

```

```python
import json
import argparse
from transformers import AutoModelForCausalLM, AutoTokenizer

class TransformerTextGenerator:
 def __init__(self, model_name_or_path):
 self.model = AutoModelForCausalLM.from_pretrained(model_name_or_path,
revision="main")
 self.tokenizer = AutoTokenizer.from_pretrained(model_name_or_path, use_fast=True)

 def generate_text_output(self, prompt, temperature):
 input_ids = self.tokenizer(prompt, return_tensors='pt').input_ids.cuda()
 output = self.model.generate(inputs=input_ids, temperature=temperature,
do_sample=True, top_p=0.95, top_k=40, max_new_tokens=512)
 return self.tokenizer.decode(output[0])

def main():
 parser = argparse.ArgumentParser(description="Generate text outputs using a transformer
model.")
 parser.add_argument("--input_file", required=True, help="Path to the input JSON file
containing input entries.")
 parser.add_argument("--output_file", required=True, help="Path to save the output JSON file
with generated text outputs.")
 parser.add_argument("--creative", action="store_true", help="Flag to enable creative mode
with varied temperature from 0.5 to 1.0.")
 parser.add_argument("--factual", action="store_true", help="Flag to enable factual mode with
varied temperature from 0.5 to 0.0.")
 args = parser.parse_args()

 if args.creative:
 temperature_range = [i / 10 for i in range(5, 11)] # Temperature range from 0.5 to 1.0
 elif args.factual:
 temperature_range = [i / 10 for i in range(5, -1, -1)] # Temperature range from 0.5 to 0.0
with decrement of 0.1
 else:
 print("Please specify either --creative or --factual mode.")
 return

 text_generator =
TransformerTextGenerator("TheBloke/CapybaraHermes-2.5-Mistral-7B-GPTQ")
 text_outputs = []

 with open(args.input_file, "r") as f:

```

```

input_data = json.load(f)

for entry in input_data:
 input_text = entry.get("input", "")
 entry_outputs = []
 for temperature in temperature_range:
 text_output = text_generator.generate_text_output(input_text, temperature)
 entry_outputs.append({"temperature": temperature, "text_output": text_output})
 text_outputs.append({"input": input_text, "outputs": entry_outputs})

with open(args.output_file, "w") as f:
 json.dump(text_outputs, f, indent=4)

if __name__ == "__main__":
 main()
...

```

**\*\*Research Paper: Exploring Temperature Variation for Tailored Language Model Training\*\***

**\*By Troy Andrew Schultz\***

**\*\*Abstract:\*\***

This study delves into the impact of temperature variation on language model training, aiming to develop specialized models optimized for creative expression and factual precision. Through systematic experimentation, I investigate how adjusting temperature settings influences the diversity and accuracy of text generation in language models.

**\*\*Introduction:\*\***

Driven by curiosity and a desire to unlock the potential of language models, I embarked on a quest to uncover the intricate relationship between temperature variation and model performance. By fine-tuning temperature settings, I aimed to develop models capable of meeting diverse linguistic needs, ranging from creative storytelling to factual reporting.

**\*\*Methodology:\*\***

Employing the Hugging Face Transformers library, I trained separate language models tailored for creative expression and factual precision. By adjusting temperature settings within a defined range during training and inference, I sought to optimize model outputs for specific tasks and requirements.

**\*\*Results:\*\***

My experiments demonstrated the effectiveness of temperature variation in shaping the outputs of language models. Enabling creative mode with a temperature range from 0.5 to 1.0 enhanced response diversity, fostering imaginative and contextually rich text generation. Conversely,

factual mode, with a temperature range from 0.5 to 0.0, yielded precise and evidence-based responses, ideal for tasks requiring factual accuracy.

**\*\*Discussion:\*\***

The findings suggest that temperature variation serves as a powerful tool in tailoring language model outputs to specific linguistic tasks. By providing control over temperature settings, I empowered users to fine-tune models for creativity or factual precision, depending on the application at hand.

**\*\*Conclusion:\*\***

This research sheds light on the nuanced relationship between temperature variation and text generation in language models. By embracing experimentation and intuitive exploration, I pave the way for the development of specialized models capable of adapting to diverse linguistic needs and improving generalization across different tasks.

**\*\*Acknowledgments:\*\***

I extend my gratitude to the open-source community and the developers of the Hugging Face Transformers library for their invaluable contributions to natural language processing research.