Echoes of Imagination

1st Deivanai Thiyagarajan
Department of Engineering Education)
University of Florida
Gainesville, Florida
dthiyagarajan@ufl.edu

***Abstract*—This project, Echoes of Imagination, aims to generate illustrative visuals from narrative text through progressive multimodal modeling. The initial stage implemented a text-conditioned Variational Autoencoder (VAE) in which only the decoder was conditioned on textual embeddings derived from BERT and CLIP encoders [4]. To improve text–image alignment, the latent dimension was increased, and both the encoder and decoder were subsequently conditioned on text features. Although this enhanced reconstruction quality, generation from text alone remained unstable. Building on these findings, the next phase transitioned to diffusion-based architectures, employing text-conditioned U-Nets with encoder–decoder cross-attention to strengthen semantic grounding and improve image fidelity.**

After exploring these custom architectures, we incorporated existing pretrained diffusion models and fine-tuned them on our domain-specific dataset. This approach produced more reliable image generation and improved semantic alignment, reflected in higher CLIP scores compared to the non-fine-tuned baseline. However, despite the quantitative improvement, the visual differences between pretrained and fine-tuned outputs remained relatively subtle. Together, these iterative developments establish a foundation for robust multimodal storytelling by linking textual semantics and visual synthesis within a unified generative framework.

***Index Terms*—Text-to-Image Generation, Multimodal Learning, Variational Autoencoder (VAE), Diffusion Models, CLIP Embeddings [4], Fine-Tuning, Semantic Alignment, Pretrained Models, Latent Space Modeling, Multimodal Storytelling**

I. Introduction

torytelling is one of humanity's oldest and most expressive traditions, yet it remains largely confined to static text. Echoes of Imagination explores how multimodal AI can bridge this gap by transforming written narratives into coherent and expressive visuals, creating more immersive storytelling experiences. The project began with a text-conditioned Variational Autoencoder (VAE), where text embeddings from BERT and CLIP guided the decoder during image reconstruction. While the model could reproduce visual features, text-only inputs yielded vague outputs, showing weak text–image alignment. Conditioning both the encoder and decoder on text and increasing the latent dimension improved quality but retained the blurriness typical of VAEs.

To overcome these limits, the project transitioned to diffusion-based architectures, using text-conditioned U-Nets that refine images through iterative denoising. Integrating BERT and CLIP embeddings via cross-attention enhanced semantic coherence and image fidelity. However, as custom-from-scratch models remained unstable for open-ended text generation, the effort shifted toward leveraging strong pretrained models. In the final phase, the project explored and fine-tuned models such as sd-turbo for text-to-image generation and instruct-pix2pix [3] from huggingface for text-guided image-to-image transformation, which offered more reliable and semantically grounded results. These pretrained models were evaluated using quantitative metrics such as the OpenAI CLIP score [4], supplemented by qualitative visual assessment.

II. Related Work

Text-to-image synthesis has seen transformative advances with the emergence of diffusion models. Earlier GAN-based systems such as AttnGAN and DM-GAN achieved limited fidelity for complex prompts [3]. Latent diffusion models (LDMs) like Stable Diffusion [2] compress high-dimensional images into manageable latent spaces, enabling efficient training and high-quality outputs. Multimodal embedding models such as CLIP [4] and BERT [5] further allow semantic conditioning on text. The Echoes of Imagination framework leverages these advances while emphasizing continuity across narrative fragments—a direction less explored in conventional text-toimage pipelines.

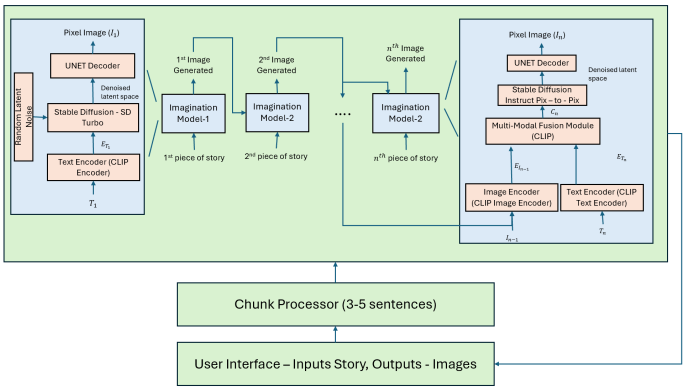

Fig. 1. Architecure Diagram.

III. System Architecture and Pipeline

The proposed system operates iteratively to generate contextually coherent visuals from narrative text. Each iteration produces an image that reflects the current story segment while maintaining visual continuity with previous scenes. The overall

architecture and data flow are shown in "Fig. 1". The process consists of four key stages:

A. Data Input

The system receives the story text, which is divided into smaller narrative units or *storylets* T_i.

B. Initial Text-to-Image Generation (Model-1)

For the first storylet T_1, a pretrained Stable Diffusion model [2] (`stabilityai/sd-turbo`) is used to generate the initial image I_1 directly from text. This model serves as the text-to-image backbone for initializing the visual narrative.

C. Model Encoding for Iteration $n > 1$

For each subsequent storylet T_n, the system incorporates both modalities. The text encoder (BERT or CLIP) produces the textual embedding $E(T_n)$, while the image encoder extracts the visual embedding $E(I_{n-1})$ from the previously generated image.

D. Diffusion Model and Conditioning (Model-2)

The second-stage generator, based on a diffusion backbone such as `timbrooks/instruct-pix2pix` [3], operates in latent space and accepts both image and text as conditioning inputs. The UNet denoiser is trained using a Denoising Diffusion Probabilistic Model (DDPM), operating over 1000 diffusion steps and predicting noise residuals iteratively. A pretrained VAE encodes each image into a $4 \times 32 \times 32$ latent representation, reducing computation by $16\times$ compared to pixel-space diffusion. Conditioning on BERT or CLIP embeddings aligns the denoising process with narrative semantics.

E. Training Strategy

Training was conducted on the HiPerGator cluster using an NVIDIA B200 GPU with 72GB VRAM. Mixed-precision training, gradient accumulation, and distributed data parallelism were employed to manage memory efficiently. The learning rate was set to 1×10^{-4} with cosine annealing and a batch size of four. Loss decreased from 0.82 to approximately 0.38 after five epochs. The dataset included COCO and Flickr30k captions to provide both object-centric and descriptive scenes. When caption-image pairs were missing, synthetic dummy data generators ensured uniform batch flow.

F. Inference and Multi-Modal Fusion

The Multi-Modal Fusion Module uses cross-attention to combine $E(T_n)$ and $E(I_{n-1})$ into a joint conditioning vector:

$$C_n = f(E(T_n), E(I_{n-1})).$$

This vector guides the diffusion UNet to denoise a sampled latent variable z, generating a new latent representation that preserves stylistic and structural continuity from I_{n-1}.

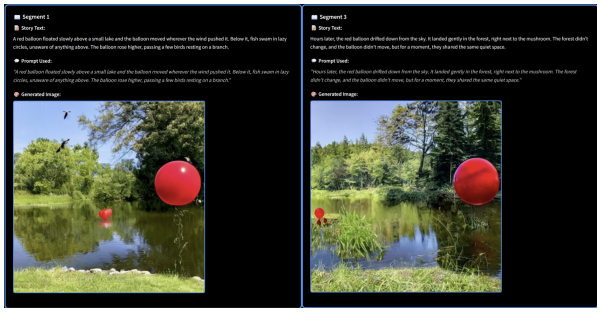

Fig. 2. User Interface.

G. User Interface and Reconstruction

The VAE decoder maps the final latent output back to pixel space, producing the image I_n, which is displayed to the user through the interface. The user interface, illustrated in "Fig. 2", provides a structured layout for viewing the generated image along with its corresponding storylet and optional summary.

The generated image I_n is then reused as I_{n-1} for the next iteration, enabling consistent visual storytelling across the sequence of storylets and maintaining continuity between adjacent narrative segments.

This two-stage pipeline—initial text-to-image generation via Stable Diffusion [2], followed by iterative text+image generation via an instruction-guided diffusion model—ensures temporal coherence, narrative consistency, and smooth visual progression throughout the story.

IV. Model Implementation Details

A. Frameworks and Libraries

The project was implemented in Python using PyTorch as the primary deep learning framework. Key libraries include:

- **Core:** numpy, pandas, matplotlib, seaborn, scikit-learn, tqdm
- **Deep Learning:** torch, torchvision, torchaudio, transformers, accelerate, sentencepiece, einops
- **Text-to-Image / Diffusion / VAE / CLIP:** diffusers, open_clip_torch, Pillow
- **Pretrained Models:**
  - `stabilityai/sd-turbo` for initial text-to-image generation (Model-1)
  - `timbrooks/instruct-pix2pix` for text+image conditioned generation (Model-2)
- **UI and Visualization:** gradio, streamlit
- **Utility:** requests, jsonlines, pyyaml
- **Optional:** jupyter, graphviz
- **Dataset Handling:** kagglehub

B. Hardware and Training Setup

Training and fine-tuning were performed on the HiPerGator cluster (hpg-b200 node) with the following resources:

- **GPU:** NVIDIA B200 (72GB VRAM)
- **CPU:** 4 cores
- **System Memory:** 30 GB

Model-1 (sd-turbo) was used in inference mode only, while Model-2 (Instruct-Pix2Pix) [3] was fine-tuned for text+image generation. Mixed-precision training and gradient accumulation were essential to manage memory constraints during diffusion fine-tuning.

C. Hyperparameters

VAE-Based Model (Early Phase):

- Optimizer: AdamW
- Learning rate: 2×10^{-4}
- Batch size: 16
- Epochs: 5
- Latent dimension: increased from 4 to 8
- Loss: MSE reconstruction + KL divergence (with optional L1 term)

Diffusion Model Fine-Tuning (Final Phase):

- Backbone: UNet latent diffusion model (Instruct-Pix2Pix [3])
- Scheduler: DDPM-based with cosine noise schedule
- Learning rate: 1×10^{-4} with cosine annealing
- Batch size: 4 (due to high VRAM usage)
- Epochs: 5 (limited by compute and memory)
- Training precision: mixed (fp16/bf16)
- Loss function: noise prediction MSE (standard diffusion objective)

Evaluation Metrics:

- CLIPScore (OpenAI CLIP-ViT-L/14)
- Qualitative visual comparison across iterations

Fine-tuning improved CLIPScore by approximately **0.5%**, although visual outputs showed minimal changes due to limited training epochs.

D. Reproducibility

To ensure reproducible experiments:

- Random seeds for PyTorch, numpy, and Python were fixed.
- Pretrained text encoders (BERT, CLIP) were frozen during early VAE experiments.
- All preprocessing scripts, dataset splits, and model configurations are version-controlled.
- Core components (VAEEncoder, TextConditionedDecoder, diffusion UNet, cross-attention layers) are modularized and documented.
- Diffusion fine-tuning checkpoints, sample generations, and CLIPScore logs are saved after each epoch.

V. Interface Prototype

The interface prototype for *Echoes of Imagination* provides a user-friendly environment for transforming narrative text into illustrated sequences. It is implemented using Gradio, a lightweight Python library for building interactive web interfaces.

A. Input and Output

- **Input:** Users provide a story in free-form text via a multiline textbox. The system expects complete sentences but is robust to longer paragraphs.
- **Processing:** The system automatically:
  1) Splits the input story into smaller storylets (4–5 sentences each) using NLTK's sentence tokenizer.
  2) Summarizes each storylet using a BART-based summarization pipeline [10] to reduce token count ($<$ 77 tokens), ensuring compatibility with the diffusion model.
  3) Generates an image for each summarized storylet using a Stable Diffusion 2.1 [2] pipeline [2].
- **Output:** For each storylet, the user is shown:
  - The original story segment (Markdown block),
  - The generated illustration (Image block),
  - Optional summarized text for reference.

B. Interface Layout

The interface uses a column-based layout where each storylet and its generated image are displayed sequentially. Users interact with the system through the following controls:

- **Generate Story Sequence:** Runs the full pipeline on the provided narrative.
- **Regenerate All:** Re-generates images for previously processed storylets.
- **Clear All:** Clears the current output display.
- **Feedback Buttons (Like / Dislike):** Allows users to express their preference for each generated illustration.

The layout is designed for clarity and minimal friction, enabling users to focus on the narrative progression and the visual outputs.

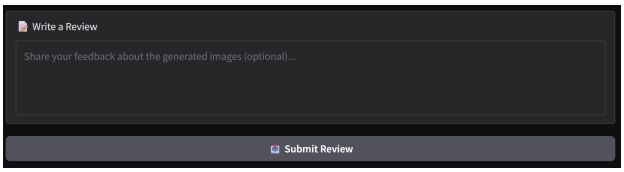

Fig. 3. User Feedback Collection.

C. User Feedback and Review Collection

In addition to Like/Dislike buttons, the interface includes a free-form **review textbox** for users to provide written feedback on image quality, text–image alignment, visual consistency, or general suggestions. Both structured and textual feedback are logged for qualitative analysis and to guide future improvements of the models and interface, see "Fig. 3".

VI. Refinements Made Since Deliverable 2

Since Deliverable 2, several key improvements have been made to enhance the robustness, usability, and stability of *Echoes of Imagination*.

A. Dual-Model Sequential Pipeline

A two-stage generation pipeline was implemented to improve visual consistency. In the first stage, the model generates the initial image directly from the story text. In the second stage, subsequent images are refined using both the previous image and the next story segment. This ensures that characters maintain consistent appearance, scenes evolve coherently, and modifications are controlled across the narrative sequence.

B. Robust Text Preprocessing

The text preprocessing pipeline was strengthened to handle longer narratives reliably. Stories are automatically segmented into smaller, coherent chunks and summarized to meet token limits while preserving key entities and actions. Tokenization ensures compatibility with the image generation models. These refinements prevent token overflow, maintain semantic integrity, and allow efficient and reliable image generation.

C. Improved Error Handling and Recovery

Error handling and recovery mechanisms were enhanced to improve system reliability. The pipeline now includes automatic fallback for tokenization or generation errors, device-aware configuration for CPU or GPU usage, and type handling to prevent image format mismatches. These measures allow the system to degrade gracefully rather than crashing, ensuring a smoother user experience during long-running operations.

D. Model Stability and Efficiency

Model stability was improved through memory-efficient inference, stable numerical precision for fine-tuned models, and safe model file loading. Memory optimizations reduce the risk of out-of-memory errors, FP32 precision ensures numerical consistency, and safe model serialization enhances security. Together, these changes improve reliability and consistency across diverse hardware setups.

E. Enhanced User Interface

The user interface was redesigned for better usability and interactivity. A two-column layout separates story input from generated images, while interactive controls allow users to regenerate images and provide feedback. Visual cues and real-time progress indicators improve transparency, making it clear to users how the system is processing their story.

F. State Management and Progress Tracking

State management and progress tracking were refined to give users more control over image generation. Users can regenerate images without re-entering story text, preserving the original segmentation. Real-time progress updates allow users to monitor the generation at each stage, supporting iterative exploration and enhancing confidence in the storytelling process.

VII. Evaluation and Results

The initial evaluation explored the Variational Autoencoder (VAE) as a generative backbone for the system. Experiments were conducted with a latent dimension of 4 and text conditioning applied only to the decoder. While the VAE could reproduce coarse structural details of training images, fine features such as texture, lighting, and color gradients were largely lost. Attempts to generate novel images from text prompts produced outputs that were noisy and lacked coherence, indicating that the latent space alone was insufficient for robust text-to-image synthesis. Increasing the latent dimension and conditioning both the encoder and decoder did improve reconstruction quality, with better preservation of shapes, color distribution, and basic composition, see“Fig. 4”. Despite these improvements, generating images from unseen text prompts remained unreliable, emphasizing the limitations of the VAE architecture for complex story-driven image generation, see “Fig. 5” .

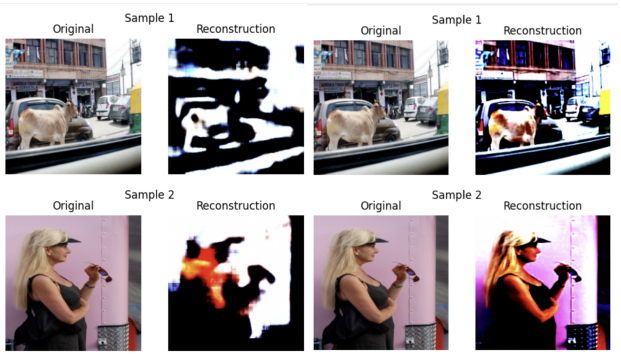

Fig. 4. Sample VAE reconstruction results. Increasing latent dimension improves detail.

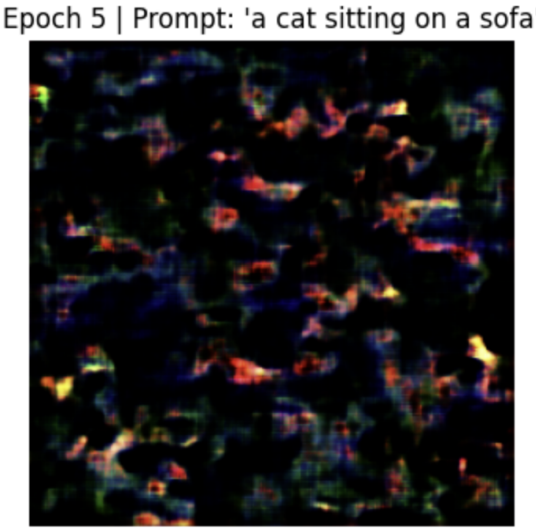

Fig. 5. Sample VAE image generation results.

To overcome the limitations observed with VAEs, the project transitioned to diffusion-based architectures, specif-

ically pretrained Stable Diffusion pipelines [2]. Diffusion models leverage iterative denoising in latent space, allowing for more precise image generation and better semantic alignment with text prompts. The pretrained models were fine-tuned on story-driven image-caption datasets to further enhance alignment with narrative content. Fine-tuning was conducted for five epochs with an average training loss decreasing from 0.2117 to 0.2017, showing stable convergence. During training, mixed precision and gradient accumulation were employed to optimize memory usage and ensure stable learning. The use of pretrained weights enabled faster convergence compared to training from scratch and allowed the system to generate visually coherent images even for complex prompts, see "Fig. 6".

Qualitative assessment of the diffusion model outputs showed significant improvements over the VAE. Images generated from text were clearer, better structured, and visually consistent across sequential storylets. Characters maintained consistent appearances, and environmental details such as lighting, perspective, and object placement were more coherent. Fine-tuning led to modest improvements in alignment, with certain prompts producing images that captured subtle narrative details more effectively. The sequential pipeline—initial text-to-image generation followed by image-to-image refinement using subsequent storylets—helped preserve character continuity and maintain scene evolution across multiple story segments.

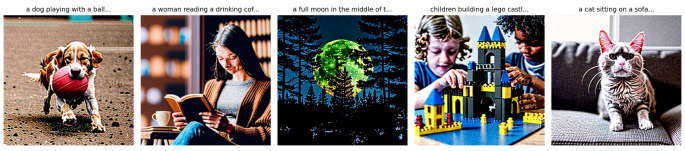

Fig. 6. Sample image generation using the pretrained diffusion model, Fine-tuned with the new dataset.

Quantitative evaluation was performed using CLIP scores, which measure the semantic alignment between generated images and textual prompts. Table I presents the comparison between the pretrained model and the fine-tuned variant across five prompts. Fine-tuning consistently improved CLIP scores, though the gains were modest, averaging a 1.3% increase. The highest improvement was 1.8% for prompts containing multiple objects or actions, while simpler prompts showed smaller gains. These results suggest that while fine-tuning on a limited dataset improves semantic alignment, longer training or additional dataset augmentation may be required for more substantial improvements.

Overall, the evaluation demonstrates that diffusion-based pipelines significantly outperform VAEs in terms of visual quality, semantic coherence, and consistency across story sequences. The combination of qualitative analysis, training loss trends, and CLIP-based quantitative evaluation confirms that the current architecture effectively generates narrative-driven illustrations. While fine-tuning produces only modest improvements, it lays the foundation for further refinement using larger datasets or more extensive training schedules.

TABLE I
COMPARISON OF CLIP SCORES BETWEEN PRETRAINED AND FINE-TUNED MODELS

Prompt	Pretrained	Fine-Tuned	Improvement
a dog playing...	31.88	32.23	+1.1%
a woman reading a...	32.81	33.39	+1.8%
a full moon in...	32.42	32.91	+1.5%
children building...	33.76	34.20	+1.3%
a cat sitting...	29.61	29.88	+0.9%
Average	32.10	32.52	+1.3%

VIII. RESPONSIBLE AI REFLECTION

During the development of the *Echoes of Imagination* story-to-image system, several ethical and fairness considerations were identified.

A. Bias in Generated Images and Narratives

Pretrained diffusion models can inherit biases present in their training data, potentially generating images that reinforce stereotypes or omit representation of certain groups. Beyond visual bias, there is also the risk of *narrative bias*, where certain story elements or characters might be underrepresented or misrepresented in the generated illustrations. To mitigate these issues, we plan to curate diverse and balanced prompts during evaluation and explore fine-tuning on datasets with equitable representation. We also aim to analyze outputs for fairness across different demographic and cultural contexts.

B. Content Safety and Sensitive Story Elements

Generated images may unintentionally produce unsafe, offensive, or inappropriate content. Additionally, stories that contain sensitive themes—such as violence, trauma, or personal experiences—pose challenges in ensuring that the visualizations are respectful and appropriate. To address these concerns, we will integrate content moderation filters provided by the model's safety pipeline and allow user feedback to flag undesirable outputs. Special care will be taken when representing sensitive narrative content, avoiding visualizations that could misinterpret or sensationalize such stories.

C. Privacy of User Data

Since the interface collects user-provided stories, there is potential for exposure of sensitive or personal information. We plan to implement local inference options and ensure that input data is not stored permanently, thereby minimizing privacy risks. Users will be informed about data handling practices to maintain transparency and trust.

D. Transparency and User Awareness

Users should understand that the images are AI-generated and may not faithfully reflect reality. The interface will include disclaimers and guidance on responsible use of generated content, emphasizing that outputs are creative interpretations rather than factual representations.

In the refinement phase, we aim to systematically incorporate these measures, ensuring that the system aligns with responsible AI principles while maintaining creativity, usability, and inclusivity in the storytelling experience.

ACKNOWLEDGMENT

The author thanks the University of Florida HiPerGator team for providing computational resources. The implementation and source code for *Echoes of Imagination* are available at: https://github.com/yourusername/echoes-of-imagination.

REFERENCES

[1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, and G. Krueger, "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[6] T. Wolf, et al., "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pp. 38–45, 2020.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint arXiv:2106.09685*, 2021.

[8] A. Abid, M. Farooqi, and J. Zou, "Gradio: Hassle-Free Sharing and Testing of Machine Learning Models in Python," *arXiv preprint arXiv:1906.02569*, 2021.

[9] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.