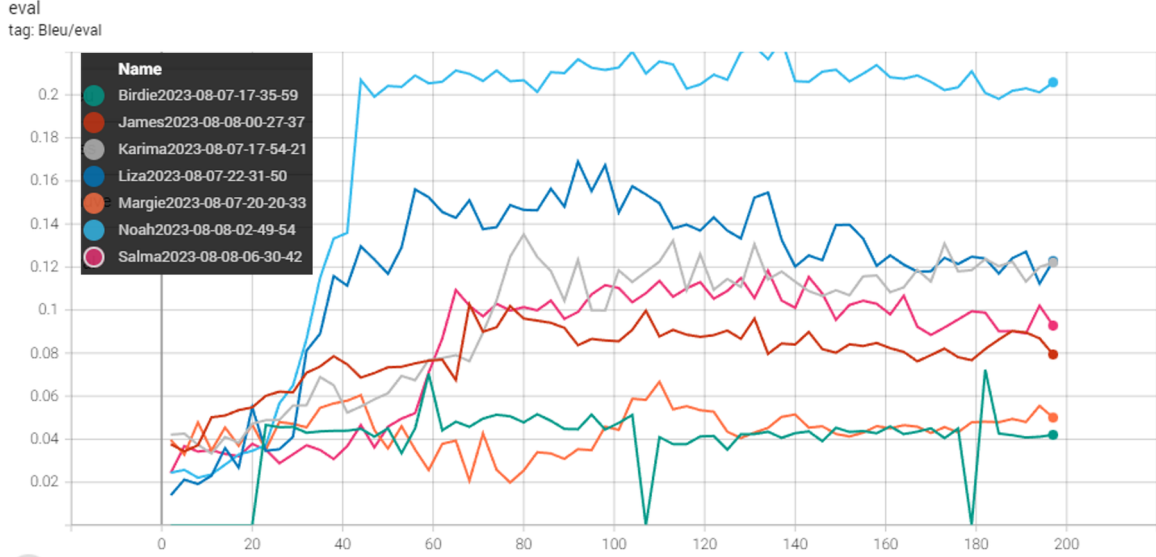


Figure 6.6: Validation BLEU-4 by training epoch.

6.6 Qualitative Analysis

To examine how character conditioning appears in individual utterances, we hold the scene description fixed and inspect outputs for Noah, Margie, and James. Table 6.2 presents short excerpts with the feature each excerpt illustrates. The complete few-shot Llama 2 and adapted FLAN-T5 outputs are provided in Appendix B.

The excerpts distinguish style expression from contextual content. Noah’s informal address and James’s aesthetic appraisal instantiate different speaking patterns for the same scene. Margie’s output contains both a recognizable stylistic marker and an added intention. Reading these features separately provides a more precise account of the examples than assigning an overall impression of similarity. Appendix B preserves both models’ complete outputs so that the excerpts can be interpreted in context.

Table 6.1: Character-specific evaluation on held-out synthetic references. n is the number of test examples. Scores are reported on their original scale; trailing zeros standardize presentation without adding measurement precision.

Character	n	MAUVE	ROUGE-L	BLEU-4
Birdie	9	0.518	0.292	0.050
James	49	0.275	0.260	0.090
Karima	61	0.450	0.256	0.120
Liza	35	0.457	0.318	0.080
Margie	56	0.457	0.284	0.040
Noah	49	0.258	0.403	0.220
Salma	45	0.537	0.222	0.080

Table 6.2: Selected excerpts from adapted FLAN-T5 outputs for a shared floating-island scene. Excerpts are taken from the full outputs in Appendix B; the final column describes an observable feature rather than a quality score.

Character	Output excerpt	Observed feature
Noah	“Dude, this place is straight out of a sci-fi movie!”	Informal address and an emphatic comparison.
Margie	“Oorah, that’s a lot of rocks.”	A military-associated interjection.
Margie	“Let’s see if we can find a way to move them around.”	An intention not specified by the observation.
James	“The Sky Garden with rare plants is a feast for the eyes”	An aesthetic appraisal of a visible entity.

6.7 Discussion

The findings address the three research questions at different levels. First, the factual-generation pipeline retains all input values in 94.5% of the evaluated records, demonstrating coverage under the specified matching criterion. Second, the tone examples and character-conditioned outputs show that explicit conditioning affects how events and observations are expressed. Third, character-specific adaptation produces distinct lexical and stylistic patterns, with automatic scores varying across character subsets.

These observations connect the two components of the thesis: structured verbalization provides an explicit representation of event content, while commentary conditioning determines how that content is expressed. The quantitative metrics and qualitative examples offer complementary views of the resulting outputs. The following limitations define the scope within which these findings should be interpreted.

6.8 Limitations and Reproducibility

The test scores measure agreement with filtered synthetic references. BLEU and ROUGE-L emphasize lexical overlap; MAUVE compares text distributions and is not a direct measure of factuality or character consistency. No uncertainty intervals, repeated-seed results, or independent human preference study are available. Birdie’s test subset contains only nine examples, so its results require particular caution. Scores across characters also reflect different reference distributions and sample sizes.

The CPU latencies reported in Chapter 5 concern illustrative generations. The CPU model, decoding lengths, repeated measurements, and runtime configuration are not specified. Consequently, the original claim of a tenfold inference improvement is not treated as a validated benchmark. A reproducible efficiency comparison should hold input data, output length, decoding settings, hardware, and measurement procedure constant.

The model used to expand the dataset also evaluates the expanded examples. This creates a risk of correlated generation and evaluation biases. Independent human evaluation should measure factual fidelity, style consistency, fluency, and appropriateness using a documented rubric. The current results support an exploratory feasibility study rather than deployment readiness or a state-of-the-art claim.

6.8.1 Data Integrity and Reproducibility

The final split contains 8,881 training, 426 validation, and 304 test examples. The per-character counts sum to these totals. These examples are filtered synthetic outputs, not independently authored human references. Splitting locations before generation reduces direct overlap, but does not establish the absence of semantic overlap or contamination in pretrained models. The supplied materials do not include the underlying split files, generation logs, or filtering-stage counts needed for an independent audit.

The prompt listings and generated examples above are retained as historical experimental materials. Their wording, including informal language and transcription errors, is not an instruction to the reader or an endorsement of the generated content. In particular, the listing labeled as an exemplar template duplicates the enclosing few-shot template in the supplied source; the actual implementation should be recovered before reproducing the experiment.

6.8.2 Experimental Scope

The full-model and LoRA preliminary runs use different character coverage and dataset sizes. Their reported training times cannot isolate the effect of the adaptation method. The tone

assessment uses only five examples per tone and lacks a documented independent annotation protocol. The seed table accounts for 348 reviewed outputs, whereas the nominal count of 50 per character implies 350; the supplied records do not resolve this discrepancy. Filtering-stage counts are also unavailable. Random seeds, LoRA target modules, decoding settings, software versions, and exact checkpoint revisions are needed for reproducibility. These gaps motivate the controlled evaluations and artifact recovery described in Chapter 7.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis examined how structured game data can support two forms of language generation: descriptions that verbalize recorded events and commentary that reflects a specified tone or character. The work connects these tasks through explicit representations of event content and speaker context, followed by generation and character-specific adaptation. The findings answer the research questions introduced in Chapter 1 at the level of the implemented pipelines and the reported evaluations.

For factual verbalization, the study implemented a modular sequence of serialization, row-level generation, and sentence fusion. In the evaluation of 30,000 records, 94.5% of generated sentences retained every input value under the specified matching criterion. This result answers the first research question in terms of input-value coverage and provides a concrete measure of the verbalization stage’s behavior.

For controlled commentary, the study represented tone through definitions and demonstrations and represented characters through profiles and example utterances. The tone assessment and shared-scene examples documented how these conditions were reflected in generated language. Informal address, military-associated expressions, and aesthetic appraisals appeared in the outputs for different characters. These observations address the second research question by showing the forms of stylistic variation obtained in the studied examples.

For compact-model adaptation, the study combined human review of seed comments with few-shot generation, model-based filtering, and cleaning. The resulting corpus supplied 8,881 training examples across seven characters, with separate validation and test sets. Character-specific LoRA adapters were then trained for a FLAN-T5-based model and evaluated using automatic metrics and qualitative examples. The resulting outputs contained recognizable character-associated language. The differing MAUVE, ROUGE-L, and BLEU-4 rankings also showed why the recorded performance is better described through several complementary measures than through a single score.

The contribution of the thesis is the connection between these components: a structured input representation, a factual-generation pipeline, an explicit conditioning strategy, and a procedure for constructing data for character-specific adaptation. Together, they provide an experimental basis for studying game-oriented language generation from both the content and speaker perspectives. The scope of the evaluation and its reproducibility requirements are discussed in Chapter 6.

7.2 Future Work

A first priority is independent evaluation of the generated commentary. A human study should assess factual consistency, character consistency, fluency, and appropriateness separately, using a documented rubric and multiple annotators. Comparisons on the same observations would help determine when stylistic expressions contribute to a convincing character voice and when they introduce unsupported content.

A second direction is controlled evaluation of adaptation and efficiency. Experiments should hold the training data and evaluation procedure constant while comparing full-model fine-tuning, a shared adapter, and character-specific adapters. Reporting hardware, decoding settings, output lengths, repeated-run timings, and variation across random seeds would make the comparison interpretable. Such experiments could also examine how adaptation rank and the number of characters affect quality and resource use.

A third priority is the data-construction process. Recovering the generation configurations and filtering logs would support an audit of split independence, resolve the seed-count discrepancy, and identify where examples are discarded. Independent review of filtered outputs would help assess the model-based quality criteria. Releasing the permitted prompts, configurations, and data artifacts would further support reproducibility.

Finally, evaluation within an interactive game would extend the present analysis from isolated outputs to situated use. This would allow latency, repetition across successive events, consistency over time, and player responses to be studied together. The framework developed in this thesis provides the components for that next stage of investigation.

Appendix A

Prompt Templates and Experimental Listings

This appendix records the prompts and schema examples used in the generation workflow. Listings retain their original experimental wording. The chapter-level descriptions explain their role in the pipeline.

Listing A.1: Question for summarizing adjectives

```
Could you use 7 adjective words to express the Liza's way of speaking? (please list them in bullet points, demonstrate them based on context provided below)

{Identity} + {Dialogue style} + {Habits} + {Colloquialism}  \\\
```

Listing A.2: Response for summarizing adjectives

```
Optimistic: Liza's way of speaking is filled with positivity and hope. She always sees the bright side of things and radiates optimism in her words.
Cheerful: Her language is bubbling with joy and happiness, reflecting her cheerful and lively personality.
Sociable: Liza's friendly and informal language reflects her sociability. She loves interacting with people and easily makes friends.
Enthusiastic: Liza's boundless enthusiasm shines through in her rapid speech and frequent use of capital letters and exclamation marks.
Curious: Her inquisitive nature is evident in the way she asks lots of questions, always eager to learn and discover new things.
Unconventional: Liza's language is informal and unconventional, with contractions and Gen Z slang, making her speech unique and contemporary.
Fearless: She freely expresses her opinions and isn't afraid to call out retrograde or sexist remarks. Her fearlessness extends to discussing sex and queer slang openly
```

```
Imagine you are a NPC on in a game. You went around the worlds and saw many different things and people.
You could see several game assets which you could interact with.
Could you give me some exemplary scenes that you could see in a game?
For example:
In the central place of the city of Glimdale: 56 of Wood on the ground, A trash can
Try to generate 5 examples which contains different assets:
```

```
Could you make a witty comment based on the event?
```

```
The event will contain the following information:
```

```
name is the name of the character; type is the type of the event, action; detail is
the detail of the event; result is the result of the event, like some character
got a new status or received damage.
```

Please a witty comment based on the following event:

```
name: {name}. type: {type}. detail: {detail}. result: {result}.
```

Listing A.3: Lamini prompt/ response definition

```
class Event(Type):
    name: str = Context("the name of character in the game")
    type: str = Context("type of event")
    detail: str = Context("details of the event")

class Comment(Type):
    title: str = Context("the title of the comment")
    fact: str = Context("the fact of event")
    content: str = Context("the content of the comment")
```

Listing A.4: violent tone prompt example

```
A violent comment should:
- be made in a serious and dark tone.
- express a desire to hurt the enemy
- it can emphasize a particular way to wound the other (ex: to target a body part,
  to specify the weapon used)
- feel threatening
```

Examples:

- I will crush your skull with my hammer !
- I should teach you respect with the edge of my sword

Based on the event, are there any remarks you can make that are violent?

```
name: {name}. type: {type}. detail: {detail}. result: {result}
```

Listing A.5: Tone definition question for ChatGPT

Given the definitions and examples for violent, flirty and sardonic tones as follows, could you provide simillar definitions and examples for worried, joyful, naive, wise, optimistic, facetious and resentful tones as well?.

```
{description}
{dialoguestyle}
For example:
{example}

{instruction}
location(location of the character): {location}; visible entities(entities that are
visible to the character): {visible_entities}

{character}'s comment:
```

In {character}'s {adjective} tongue, how would he/she talk about what he/she sees?

Listing A.6: system instruction

```
<<SYS>>
You are a helpful, respectful and honest assistant. Always answer as helpfully as
possible, while being safe. Your answers should not include any harmful,
unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure
that your responses are socially unbiased and positive in nature. If a question
does not make any sense, or is not factually coherent, explain why instead of
answering something not correct. If you don't know the answer to a question,
please don't share false information.
<</SYS>>

[INST]
{few-shot prompt}
[/INST]
```

Listing A.7: few-shot prompt

```
{description}
{dialoguestyle}
We will make comment based on the character, for example:
{exemplar prompt 1}
{exemplar prompt 2}
{instruction}
location(location of the character): {location}; visible entities(entities that are
visible to the character): {visible_entities}
Please just provide the comment without any reasoning or explanation.
{character}'s comment:
```

Listing A.8: exemplar prompt

```
{description}
{dialoguestyle}
We will make comment based on the character, for example:
{exemplar prompt 1}
{exemplar prompt 2}
{instruction}
location(location of the character): {location}; visible entities(entities that are
visible to the character): {visible_entities}
Please just provide the comment without any reasoning or explanation.
{character}'s comment:
```

Listing A.9: Prompt for congruence to the speaker's characteristics

```
{{description}}

{dialoguestyle}

We have a comment like this: {comment}
Based on the profile of {character}, could you tell whether this comment is from {
character}?
Please give reasons with demonstration from the comment.
Tell me your answer (Yes or No) first and give me your reason in one sentence.
```

Listing A.10: Prompt for semantic correctness

```
Now we need to know whether {character}'s comment is of good quality in respect of
meaning and semantic?
```

A good comment should be

- All the words in the comment make sense.
- The slang used in the comment is understandable thanks to an explicit explanation or a contextualization within the comment.
- The comment looks like something that could be said, and does not use abbreviation.

Could you tell me whether {character}'s comment is of good quality in respect of meaning and semantic?

Comment: {comment}

Please answer yes or no first and give me your reason with demonstration from the comment in one sentence.

Start with "Yes" or "No".

Appendix B

Complete Qualitative Examples

The full outputs below accompany the excerpt-based analysis in Chapter 6. The original wording is preserved.

Table B.1: Few-shot Llama 2 and adapted FLAN-T5 outputs for one scene and three character profiles.

Location	On a floating island in the sky
Visible_entities	Levitating rocks forming a landmass, floating waterfalls, a community of aerial dwellings, a sky garden with rare plants, winged creatures soaring overhead.
character	Noah(generous, sociable, simple, emotional, relaxed, party-goer , in love)
Few-shot llama2	Dude , check out that levitating landmass! It's gnarly how those rocks are just chillin' up there. And look at those floating waterfalls, man! They're so cool! I wanna check out that community of aerial dwellings, they look like they'd be fun to explore. And that sky garden with those rare plants? Forget about it, bro! I wanna see 'em up close. And oh, look at those winged creatures soaring overhead! They're straight outta a fantasy movie, man!
Fine-tuned FLAN-T5	Dude , this place is straight out of a sci-fi movie! Check out those rocks , man! They're like, totally floating in the sky. And those waterfalls, they're like nature's own personal water park. And that community of aerial dwellers, it's like a whole other world up there . And that sky garden with the rare plants, it looks like it's straight outta a fairy tale. And those winged creatures, they've got some serious attitude, bro! This place is off the chain, bro!
character	Margie(authoritative, emotionally reserved, efficient, military acronyms, military slang and metaphors, purposeful communication, brief language)
Few-shot llama2	Situation normal . Floating rocks forming landmass. Waterfalls levitating. Community of dwellings up here. Sky garden got some rare gems. Creatures soaring overhead. Oorah .
Fine-tuned FLAN-T5	Oorah , that's a lot of rocks. Let's see if we can find a way to move them around. We've got some floating waterfalls, and a community of air dwellings, and we've got a sky garden.
Character	James(ambitious, precise, formal , competitive, aesthete , witty, gossipy)
Few-shot llama2	Ah, look at this marvelous floating island! The levitating rocks are quite impressive, aren't they? I do love the way the waterfalls float along with the island. And the community of aerial dwellings is simply delightful. I must say, the sky garden with its rare plants is a true masterpiece . And those winged creatures overhead are just the bee's knees! I do believe I'm in awe of this wondrous place .
Fine-tuned FLAN-T5	Oh, wow! Look at that magnificent floating island! It's like something straight out of a sci-fi movie! The Levitating Rocks forming the landmass are simply breathtaking! The Floating Waterfalls are a marvel of engineering , and the Community of Aerial Dwellings is a testament to the wonders of nature. The Sky Garden with rare plants is a feast for the eyes , and the Winged Creatures soaring overhead are a sight to behold. It's a truly breathtaking sight!

Bibliography

- [1] Introducing Lamini, the LLM Engine for Rapid Customization — lamini.ai. <https://www.lamini.ai/blog/introducing-lamini>. [Accessed 06-08-2023].
- [2] Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. Cont: Contrastive neural text generation. *Advances in Neural Information Processing Systems*, 35:2197–2210, 2022.
- [3] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [5] Wenhua Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. Logical natural language generation from open-domain tables. *arXiv preprint arXiv:2004.10404*, 2020.
- [6] Zhiyu Chen, Wenhua Chen, Hanwen Zha, Xiyu Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. Logic2text: High-fidelity natural language generation from logical forms. *arXiv preprint arXiv:2004.14579*, 2020.
- [7] Jordan Clive, Kris Cao, and Marek Rei. Control prefixes for text generation. *CoRR*, abs/2110.08329, 2021.
- [8] Emilie Colin, Claire Gardent, Yassine M’rabet, Shashi Narayan, and Laura Perez-Beltrachini. The webnlg challenge: Generating text from dbpedia data. In *Proceedings of the 9th international natural language generation conference*, pages 163–167, 2016.
- [9] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world’s first truly open instruction-tuned llm, 2023.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [11] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.

- [12] Shailza Jolly, Zi Xuan Zhang, Andreas Dengel, and Lili Mou. Search and learn: improving semantic coverage for data-to-text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10858–10866, 2022.
- [13] Zdeněk Kasner and Ondřej Dušek. Neural pipeline for zero-shot data-to-text generation. *arXiv preprint arXiv:2203.16279*, 2022.
- [14] Jonathan Lessard and Quinn Kybartas. Blabbeur-an accessible text generation authoring system for unity. In *Interactive Storytelling: 14th International Conference on Interactive Digital Storytelling, ICIDS 2021, Tallinn, Estonia, December 7–10, 2021, Proceedings 14*, pages 255–259. Springer, 2021.
- [15] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.
- [16] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
- [17] Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. Plog: Table-to-logic pretraining for logical table-to-text generation. *arXiv preprint arXiv:2205.12697*, 2022.
- [18] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.
- [19] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.
- [20] Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. Open domain question answering with a unified knowledge interface. *arXiv preprint arXiv:2110.08417*, 2021.
- [21] Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chichun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*, 2020.
- [22] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*, 2017.
- [23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [24] Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*, 2020.
- [25] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.
- [26] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

- [27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.
- [28] Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*, 2020.
- [29] Swarnadeep Saha, Xinyan Velocity Yu, Mohit Bansal, Ramakanth Pasunuru, and Asli Celikyilmaz. Murmur: Modular multi-step reasoning for semi-structured data-to-text generation. *arXiv preprint arXiv:2212.08607*, 2022.
- [30] AI21 Studio. Announcing jurassic-2 and task-specific apis.
- [31] Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Mvp: Multi-task supervised pre-training for natural language generation. *arXiv preprint arXiv:2206.12131*, 2022.
- [32] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [33] Technology Innovation Institute. Falcon-7B-Instruct model card. <https://huggingface.co/tiiuae/falcon-7b-instruct>, 2023.
- [34] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [35] Zihan Wang and Naoki Yoshinaga. Esports data-to-commentary generation on large-scale data-to-text dataset. *arXiv preprint arXiv:2212.10935*, 2022.
- [36] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- [37] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022.
- [38] Jiannan Xiang, Zhengzhong Liu, Yucheng Zhou, Eric P Xing, and Zhiting Hu. As-dot: Any-shot data-to-text generation with pretrained language models. *arXiv preprint arXiv:2210.04325*, 2022.

- [39] Dawei Zhang, Sixing Wu, Yao Guo, and Xiangqun Chen. Moba-e2c: Generating moba game commentaries via capturing highlight events from the meta-data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4545–4556, 2022.