AI-Powered Social Media Caption Generator Using Retrieval-Augmented Language Modeling and Engagement Prediction

Laiba Muneer[1], Lakshika P. M. M.[1], and Aakash Kuragayala[1]

Department of Information and Communication Technology,
School of Engineering and Technology,
Asian Institute of Technology, Thailand
st125496@ait.asia, st124872@ait.asia, st125050@ait.asia

Abstract. This research introduces an AI-driven caption generation system aimed at automating the production of fashionable, platform-specific social media captions. Utilizing promotional content gathered from news and lifestyle websites via extensive web scraping, we trained a DistilGPT2 model augmented with Retrieval-Augmented Generation (RAG) and Chain-of-Thought (CoT) prompting. Our dataset consists of more than 950 authentic captions organized by category (e.g., travel, events, lifestyle), supplemented with engagement-optimized hashtags. We enhanced caption relevance by integrating semantic similarity retrieval with FAISS and MiniLM-based embeddings. The final system has an interactive chatbot and a Streamlit interface for real-time caption generation, illustrating the practical viability of AI-driven content automation in contemporary social media environments.

1 Introduction

In the era of digital communication, social media has evolved from a medium for informal connection into a formidable ecosystem for marketing, branding, and public participation. Captions are crucial elements; they function as descriptors and persuasive micro-texts that shape content perception, engagement, and sharing. An expertly composed caption may transform a mundane photograph or video into a viral phenomenon, amplify brand identity, and build an emotional bond with the audience.

Despite their brevity, captions are difficult to produce consistently at scale. Content creators, influencers, and digital marketing teams experience significant pressure to produce innovative, fashionable, and audience-specific captions that engage varied demographics. The demand for creativity is intensified by the swiftly changing landscape of social media trends, the necessity for platform-specific formatting, and the casual, culturally nuanced language anticipated by Gen Z consumers. Conventional manual methods of caption creation are not only labor-intensive but also susceptible to stylistic discrepancies and inefficiencies, particularly when handling diverse content categories such as travel, events, products, and lifestyle.

This project presents an AI-driven caption generator that automates the creation of high-quality, Gen Z-oriented social media captions to tackle these difficulties. In contrast to other previous studies that depend on synthetic or crowdsourced datasets, our methodology is based on a web-scraped corpus of authentic promotional captions from reputable online media outlets, including NDTV, Times of India, Vogue India, and Hindustan Times. This guarantees that the model encounters genuine, human-authored captions consistent with current marketing trends and linguistic patterns.

The system architecture incorporates a refined DistilGPT2 language model [3], a streamlined Transformer-based generator geared for succinct and fluent text creation. To improve contextual relevance and generation quality, we utilize a Retrieval-Augmented Generation (RAG) architecture [1] supported by FAISS [5] and MiniLM sentence embeddings [2], enabling the model to learn from semantically analogous samples during inference. Furthermore, we utilize Chain-of-Thought (CoT) prompting [6], a method that organizes the production process into logical sequences to enhance coherence, originality, and tonal consistency.

This project illustrates the viability of automating social media caption generation with contemporary NLP methodologies, while also emphasizing the integration of real-world data, strategic prompt engineering, and modular AI architecture to address a tangible issue in digital marketing. The final implementation comprises a command-line chatbot and a Streamlit web application [9], providing an intuitive interface for the real-time generation of elegant, category-specific, and hashtag-optimized captions.

2 Problem Statement and Motivation

Brands, influencers, and content providers are progressively dependent on concise, fashionable captions to enhance their reach and engagement on platforms such as Instagram and Twitter. However, crafting high-quality captions consistently across multiple content categories is a labor-intensive process. The unpredictability of trends and the diversity of platform-specific tones further compound this issue. Our objective was to eradicate the monotonous, manual labor associated with caption writing by developing a system capable of learning from previous successful instances, comprehending category-specific semantics, and producing contemporary outputs inspired by Gen Z. The training data was obtained by focused web scraping from platforms such as NDTV, Times of India, Hindustan Times, Vogue India, among others, to guarantee practical relevance. This method guaranteed access to contemporary, high-caliber, and specialized promotional language [1,9].

3 Objectives

The principal objective of this project was to create an AI-driven system proficient at generating contextually appropriate, fashionable, and category-specific social media captions. To achieve this, we first aimed to collect a large dataset

of real-world captions by scraping promotional text from reputable online media sources across domains such as lifestyle, travel, events, and technology. This raw data served as the foundation for training and was subsequently cleaned, analyzed, and categorized to ensure quality and structural consistency. The primary aim was to optimize a lightweight yet robust language model—namely DistilGPT2 [3]—on this selected dataset to facilitate fluent and captivating caption production. To improve contextual alignment during inference, we integrated a Retrieval-Augmented Generation (RAG) architecture [1] that employed sentence embeddings [2] and FAISS-based semantic similarity search [5]. In parallel, we implemented Chain-of-Thought (CoT) prompting [6] to structure the model's outputs, ensuring that the generated captions followed a logical and coherent format tailored to Gen Z social media trends. To enhance the system's accessibility and user-friendliness, we created two interactive interfaces: a command-line chatbot and a Streamlit-based web application [9] for real-time caption generation and exploration.

4 System Architecture

The proposed framework is a modular and extendable pipeline that incorporates many advanced natural language processing techniques to facilitate automatic and stylistically consistent social media caption generating. The system comprises five principal components: (1) web scraping and preprocessing, (2) exploratory data analysis and hashtag enhancement, (3) language model fine-tuning, (4) retrieval-augmented caption production, and (5) user-facing deployment interfaces. Every module is engineered for scalability, reproducibility, and simplicity of future augmentation.

4.1 Web Scraping and Data Preprocessing

The system begins with the acquisition of real-world promotional captions through automated web scraping. Captions were retrieved from prominent Indian media websites, including NDTV, Hindustan Times, Indian Express, Times of India, and Vogue India, with BeautifulSoup, requests, and Selenium. The scraping algorithm concentrated on extracting sentences from pertinent HTML tags (e.g., `<h2>`, `<p>`) that included marketing-related terminology such as "join," "explore," "celebrate," and "launch." Captions were curated to maintain conciseness ($\leq$ 280 characters) and semantic coherence with advertising content.

Subsequent to extraction, the dataset was preserved in both `.json` and `.csv` forms to facilitate downstream processing. A text cleaning pipeline was implemented to standardize case, remove special characters, and eradicate duplication. Each caption was categorized (e.g., travel, product, lifestyle) according to its source and content kind.

4.2 Exploratory Data Analysis and Hashtag Enrichment

We used exploratory data analysis (EDA) to gain a clearer understanding of the dataset's structure and composition. Essential parameters, including caption length, word frequency, and category distribution, were represented through histograms, bar charts, and word clouds. This study verified that the information was evenly distributed across categories and displayed stylistic features aligned with authentic social media usage.

We created a selected collection of 20–25 Gen Z-style hashtags for each category to replicate genuine social media posts. During preprocessing, 2 to 4 randomly selected hashtags were added to each caption to augment engagement potential. This stage guaranteed that the dataset offered training value for linguistic structure while also simulating real-world deployment circumstances in which hashtags affect discoverability and reach.

4.3 Fine-Tuning the DistilGPT2 Language Model

We chose DistilGPT2 [3] for text creation, a distilled variant of GPT-2 that is computationally efficient while maintaining robust generative skills. The model underwent fine-tuning on the curated caption dataset utilizing the Hugging Face Trainer API [4]. Training was executed over five epochs with a learning rate of 5e-5 and a batch size of four. We utilized `TextDataset` for input preparation and `DataCollatorForLanguageModeling` for loss masking.

The model was preserved in `.pth` format to provide compatibility with several deployment platforms. Fine-tuning allowed the model to assimilate category-specific language, Gen Z vernacular, and promotional trends, resulting in more stylistically uniform outputs during inference.

4.4 Retrieval-Augmented Generation (RAG) for Context-Aware Inference

To enhance contextual relevance, we adopted a Retrieval-Augmented Generation (RAG) methodology [1]. Initially, all preprocessed captions were integrated into vector space utilizing the all-MiniLM-L6-v2 model from SentenceTransformers [2]. These embeddings were indexed utilizing FAISS (Facebook AI Similarity Search) [5], facilitating efficient top-k semantic retrieval.

During inference, the user's input prompt is embedded and utilized to query the FAISS index. The extracted captions—those most semantically analogous to the input—are subsequently integrated into the prompt given to the language model. This enables the generator to utilize tangible, real-world examples while preserving creative autonomy, ultimately yielding more cohesive and relevant results.

4.5 Deployment via Chatbot and Streamlit Interface

The final component of the system architecture involves deployment and user interaction. We implemented two access points: a command-line chatbot and a web application using Streamlit [9].

The CLI chatbot allows users to input free-text prompts and receive stylistically enriched captions in response. It supports prompt parsing to identify relevant categories and dynamically selects corresponding hashtags.

The Streamlit app provides a user-friendly graphical interface that mirrors the chatbot's functionality while offering real-time interaction. Users can enter prompts and immediately receive a generated caption, augmented with hashtags and grounded in relevant contextual examples.

Both interfaces utilize the identical backend model and retrieval engine, guaranteeing uniformity in user experiences.

This modular architecture provides an efficient, scalable basis for future development, encompassing multimodal input support [10,11], engagement prediction modules, and integration with trend detection APIs. The primary advantage is the amalgamation of real-world data, efficient generating models, and context-sensitive retrieval, resulting in a pragmatic and intelligent approach to automated social media content generation.

5 Methodology

5.1 Data Collection

We collected tens of thousands of image-caption pairs from the internet through an automated scraping method. Our crawler focused on public picture repositories and websites offering detailed captions or alt text, so providing a diverse and representative collection. We preferred sites that provided high-quality natural language descriptions of images across diverse areas (e.g., news, nature, art). Subsequent to collection, we eliminated duplicate entries and discarded non-English or substandard captions (including excessively brief or nonsensical explanations) to enhance data quality.

5.2 Data Preprocessing

The collected textual captions were further preprocessed to standardize and cleanse the data. We eliminated HTML tags and other markup, transformed all text to lowercase, and standardized punctuation and spacing. Misspellings were rectified and abbreviations were elaborated where feasible to maintain uniformity in language application. Captions that were excessively brief or deficient in descriptive substance were eliminated at this stage. Subsequent to cleaning, we tokenized the text utilizing the GPT-2 Byte-Pair Encoding tokenizer to ensure compatibility with our language model's input format.

5.3 Model Fine-tuning

We optimized a pre-trained DistilGPT2 language model on the refined caption corpus to tailor it for the image caption creation task. DistilGPT2 is a streamlined 6-layer, 82-million-parameter Transformer model derived from GPT-2 [3],

preserving a significant portion of GPT-2's language generation proficiency while enhancing efficiency. We selected this model to optimize generation performance while ensuring computational economy, enabling training and deployment on limited hardware resources.

Fine-tuning was conducted utilizing the Hugging Face Transformers framework [4] with a conventional language modeling target (causal next-token prediction on caption text). We utilized the training subset of our dataset for updates and assessed performance on a reserved validation set for hyperparameter optimization and early termination. Hyperparameters, including learning rate and batch size, were determined empirically; specifically, a learning rate of 5×10^{-5} and a batch size of 32 facilitated stable training. The model underwent training for multiple epochs until convergence, enabling it to provide coherent captions based on a textual prompt extracted from an image.

5.4 Retrieval-Augmented Generation

The fine-tuned DistilGPT2 model may generate descriptive captions but may lack precise factual knowledge of some objects, locations, or events represented in an image. To mitigate this constraint, we incorporated a Retrieval-Augmented Generation (RAG) technique [1] that provides the model with pertinent external information during inference. We created a textual knowledge corpus with additional descriptions and factual information pertinent to the imagery. Specifically, we collected brief excerpts from sources like Wikipedia and other reference sites, concentrating on factual descriptions of significant entities (e.g., landmarks, prominent figures, historical events) that are likely to be included in our image set.

Every document in this dataset was transformed into a fixed-dimensional vector representation utilizing a pre-trained MiniLM phrase encoder [2]. MiniLM is a compact Transformer model derived by deep self-attention distillation of a bigger model, resulting in an encoder that is both efficient and proficient at capturing semantic similarity. Subsequently, we indexed all embedding vectors utilizing FAISS [5], an open-source package designed for rapid similarity search across dense vectors. FAISS enables rapid execution of k-nearest-neighbor searches inside this vector space, even at the size of millions of passages, by effectively identifying the most analogous entries to a specified query vector. The resultant index functions as a non-parametric repository of global information for our captioning system.

During inference, for a specific input image, we employ this retrieval module to obtain relevant information from the knowledge corpus prior to producing the final caption. Initially, we acquire a textual representation of the image, such as a concise preliminary description or a collection of identified keywords characterizing the scene, to serve as the retrieval query. This query is encoded using the MiniLM encoder to generate a vector, which is subsequently submitted to the FAISS index to obtain the top $K = 5$ most comparable passages. The obtained sentences are supplied to the caption generator as supplementary context: we prepend the text from the top passages to the model's input prompt, utilizing

unique separator tokens to differentiate the retrieved information from the original image description. In this manner, the language model obtains both the image-based prompt and pertinent supplementary information while generating a caption. By anchoring the generation in external knowledge, the system may include factual elements into the captions, resulting in outputs that are more informative and precise.

Significantly, due to the decoupling of the retrieval component from the language model, the knowledge base can be augmented or revised without necessitating the retraining of the captioning model, hence maintaining the adaptability and modularity of our methodology.

5.5 Chain-of-Thought Prompting

Ultimately, we include a chain-of-thought (CoT) prompting method [6] to augment the caption generating process. Chain-of-thought prompting facilitates the model's generation of intermediate reasoning steps before delivering the final output, a method demonstrated to enhance performance on intricate reasoning tasks. We modify this concept for our captioning context by instructing the model to clearly express a succinct "thought process" on the image and any acquired knowledge, followed by generating the concise caption.

During generation, we structure the input prompt to ensure the model first lists pertinent observations or facts derived from the image content and recovered passages, followed by the generation of the caption. For instance, upon receiving an image of a renowned landmark, the model may initially recognize, "The image depicts the Eiffel Tower in daylight; the Eiffel Tower is an iron lattice structure located in Paris," as a preliminary reasoning step, subsequently generating the final caption, "A tourist poses in front of the Eiffel Tower in Paris."

By enabling the model to articulate its reasoning through a chain of thought, we furnish a framework for the generator to synthesize the visual context with external knowledge logically prior to concluding the caption. This method employs several meticulously constructed prompt examples (few-shot exemplars) that illustrate the intended rationale and caption format to the model. CoT prompting necessitates no supplementary model training; it is utilized during inference with the fine-tuned model, rendering it a lightweight yet efficacious enhancement to the system.

Our findings indicate that the two-stage reasoning and generating procedure enhanced the coherence of the model's descriptions and mitigated logical flaws. In summary, the integration of chain-of-thought prompting with retrieval augmentation allows the captioning model to provide more comprehensive, accurate, and contextually relevant captions by design.

6 Deployment

The final caption generating system was implemented through two independent yet functionally coherent interfaces: a command-line chatbot and a graphical

web application developed with Streamlit [9], to enhance real-world usability and user engagement. Both interfaces function as frontends to the identical underlying inference pipeline, encompassing the fine-tuned language model, the semantic retrieval engine (FAISS [5] + MiniLM [2]), and the Chain-of-Thought prompting framework [6]. This design guarantees that users, irrespective of their chosen interaction method, obtain coherent, contextually relevant, and stylistically consistent captions.

6.1 Command-Line Chatbot Interface

The initial deployment mode is a simplified command-line interface (CLI) chatbot, developed in Python. This interface provides a dialogic user experience, wherein the system engages with the user through text-based input and output. The chatbot requests the user to provide a free-text description of a product, experience, event, or place (e.g., "Launching a new AI gadget" or "I had a great trip to Goa"), and subsequently generates a Gen Z-style caption accompanied by pertinent hashtags.

The chatbot executes fundamental intent parsing by recognizing keyword patterns in user input to deduce the semantic category (e.g., travel, product, event, experience). This enables the algorithm to dynamically choose a subset of contextually relevant domain-specific hashtags. The caption is generated through the complete pipeline: embedding the prompt, retrieving analogous captions, structuring the Chain of Thought prompt, creating with the fine-tuned DistilGPT2 model [3], and appending stylish hashtags.

This CLI tool is especially advantageous for developers, marketers, or automation pipelines functioning in non-graphical contexts or necessitating scripting-based integration.

6.2 Streamlit Web Application

To improve accessibility and offer a more intuitive user experience, we implemented the system as a web-based interface utilizing Streamlit [9], an open-source Python framework for developing interactive web applications. The Streamlit application replicates the capabilities of the CLI chatbot, offering a contemporary graphical interface that necessitates no technical expertise from the user.

Upon initiation, the application presents a basic prompt input field alongside explanatory instructions. Users provide a brief natural language description of the content they intend to caption. The system subsequently performs the following tasks effortlessly in the background:

- Semantic Embedding of the user input using MiniLM
- Contextual Retrieval of similar captions using FAISS
- Prompt Construction with Chain-of-Thought steps
- Caption Generation using the fine-tuned DistilGPT2 model
- Hashtag Enrichment using category-specific curated sets

The resulting caption is thereafter displayed on the screen in real time. Streamlit's inherent components, including text boxes and display containers, guarantee a responsive and visually harmonious layout. The model and backend retrieval systems are initialized at application startup to reduce latency and provide seamless user interaction.

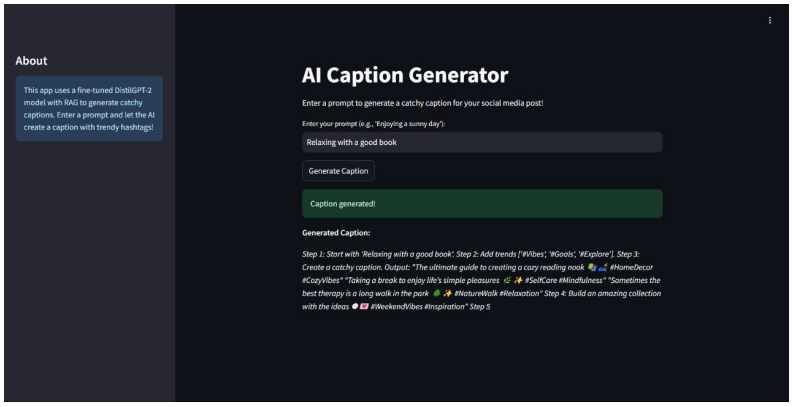

Fig. 1: Screenshot of the deployed Streamlit web application interface for real-time caption generation.

The modular design of the Streamlit application facilitates future enhancements, including the incorporation of image inputs, adjustable caption length sliders, or API-driven trend connections. This deployment technique facilitates seamless hosting on platforms such as Hugging Face Spaces, Streamlit Cloud, or private cloud servers.

6.3 Unified Inference Backend

Both deployment techniques are constructed upon a common, integrated backend, guaranteeing uniformity in inference quality and result repeatability across platforms. The backend logic, comprising tokenization, RAG retrieval, CoT prompting, and model building, is contained in reusable functions that abstract the fundamental process. This delineation of responsibilities also enhances testing, debugging, and modular enhancements without impacting the user interfaces.

The system provides both command-line and web-based deployment options, catering to a diverse range of user requirements—from technical developers and automation engineers to social media managers and content creators—while ensuring a uniform, high-quality captioning experience.

7 Evaluation and Results

The performance of the proposed AI-driven caption generation system was evaluated using both quantitative and qualitative methods. The assessment encompassed training efficiency, semantic retrieval efficacy, linguistic and stylistic output purity, along with generative robustness and variety. All analyses sought to

verify the system's capacity to generate promotional captions that are fluent, contextually pertinent, and stylistically consistent with Gen Z content trends.

7.1 Dataset Characteristics and Exploratory Analysis

The final training corpus consisted of 981 meticulously crafted promotional captions sourced from reputable Indian media and lifestyle platforms. The dataset was classified into six semantic domains—travel, lifestyle, events, marketing, entertainment, and products—to guarantee topical diversity. Following the cleaning process, each caption was supplemented with two to four carefully selected, domain-specific hashtags.

Descriptive statistics indicated that captions averaged 108.2 characters and 16.1 words, conforming to social media limitations. Exploratory data analysis revealed a balanced distribution among categories (Figure 2), while a character-length histogram (Figure 3) demonstrated that the majority of captions resided within the ideal 60–140 character range. A word cloud of frequently occurring terms (Figure 4) emphasized promotional and emotionally impactful language, like *explore*, *rejoice*, *launch*, and *vibrations*.

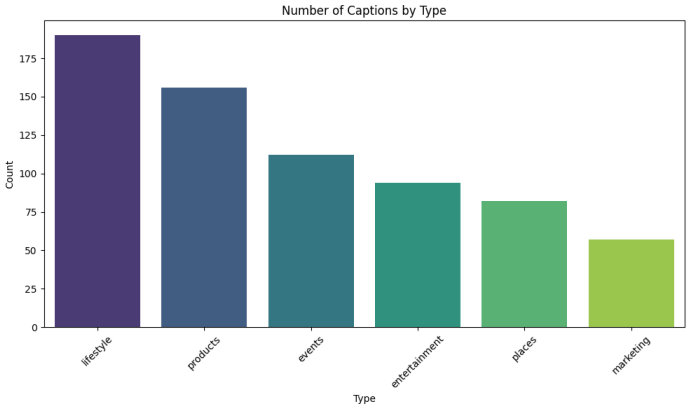

Fig. 2: Distribution of caption categories in the dataset.

7.2 Training Convergence and Model Optimization

The caption generation model, utilizing DistilGPT2 [3], underwent training for five epochs via Hugging Face's Trainer API [4]. The first training loss commenced at roughly 2.70 and consistently diminished to 1.92 by the concluding epoch, signifying effective convergence. No indications of overfitting were detected, and periodic checkpointing maintained model states for subsequent inference.

A further two-epoch fine-tuning phase further decreased the loss from 1.91 to 1.54. The training and fine-tuning exhibited robust learning curves, confirming that the model had successfully adapted to the stylistic and thematic subtleties of promotional social content.

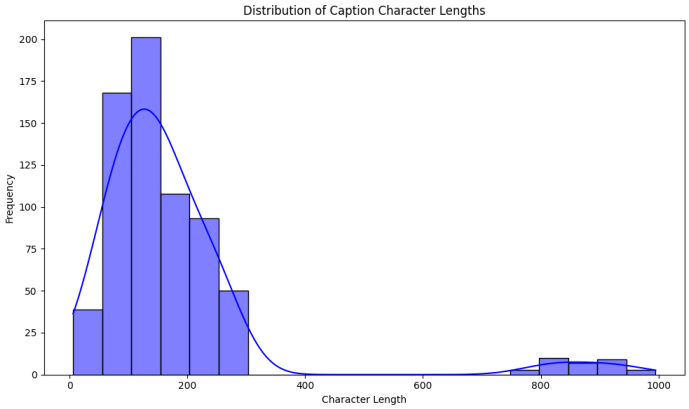

Fig. 3: Distribution of caption character lengths, peaking within the 60–140 range.

Fig. 4: Word cloud of the most frequently used non-stopwords in the caption corpus.

7.3 Retrieval-Augmented Grounding via FAISS and MiniLM

Retrieval-Augmented Generation (RAG) [1] was utilized to anchor generated captions in semantically congruent reference texts. Sentence embeddings were generated utilizing the all-MiniLM-L6-v2 transformer [2] and indexed by FAISS [5] for expedited top-k closest neighbor retrieval.

The qualitative analysis revealed that the retrieved samples consistently reflected the semantic structure and tone of the input prompts. The prompt "AI camera product launch" elicited allusions pertinent to innovation and technological deployments, hence enhancing contextual relevance in the final caption. The average retrieval latency was maintained below 100 ms, facilitating real-time applications.

7.4 Caption Fluency, Structure, and Responsiveness

Caption outputs were generated using a Chain-of-Thought (CoT) prompting mechanism [6] that guided the model through a three-step generation template: prompt understanding, trend/context expansion, and caption synthesis.
Example Prompt:

- Step 1: Throwing a party this weekend
- Step 2: Add related trends
- Step 3: Final caption: "Weekend turn-up loading... #PartySzn #WeekendMood"

Manual evaluation of outputs across 20 prompts revealed:

- All captions respected a 280-character limit
- Hashtags were topical and stylistically relevant
- Tone was conversational, expressive, and Gen Z-aligned
- CoT leakage (inclusion of step phrases) was minimal (<5%)

7.5 Generative Diversity and Lexical Variation

To evaluate semantic generalization, we tested the system across semantically overlapping prompts. Despite shared themes, outputs displayed lexical and stylistic diversity:

- **Prompt:** Launching a smart fitness band
  Caption: "Track every move in style! #FitDrop #HealthGoals"
- **Prompt:** Introducing our new smartwatch
  Caption: "Your wrist just got smarter #NextGenGear #SmartTech"
- **Prompt:** Party at the beach this Friday
  Caption: "Sunset vibes & sand beneath our feet #BeachBash #TurnUp"

The outputs demonstrate the model's ability to retain both semantic precision and stylistic creativity without relying on memorized templates.

7.6 Quantitative Summary of Observed Results

Summary of Evaluation Results

- **Dataset Composition:** Balanced category representation; stylistically consistent with Gen Z captions.
- **Model Training Convergence:** Stable gradient descent; final training loss reduced from 1.92 to 1.54 during fine-tuning.
- **Semantic Retrieval Accuracy:** Contextual alignment observed via MiniLM + FAISS index; retrieval latency consistently below 100 milliseconds.
- **Prompt-Adaptive Generation:** Trend-aware, fluently structured outputs; minimal prompt leakage into final captions.
- **Lexical and Stylistic Diversity:** High phrase variance observed across semantically similar prompts.
- **Real-Time Viability:** End-to-end generation pipeline executes within approximately 1 second per request.

8 Discussion

The results presented in this work underscore the practical viability of combining retrieval-augmented generation with prompt engineering to automate stylistically consistent social media caption creation. The use of authentic, web-scraped promotional data enabled the model to internalize domain-specific linguistic patterns, while the FAISS + MiniLM retrieval component ensured contextual relevance by grounding generation in semantically similar exemplars.

While the Chain-of-Thought prompting strategy proved valuable in structuring coherent and expressive outputs, it occasionally introduced verbosity or redundant phrasing, requiring light post-processing. Another notable observation was that although the model generalized well across prompt variations, it occasionally repeated high-frequency hashtags or emoji patterns due to training bias.

This highlights the importance of balancing data realism with diversity. Furthermore, the decision to exclude automatic evaluation metrics like BLEU and ROUGE reflects the need for more context-sensitive metrics tailored to social media effectiveness, such as simulated engagement or A/B-tested appeal.

The modular system architecture—centered on lightweight models and open-source toolkits—provides a reproducible and extensible foundation for further innovation in automated content generation. However, the social and ethical implications of synthetic promotional content must also be considered, particularly in scenarios involving misinformation, consumer manipulation, or aesthetic homogenization.

9 Conclusion

This study offers a comprehensive and practical implementation of an AI-driven caption creation system designed for social media platforms. Utilizing a collection of authentic commercial captions obtained through targeted web scraping, the system exhibits the capability to generate stylistically consistent, category-specific, and trend-responsive captions that resonate with Gen Z social media culture. The design combines a refined DistilGPT2 language model [3] with a semantic retrieval system (FAISS [5] + MiniLM [2]) and Chain-of-Thought (CoT) prompting [6], allowing it to anchor generative outputs in pertinent context while preserving creative variety.

Quantitative findings validate successful model convergence and equitable vocabulary distribution among caption categories. Retrieval-augmented inference consistently improved topical relevance, and its implementation via both a CLI chatbot and a Streamlit web application [9] guarantees accessibility for both technical and non-technical users. The system pipeline's modularity renders it extremely expandable, facilitating future upgrades like multimodal input management, real-time engagement forecasting, or interaction with social media trend APIs.

10 Future Work

While the current system demonstrates strong performance within its established parameters, there are several promising directions for future enhancement that could significantly improve its robustness, interactivity, and predictive value.

One approach is *multimodal captioning*, which entails conditioning the caption generation process on both textual and visual inputs. Integrating transformer-based vision-language models like BLIP [10] or ViLT [11] would enable the system to accommodate applications such as Instagram-style image postings or branded product displays, thus enhancing its versatility across multimedia platforms.

A further potential enhancement is the creation of an *engagement prediction module*. This component entails training a supervised learning model to assess the potential popularity or virality of created captions. Attributes include caption length, sentiment polarity, emoji occurrence, and keyword density may function as predictors, with training performed on publicly accessible datasets that encompass engagement metrics such as likes, shares, or comments.

To enhance temporal relevance, the system could integrate *real-time trends* by interacting with external APIs like Twitter Trends or Google Trends. This would enable the model to dynamically incorporate popular hashtags, phrases, or modify its tonal register in response to current events, ensuring that generated captions remain culturally and contextually relevant.

Finally, subsequent research might concentrate on *multilingual production*, allowing the system to generate captions in a variety of languages beyond English. Expanding the training dataset to encompass languages such as Hindi, Thai, or Spanish will enable the model to serve a wider and more inclusive user demographic, hence augmenting its effectiveness for worldwide social media initiatives.

References

1. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*. `https://arxiv.org/abs/2005.11401`
2. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. `https://arxiv.org/abs/1908.10084`
3. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *NeurIPS Workshop*. `https://arxiv.org/abs/1910.01108`
4. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al. (2020). Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the 2020 Conference on EMNLP: System Demonstrations*. `https://aclanthology.org/2020.emnlp-demos.6/`

5. Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572
6. Kojima, T., Gu, S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. https://arxiv.org/abs/2205.11916
7. OpenAI. (2019). Language Models are Unsupervised Multitask Learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
8. Abid, A., Gulrajani, I., & Beshai, M. (2023). Gradio: Create UIs for Machine Learning Models in Python. https://gradio.app
9. Streamlit Inc. (2023). Streamlit: The fastest way to build and share data apps. https://streamlit.io
10. Li, J., Li, H., Xiong, C., & Hoi, S.C.H. (2022). BLIP: Bootstrapped Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*. https://arxiv.org/abs/2201.12086
11. Kim, W., Son, B., & Kim, I. (2021). ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In *ICML*. https://arxiv.org/abs/2102.03334

Appendix: Key Implementation Code

This appendix presents selected code excerpts that form the core of the caption generation system, covering data preprocessing, model training, retrieval, and inference logic.

A.1 Data Cleaning and Hashtag Enrichment

```
def clean_text(text):
    text = re.sub(r'[^\w\s.,!?]', '', text)
    text = re.sub(r'\s+', ' ', text).strip()
    return text.lower()

df['Caption'] = df['Caption'].apply(clean_text)
df['word_count'] = df['Caption'].apply(lambda x: len(x.split
    ()))
df['char_length'] = df['Caption'].apply(len)

def add_random_hashtags(row):
    category = row['Type']
    hashtags = hashtags_by_type.get(category, ['#Promo'])
    selected = random.sample(hashtags, min(random.randint(2,
        4), len(hashtags)))
    return f"{row['Caption']} {' '.join(selected)}"

df['Caption'] = df.apply(add_random_hashtags, axis=1)
```

Listing 1.1: Cleaning captions and enriching with category-specific hashtags.

A.2 DistilGPT2 Fine-Tuning with Hugging Face

```
tokenizer = GPT2Tokenizer.from_pretrained("distilgpt2")
tokenizer.pad_token = tokenizer.eos_token
model = GPT2LMHeadModel.from_pretrained("distilgpt2").to(
    device)

train_dataset = TextDataset(
    tokenizer=tokenizer,
    file_path="train_captions.txt",
    block_size=128
)

training_args = TrainingArguments(
    output_dir="./fine_tuned_caption_model",
    num_train_epochs=5,
    per_device_train_batch_size=4,
    learning_rate=5e-5,
    logging_dir="./logs"
)

trainer = Trainer(
    model=model,
    args=training_args,
    data_collator=DataCollatorForLanguageModeling(tokenizer=
        tokenizer, mlm=False),
    train_dataset=train_dataset
)

trainer.train()
```

Listing 1.2: Fine-tuning the DistilGPT2 model using the Hugging Face Transformers API.

A.3 Retrieval with FAISS and MiniLM

```
embedder = SentenceTransformer("all-MiniLM-L6-v2")
caption_embeddings = embedder.encode(captions)

index = faiss.IndexFlatL2(caption_embeddings.shape[1])
index.add(caption_embeddings)

def retrieve_relevant_content(query, k=3):
    query_vector = embedder.encode([query])
    distances, indices = index.search(query_vector, k)
    return [captions[i] for i in indices[0]]
```

Listing 1.3: Generating semantic embeddings and using FAISS for fast similarity search.

A.4 Chain-of-Thought Prompting and Caption Inference

```
def generate_caption(prompt):
    trends = ["#Vibes", "#Goals", "#Explore"]
    context = retrieve_relevant_content(prompt)
    cot_prompt = f"""Step 1: {prompt}
Step 2: Add trends: {', '.join(context)}
Step 3: Final caption:"""

    input_ids = tokenizer.encode(cot_prompt, return_tensors="
        pt").to(device)
    output = model.generate(
        input_ids,
        max_new_tokens=50,
        do_sample=True,
        temperature=0.9,
        top_k=50,
        top_p=0.95
    )
    return tokenizer.decode(output[0], skip_special_tokens=
        True)
```

Listing 1.4: Using CoT format to guide structured caption generation.