Character Grounding and Planning in Visual Story Generation

Frank Keller

School of Informatics
University of Edinburgh

THE UNIVERSITY *of* EDINBURGH
informatics

Joint work with Danyang Liu and Mirella Lapata

Content

- Visual storytelling task
- Modeling characters; planning

Part 1: Characters in visual stories

- The VIST-Character dataset
- Tasks and baselines: character detection, grounding, and ranking

Part 2: Planning in visual stories

- Using Blueprints to improve story planning
- Human evaluation of generated stories

Visual Storytelling

Input: Image sequence
Output: Story

Image Sequence:





Gold-standard Story:

It was customary to give big brass a full tour. Of course, the mess hall would usually serve something a little better than average on "Big Brass" days. He looked like he was okay with it. After lunch the tour continued. He was introduced to the Commanding officers.

Why is this Task Hard?

- It’s not just image description: we want a coherent story
- A story has a narrative structure involving:
  - Characters
  - Events
  - Locations
- Characters re-occur, **need to be recognized and grounded; need to distinguish protagonists from side characters**
- Events connect characters and locations and drive the narrative; **this needs to be planned**
- Humans want stories that are interesting, suspenseful, evocative; **traditional evaluation (Bleu etc.) doesn’t work well**

Characters and Plans

In this talk, we will focus on:

Characters in visual stories:

- Detect the characters in an image sequence
- Ground them in the text of the story
- Resolve co-reference across the two modalities
- Rank characters by importance

We introduce a new dataset and an unsupervised model

Liu and Keller, AAAI 2023

Planning in visual story generation:

- Generate questions-answer plans (Blueprints) for stories
- Use them to select key concepts and construct a coherent narrative
- Build a controllable, iterative story generation model

Improves grounding, coherence, interestingness of stories

Liu, Lapata, and Keller, EMNLP Findings 2023

Part 1: Characters

Current Visual Story Telling Models

- SotA models exploit external knowledge bases to enrich the detected objects
- Some models construct scene graphs as input for text generation
- Such methods are **unable to model the characters** in a story
- Existing approaches also don’t capture the **importance of characters,** and **can’t distinguish protagonists from side characters**

Output of Existing Systems

KE-VIST generates stories with **incorrect co-reference** and **arbitrary characters**

Image Sequence:

Story Generated by KE-VIST:

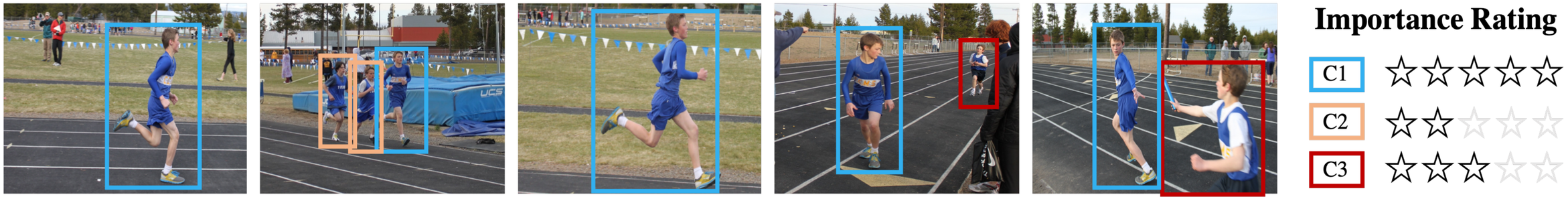
We all met up with the men. They walked a lot in the man walking. We had a lot of food. I gave a speech. Everyone was looking forward to it. Afterward, we all got together for pictures.

There’s no previous visual story dataset with character annotation

VIST-Character Dataset

We augment the test set of VIST dataset with:

- Visual and textual character co-reference chains, and their alignments
- Importance ratings for characters



Tom was getting ready for the track meet up. His friends were helping him by chasing after him. This wasn't good for Tom's nervous though so he can faster. He finished his lap and turned around because he heard someone call his name. It was just Steve trying to hit him with one of the batons. Grow up Steve.

Tasks

1. Based on VIST-Character dataset, we propose three tasks:
   - **Character detection and co-reference:** identify the characters and their co-reference in text and image sequence
   - **Character grounding:** ground the textual mentions of characters to the relevant bounding boxes in the image sequence
   - **Character ranking:** rank characters based on their importance to the story
2. For each task, we develop simple, unsupervised models as baselines

Overall Architecture

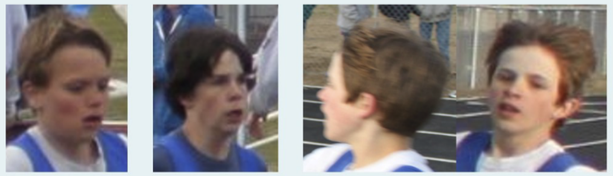
Input Story and Images → **Detection and Coreference** → **Grounding and Ranking**

Tom was getting ready for the track meet up. **His friends** were helping **him** by chasing after **him**. This wasn't good for **Tom**'s nervous though so **he** can faster. **He** finished his lap and turned around because **he** heard **someone** call his name. It was just **Steve** trying to hit **him** with one of the batons. Grow up **Steve**.

Textual Characters

[Tom_0, $\text{him}_{1(1)}$, $\text{him}_{1(2)}$, Tom_2, he_2, He_3, he_3, him_4],
[His friends_1],
[someone_3, $\text{Steve}_{4(1)}$, $\text{Steve}_{4(2)}$]

Visual Characters

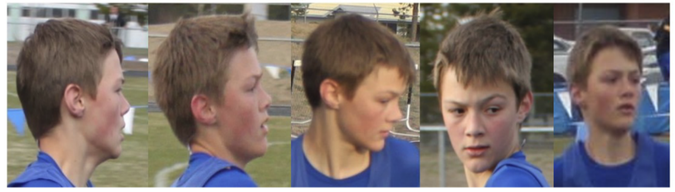


…

Multi-modal Characters and Their Ranking

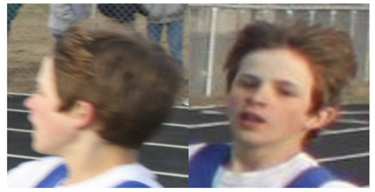
Rank: 1st (Length=13)

[Tom_0, $\text{him}_{1(1)}$, $\text{him}_{1(2)}$, Tom_2, he_2, He_3, he_3, him_4]



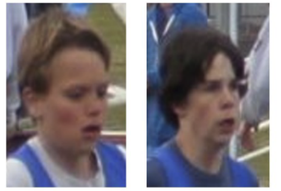
Rank: 2nd (Length=5)

[someone_3, $\text{Steve}_{4(1)}$, $\text{Steve}_{4(2)}$]



Rank: 3rd (Length=3)

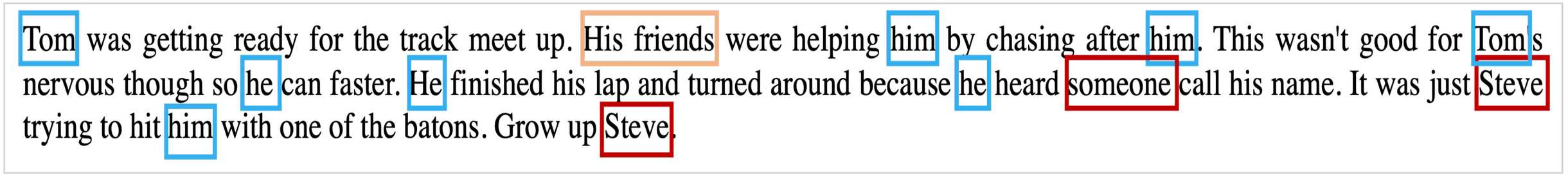
[His friends_1]



Character Detection and Co-reference

In text:

1. Use POS tagger and WordNet to detect the character mentions
2. Use pre-trained co-reference resolution tools (Span-BERT and NeuralCoref) to group the mentions from step 1 into co-reference chains



Character Detection and Co-reference

In images:

1. Obtaining face regions and features, we tried: (1) MTCNN + Inception ResNet and (2) MTCCN + CLIP vision encoder.

2. Employ k-means on the face features to obtain the co-reference chains.

Character Grounding

- We model character grounding as a **bipartite graph matching** problem
- First, compute the similarity between textual and visual chains. We propose two methods for this:
    1. **Distributional similarity:** Textual and visual mentions of the same character should have a similar distribution across the five images/sentences
    2. **CLIP-based similarity:** Use CLIP to compute average similarity of textual and visual mentions across the two chains
- Then apply the Hungarian algorithm to obtain the grounding results

Character Grounding

Tom was getting ready for the track meet up. His friends were helping him by chasing after him. This wasn't good for Tom s nervous though so he can faster. He finished his lap and turned around because he heard someone call his name. It was just Steve trying to hit him with one of the batons. Grow up Steve.

Importance Ranking

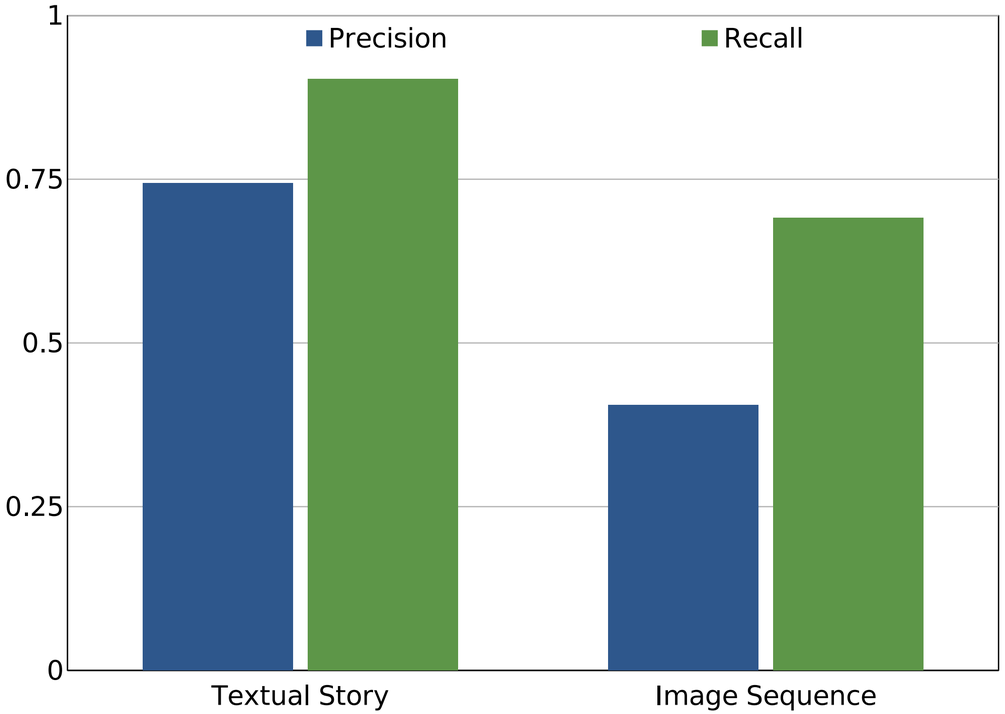
- Intuitively, the more important a character is, the more often it will be mentioned in the story
- Character frequency and importance are well correlated on the the gold-standard data
- We use **count-based importance ranking**

Story Type	Pearson's Correlation
Textual Stories	0.61
Visual Stories	0.55
Multi-modal Stories	0.62

Results

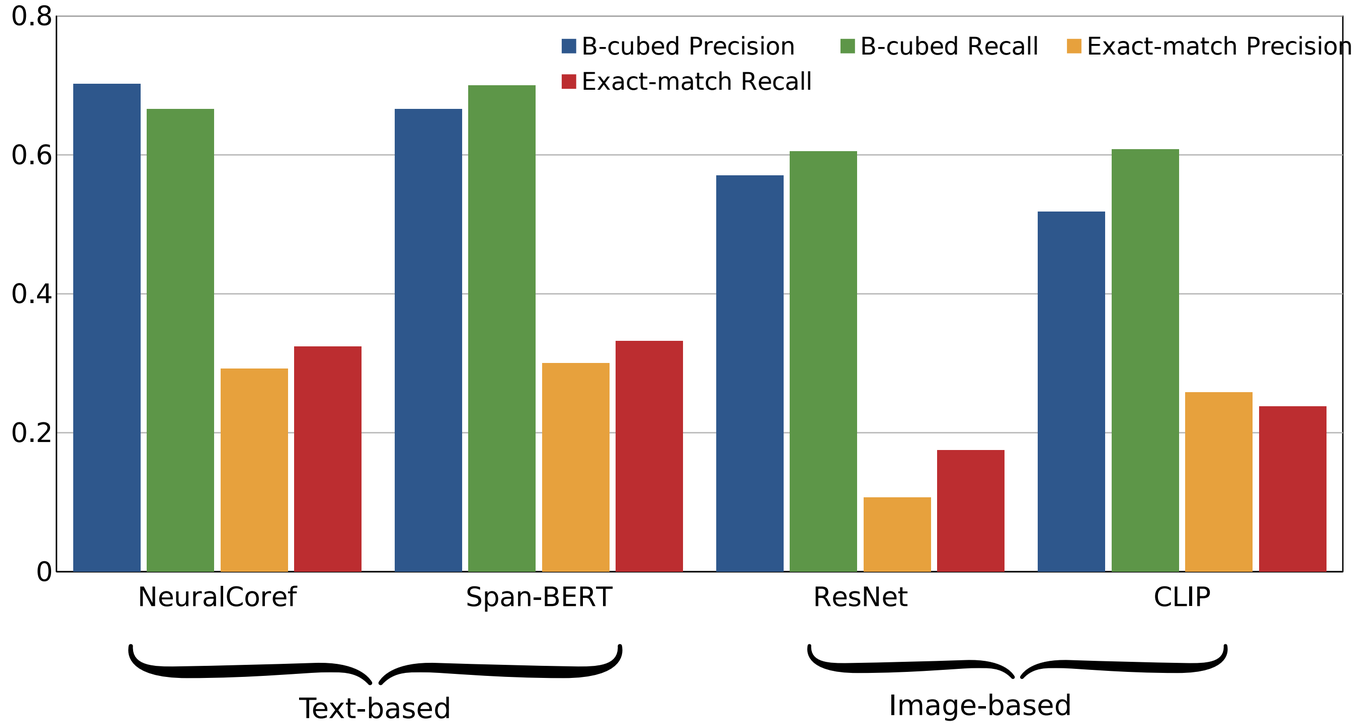
Character Detection:

1. **Text is less noisy:** better character detection performance in text than in images
2. Recall is higher than precision in images because of **background characters** unrelated to the story in the images



Results

Character Co-reference:

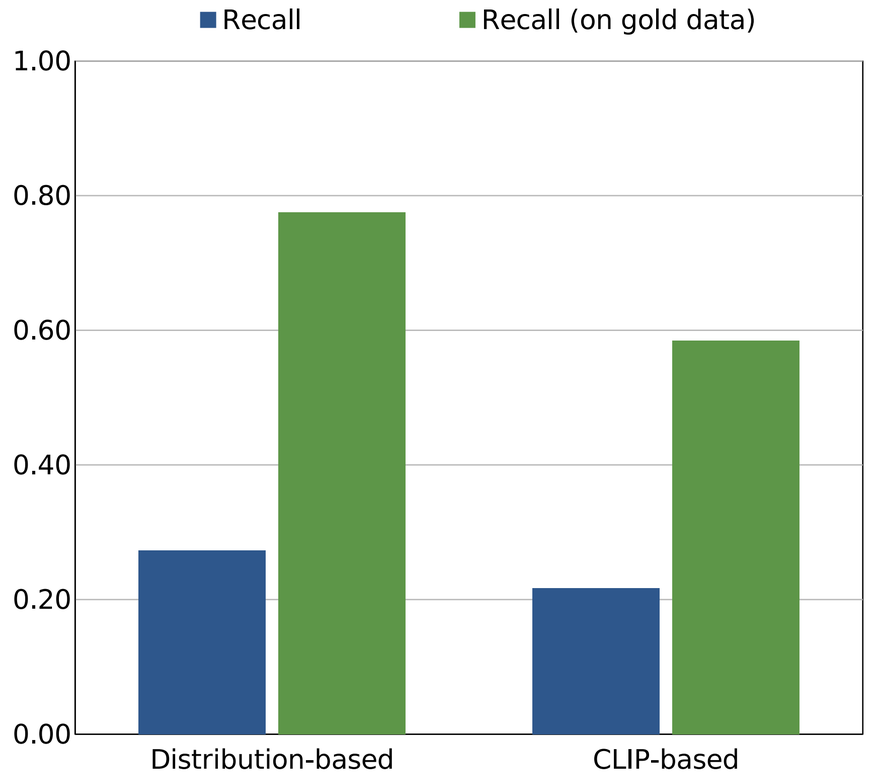


1. B-Cubed scores are higher than exact match: difficult to identify whole chains correctly
2. Co-reference resolution in text performs better than in images due to **redundant characters detected in images**

Results

Character Grounding:

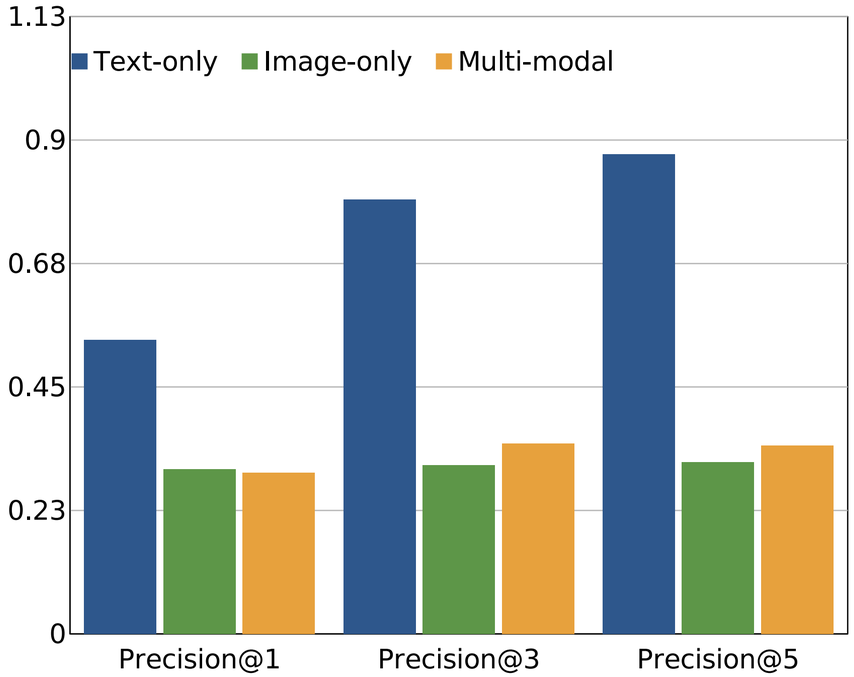
1. Performance is clearly better with gold-standard input: alignment algorithm works well, but is **sensitive to errors of previous components**
2. CLIP worse than distribution-based model: stories contain many **generic words** (e.g., *he, boy*) which CLIP hasn't been trained on



Results

Importance Ranking:

1. The text-only model performs best because there is **less noise in text than in image sequence**
2. The image-only model performs worse because the **performance of visual co-reference** resolution is not sufficiently high
3. The multi-modal model performs slightly better than the visual-only model



Interim Conclusions

- **VIST-Character dataset** extends the test set of VIST with co-reference chains for textual and visual character mentions and importance ranking
- This dataset can be used for **important character detection and grounding,** which requires both visual and textual co-reference resolution
- Two simple, unsupervised models for this task: one using **distributional similarity** and one based on CLIP

Part 2: Planning

From Analyzing to Generating Stories

- So far, we've talked about analyzing visual stories: detecting, grounding, and ranking their characters
- Ultimately, we want to use character information to generate better stories: **more grounded, more coherent, more interesting**
- We also want to make generation more **controllable**, e.g., generate stories of different lengths or with different main characters
- The standard way of achieving this in text generation is **planning:** select the content of a story and decide how to present it

From Analyzing to Generating Stories

- In this work, we will generate visual stories using a sequence of question-answer pairs, a **Blueprint**
- Previous work has used Blueprints for text summarization; we’re the first to use them in a multimodal context
- We turn images into **visual prefixes**, based on which a **pretrained language model** generates Blueprint annotations
- Stories are then generated iteratively in tandem with Blueprints

Images → Visual Prefix → LM → Blueprint (QA pairs) + Story

Automatic Blueprint Annotation

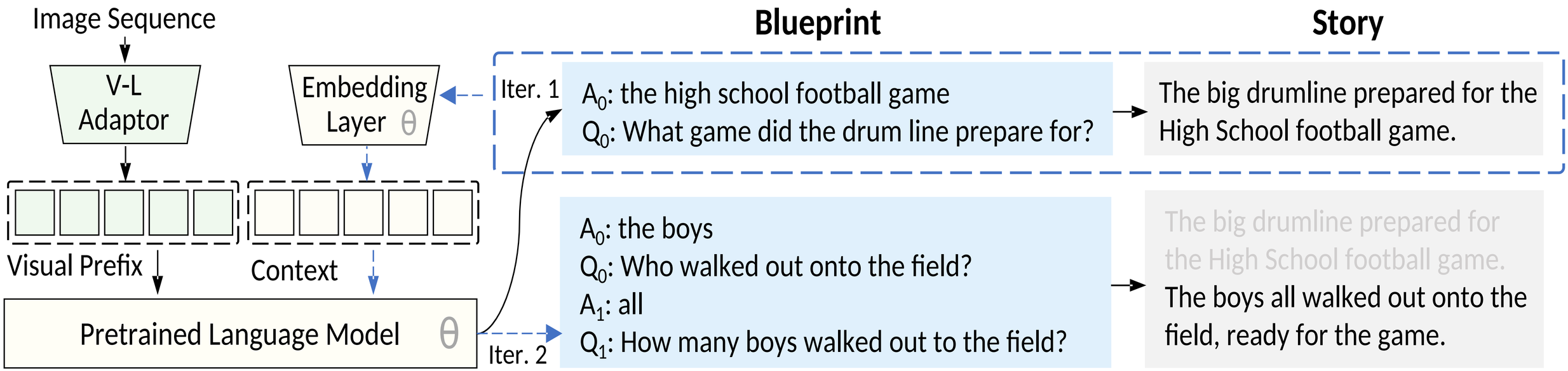
My Uncle Jack made us go to the Purdue game on Saturday. He went to Purdue and he thinks they are great no matter what else anyone thinks. We all had to sit in the bleachers and wear some ridiculous brown coats…

Answer Extraction using SpaCy

Q_0: Where did my uncle Jack made us go on Saturday?
A_0: the Purdue game
Q_1: When did my uncle Jack made us go to the Purdue game?
A_1: Saturday
Q_2: Where did we all sit?
A_2: the bleachers
Q_3: What did the fans of Purdue have to wear?
A_3: some ridiculous brown coats

Question generation using T5

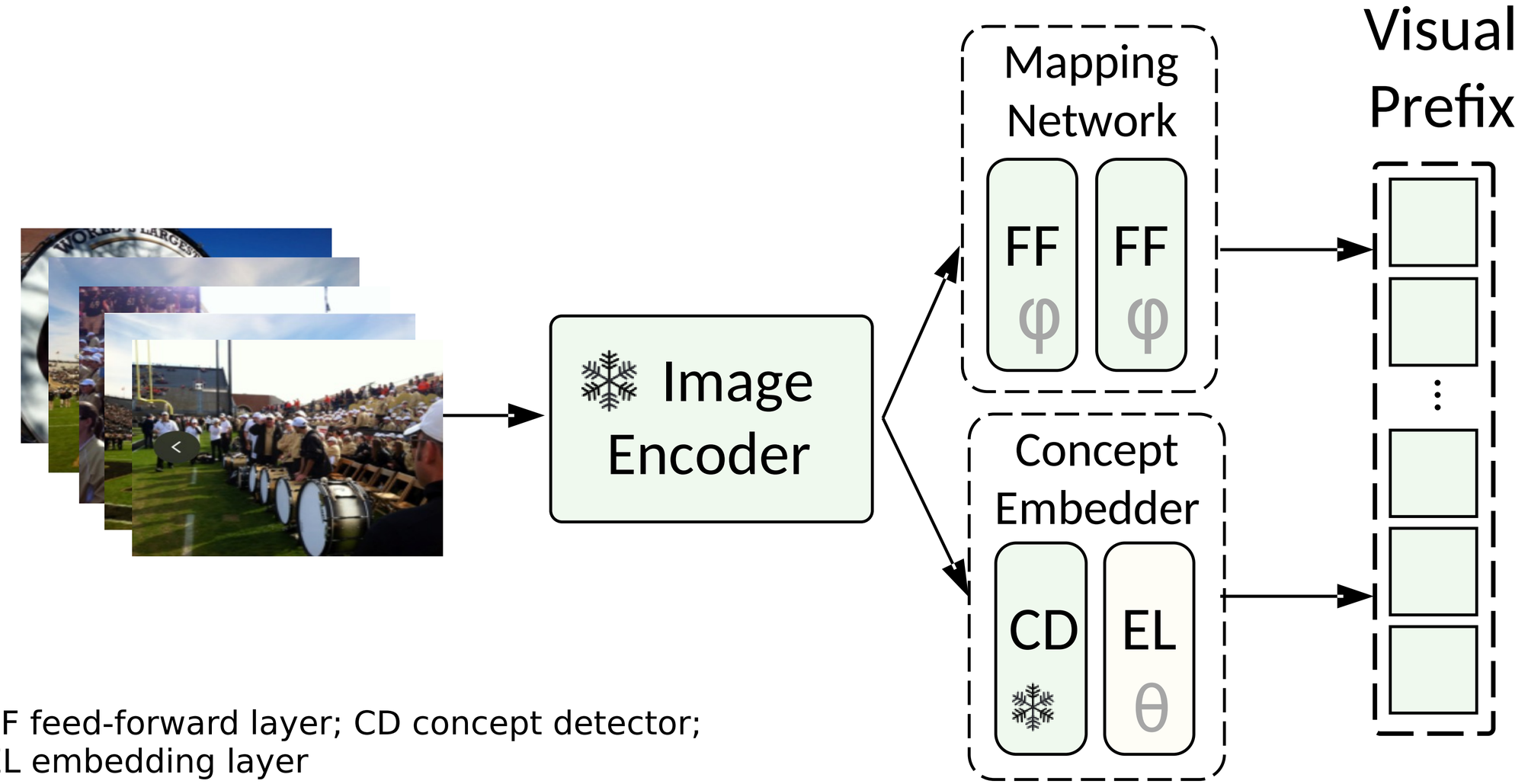
Blueprint VIST Model (iterative)



Blueprint VIST Model (iterative)

- Use a pretrained **question generation** model (based on T5) to turn VIST stories into Blueprints
- Train a linear model to map image sequences into **visual prefixes** (based on ResNet-152)
- Fine tune a pretrained model (BART) to **generate Blueprints and stories** from visual prefixes
- Generation proceeds sentence-by-sentence, where previous sentences are context for the current one: **iterative model**
- A **top-down model** that generates all sentences at once from the visual prefix performs worse

Visual Prefix Generation

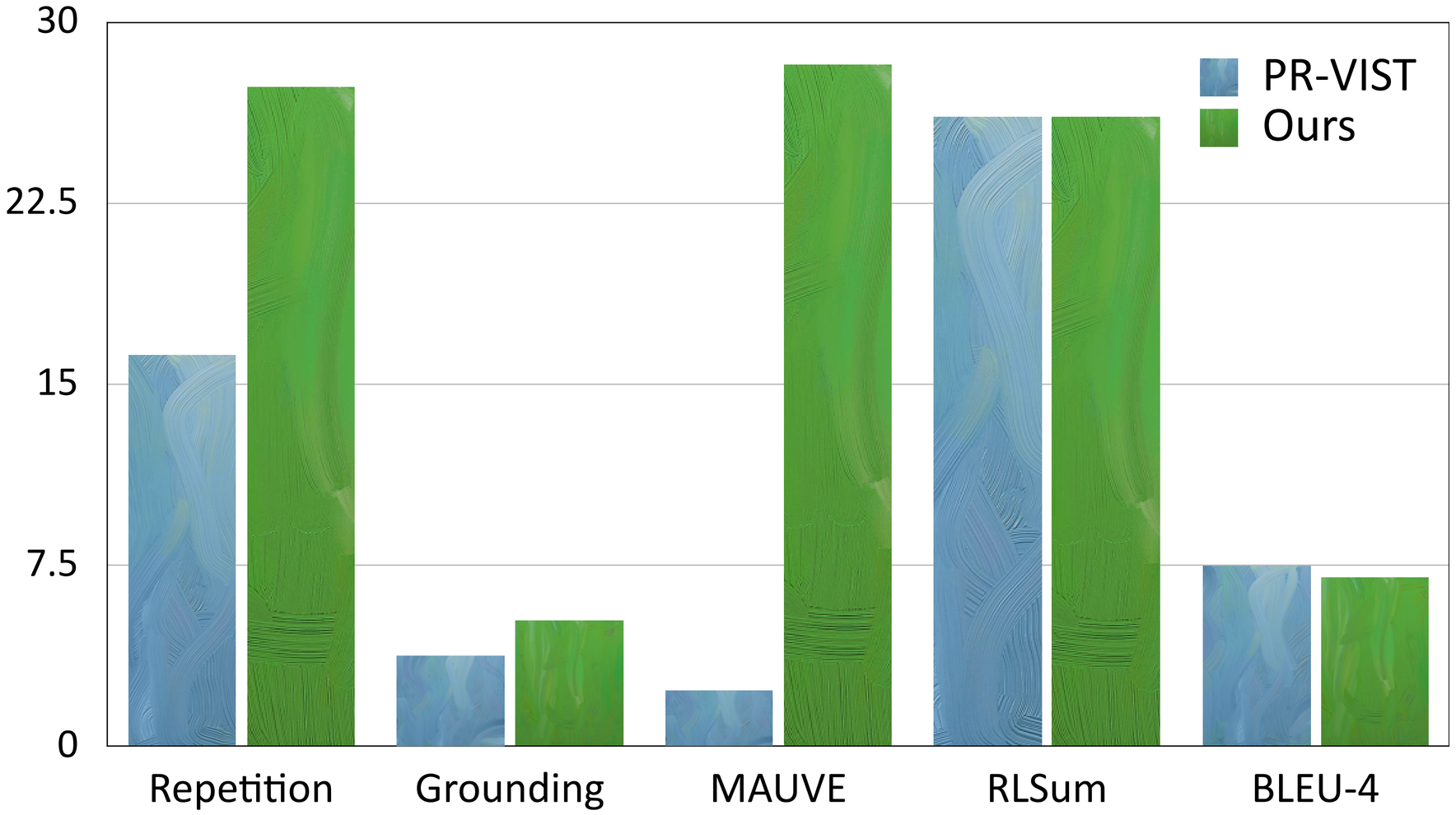


FF feed-forward layer; CD concept detector;
EL embedding layer

Evaluation

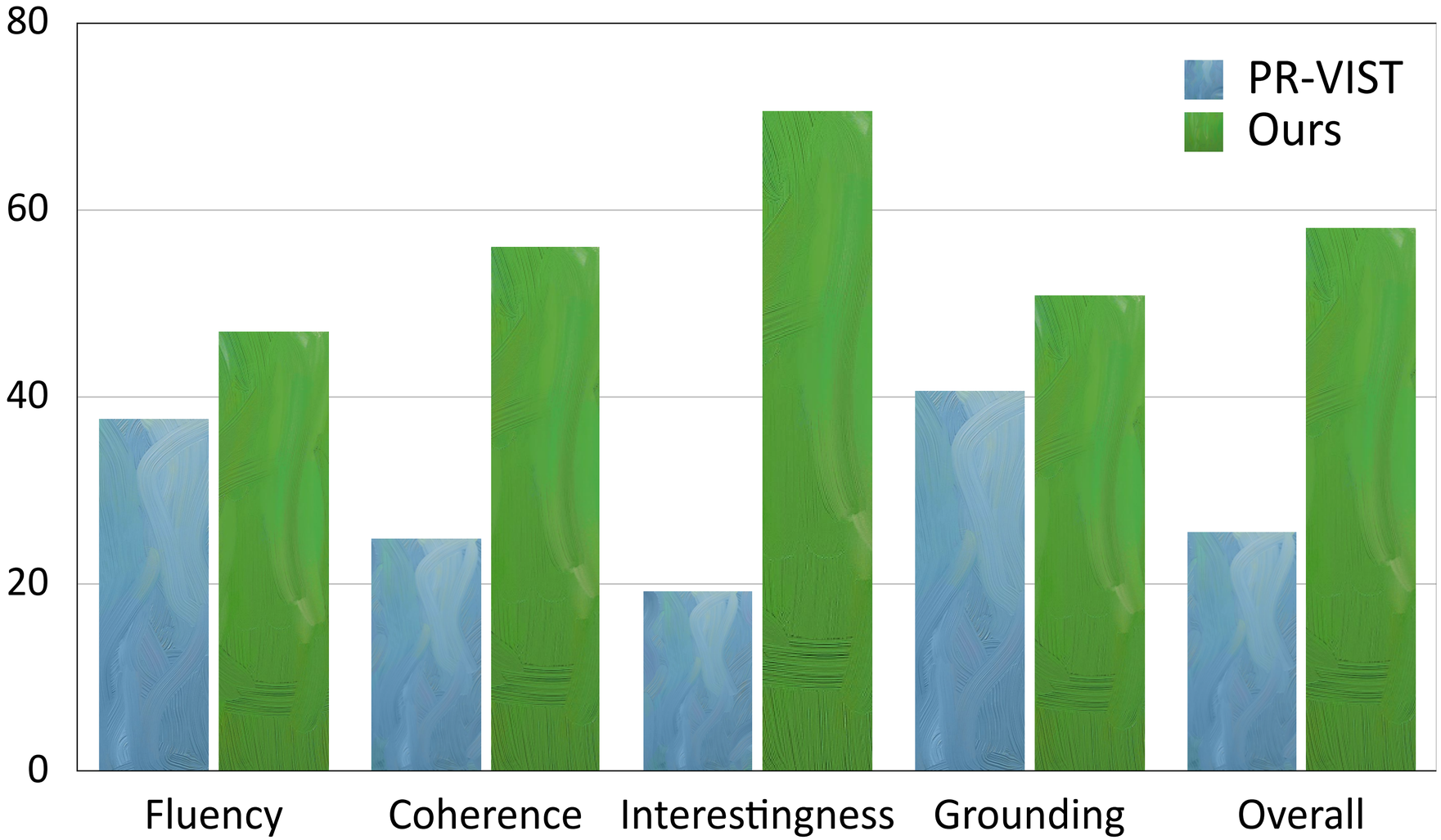
- Automatic evaluation using Blue, Rouge, Meteor is problematic – mostly measures fluency, and all modern LMs generate fluent output
- Also, a story may be interesting, coherent, and grounded in the images, but completely different from the reference
- We use automatic evaluation using **repetition and grounding, plus Mauve** for naturalness (similarity of distribution with human text)
- More importantly, we ask human judges to evaluate **fluency, coherence, interestingness** and **grounding**
- We compare to a range of SotA models for VIST and to GPT 3.5 (with and without Blueprint)

Automatic Evaluation



PR-VIST is a SotA model in the literature. It doesn’t use pretrained LMs; see paper for other comparisons

Human Evaluation



Example Output

PR-VIST: I bought a cart to the market with some food. It looked great in the market. There were so many of their enthusiasm. They even had a sale with their bumper. The market was sold out of fresh vegetables. the [location] car is very nice and beautiful.

VP-BART: I went to the market yesterday. There were many different kinds of fruits there. I bought a lot of them. They were very expensive. Afterward I went back home.

Iterative: Today we decided to take a small shopping trip to the Market. The market had so many wonderful things to choose from that we looked at and bought so many of them. There was an array of different fruits that I could hardly resist buying. There was also a huge array of various types of pepper. Finally the day had come to an end and we piled in the taxi back to head home.

GPT-3.5 + BP: On Saturday morning, I visited the local market. It had a lot of fruits and vegetables being sold. The atmosphere was friendly and buzzing with many vendors ready to help you. The fresh produce was the best thing about the market. You can always find friendly vendors that sell the most delicious produce. The key to finding the best bargains at the market is bargaining. The market is always a great place to shop for fresh fruits and vegetables.

Human: Shoppers arrived early for the market. There was a variety of goods sold. Some carried away large bundles. Every item was fresh and colorful. The spot was a favorite among produce shoppers.

Analysis of Results

- Pretrained language models can produce better stories than specialized models trained from scratch
- The visual prefix is an effective interface between image and text; we don’t meed multimodal models
- Blueprint model output is most grounded, by automatic and human evaluation
- An iterative planning strategy that generates sentence by sentence works best (see paper for comparison with top-down)
- GPT-3.5 struggles with Blueprints; they reduce its performance in most metrics

Conclusions

- Blueprint-based model generates stories that are **more coherent, interesting, and grounded** than existing methods
- Blueprints aid in **selecting key concepts** and **guiding narrative construction**
- Blueprints are **controllable:** generate longer or shorter stories (more/less iterations), emphasize entities or characters (filter Blueprint), etc.
- Blueprints are **interpretable** and could enable human-in-the-loop and personalized storytelling
- Future work: **combine Blueprints and characters!**

Additional Examples

CLIP Alignments

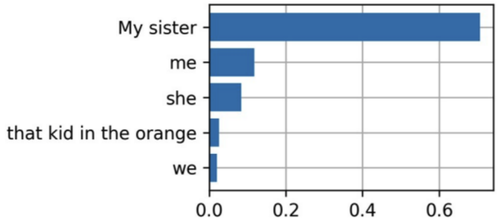
[Case 1] Input:



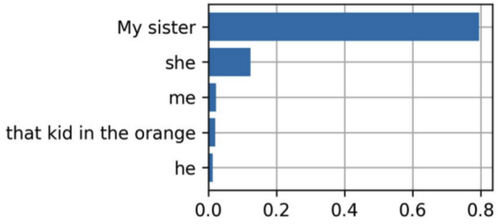


My sister caught me eating her cookies. Now we all have cookies. Don‘t tell mom, she is oblivious. Don’t tell that kid in the orange either, he dresses funny and we don‘t like him. Down with orange shirts!!

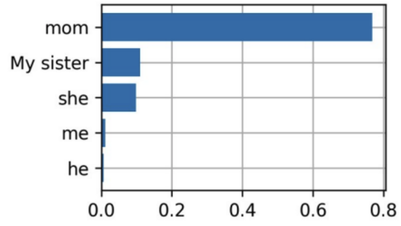
[Case 1] Some Representative Alignments:

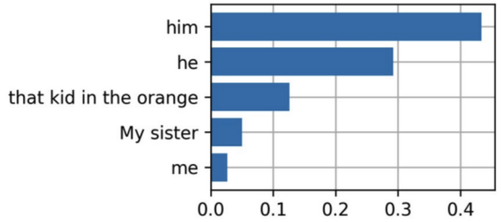
1.1 (✔)

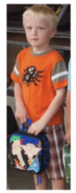
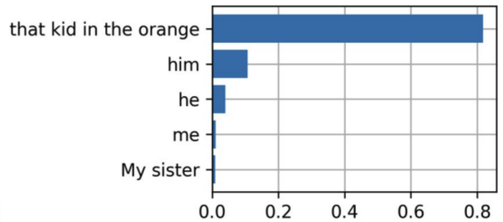
1.2(✘)

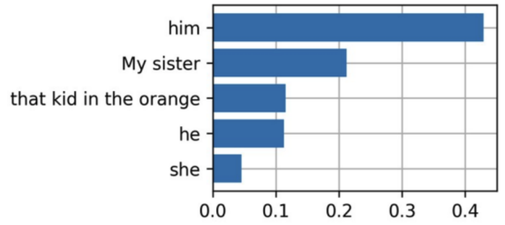
1.3(✔)

1.4 (✘)

1.5 (✔)

1.6 (✘)

CLIP Alignments

[Case 2] Input:



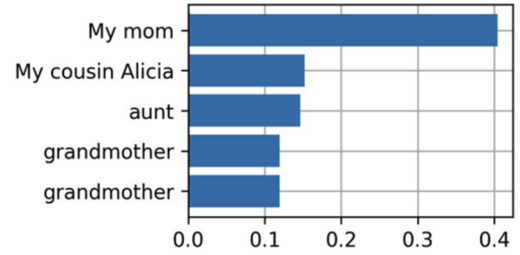


This month, my family had their annual reunion. My mom and grandmother made dinner for everyone. My great-grandmother, aunt, and grandmother took turns telling all kinds of stories. My cousin Alicia took pictures of everything! My dad got so tired, he fell asleep watching television

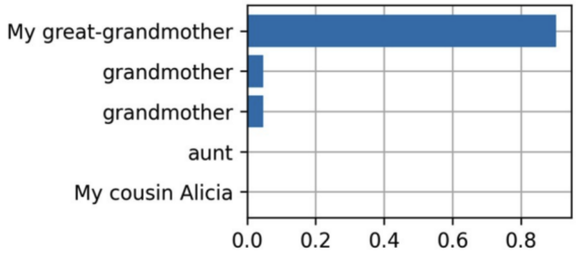
---

[Case 2] Some Representative Alignments:

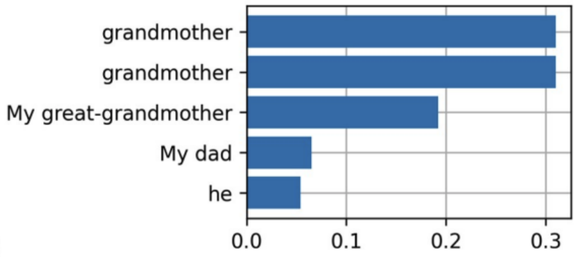
2.1 (✔)



2.2 (✔)



2.3 (✔)



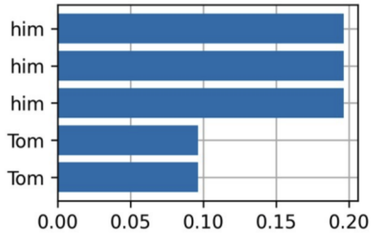
CLIP Alignments

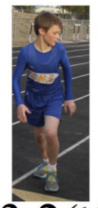
[Case 3] Input:

Tom was getting ready for the track meet up. His friends were helping him by chasing after him. This wasn't good for Toms nervous though so he can faster. He finished his lap and turned around because he heard some one call his name. It was just Steve trying to hit him with one of the batons. Grow up Steve.

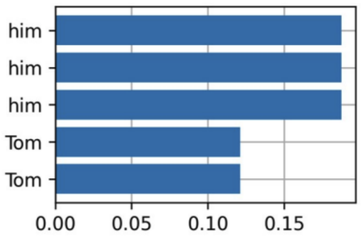
[Case 3] Some Representative Alignments:

3.1 (✘)





3.2(✘)



3.3 (✘)