Driving with Language

Driving with Language | APPRAS | Final Presentation | 26.02.2025

Fatih Mercan, Can Aydin, Justin Eduard Hulha, Jasmin Michelle Hulha, Jiaao Li, Jan Henri Christian Evard

Recap

Project Description

Project Focus

- In this project, we study how to connect vision-language models (VLMs) and autonomous driving systems

Project Goal

- Fine-tune a pre-trained VLM on driving scenario data to introduce the reasoning ability of LLMs to make decisions, and pursue generalizable and explainable driving behavior

Dataset and Benchmark Setup

LingoQA dataset consists of two complementary training datasets & evaluation dataset

Scenery Dataset

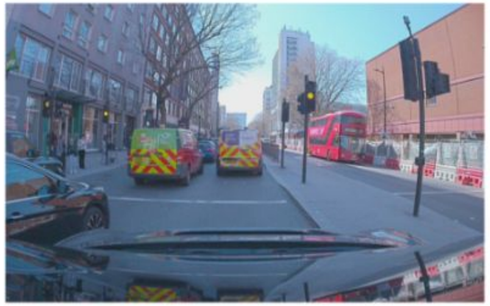
Q: Where is the pedestrian who is wearing a blue coat?

A: He is crossing the zebra from the right to the left.

Q: Can you describe the road you are on?

A: I'm on a T junction with a yellow boxed intersection with no cycle lane or road markings.

Action Dataset



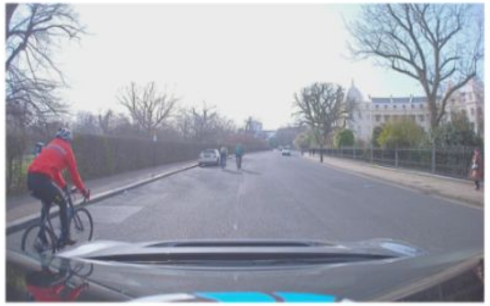
Q: How are the amber traffic lights impacting your actions?

A: The amber traffic lights are causing me to decelerate and be prepared to stop if they turn red.

Q: What is causing you to steer right?

A: The bus lane on the left is causing me to steer right to maintain a safe distance from it.

Evaluation Dataset



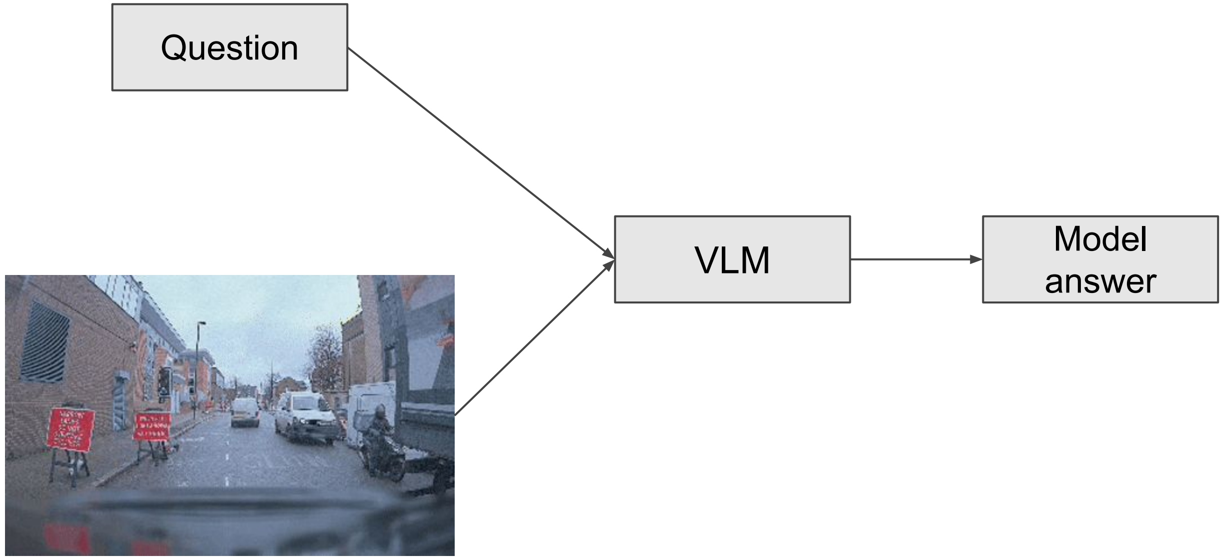
Q: Do you see any cyclists around? Where can you spot them?

A: Yes, there are two cyclists side by side ahead of me and a cyclist accelerating to my left.

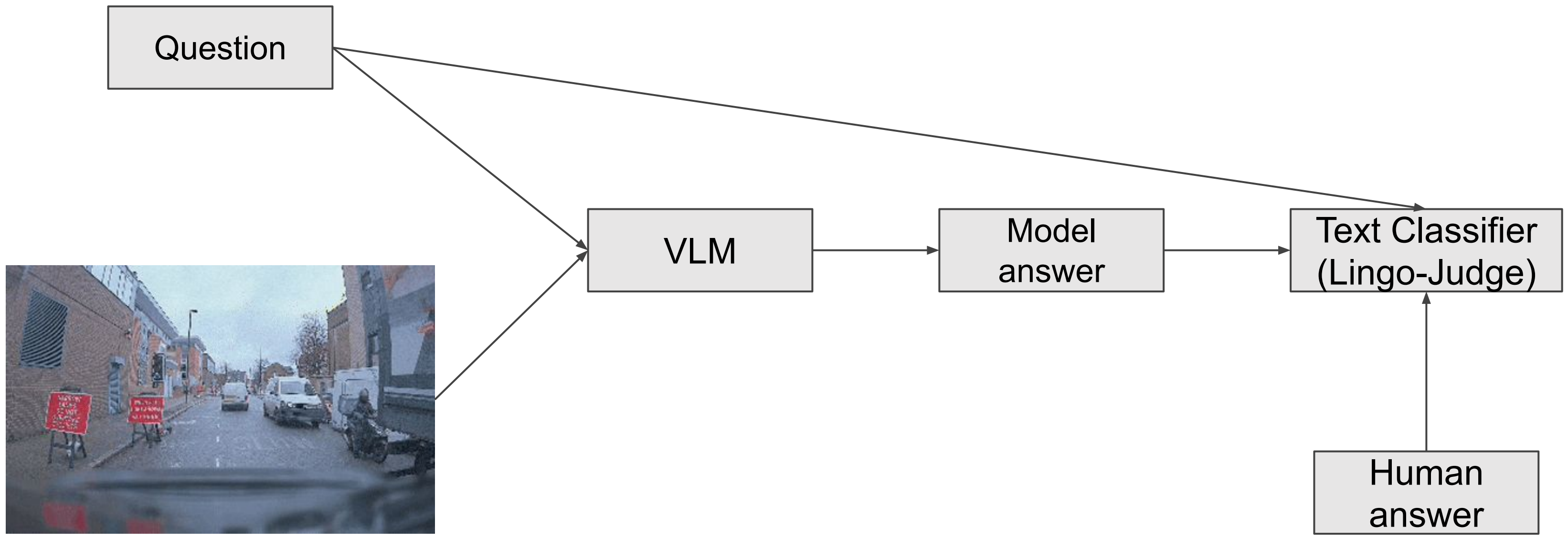
Q: Can you proceed straight ahead at this point? Explain why or why not.

A: No, there are pedestrians crossing in front of us so we must slow down.

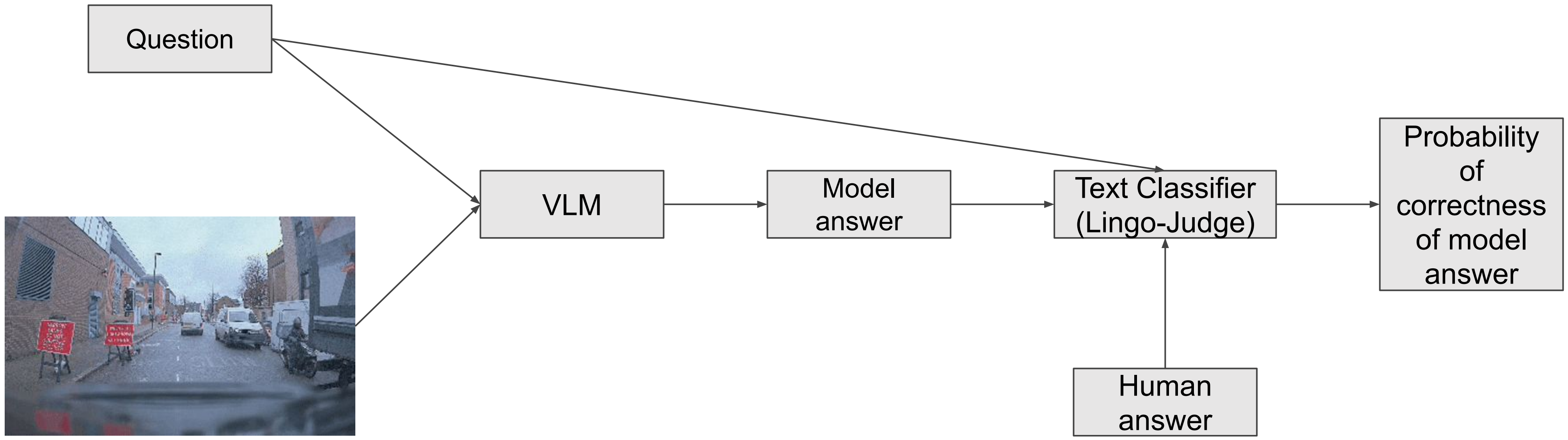
LingoQA Benchmark consists of evaluation metric & dataset



LingoQA Benchmark consists of evaluation metric & dataset



LingoQA Benchmark consists of evaluation metric & dataset



Inference & Benchmark Results

Inference Approaches - Recap

1. API:
   - Handles multiple queries simultaneously **with threading**
     → Good for multiple single requests

2. Chat:
   - Includes chat history
   - **No parallelization nor threading**

3. vLLM:
   - Supports **batch processing**
     → Advantageous for multiple inputs and dataset processing

Inference Performance

Scenario	API (with threading)			Chat			vLLM		
	Time	PTs	CTs	Time	PTs	CTs	Time	PTs	CTs
multiple_queries_multiple_images	3.61	3891.74	15.12	3.94	3891.74	15.07	1.15	3891.74	44.34
single_query_single_image	0.80	803.74	15.45	1.01	803.74	15.10	35.54	803.74	6.40
single_query_multiple_images	3.74	3891.74	15.09	3.96	3891.74	15.32	37.09	3891.74	7.40
multiple_queries_single_image	0.74	803.74	15.07	1.00	803.74	15.13	0.33	803.74	45.35

Table 1: Comparison of scenarios across API (4 threads), Chat, and vLLM for three key metrics on the first 500 samples of the LigoQA Evaluation dataset. The metrics are time in seconds, prompt tokens (PTs), and completion tokens (CTs). Single queries are averaged over 500 single-sample batches, while multiple queries use a single batch of 500 samples. Multiple images refers to five images per query, single image refers to one.

Fine-tuning Hyperparameters & Benchmark Results

Model	Dataset	Size	Image Res.	Cutoff Length	Batch Size	Quantization	Lingo Score
OpenaiAI GPT-4o	-	-	-	-	-	-	53.90
Qwen2-VL-7B-Instruct Base	-	-	-	-	-	-	54.20
Qwen2-VL-7B-Instruct #1	Action	1,000	262,144	2,048	2	-	55.30
Qwen2-VL-7B-Instruct #2	Scenery	1,000	262,144	2,048	2	-	52.90
Qwen2-VL-7B-Instruct #3	Action	25,000	262,144	2,048	2	-	55.30
Qwen2-VL-7B-Instruct #4	Scenery	25,000	262,144	2,048	2	-	**63.50**
Qwen2-VL-7B-Instruct #5	Action	10,000	700,000	8,192	1	4	60.09
Qwen2-VL-7B-Instruct #6	Action, Scenery	20,000	700,000	8,192	1	4	62.40

Table 3: Overview of multiple fine-tuning runs on the Qwen2-VL-7B-Instruct Model. For each run, 1,000 samples from the LingoQA Evaluation Dataset were used to generate predictions, which were then scored by the Lingo Judge Benchmark to yield a single overall score. The table lists the key hyperparameters for each run alongside the resulting score.

LingoQA Benchmark Results

Table 5: Evaluating vision-language models on LingoQA. The performance of existing vision-language models is far from human capability.

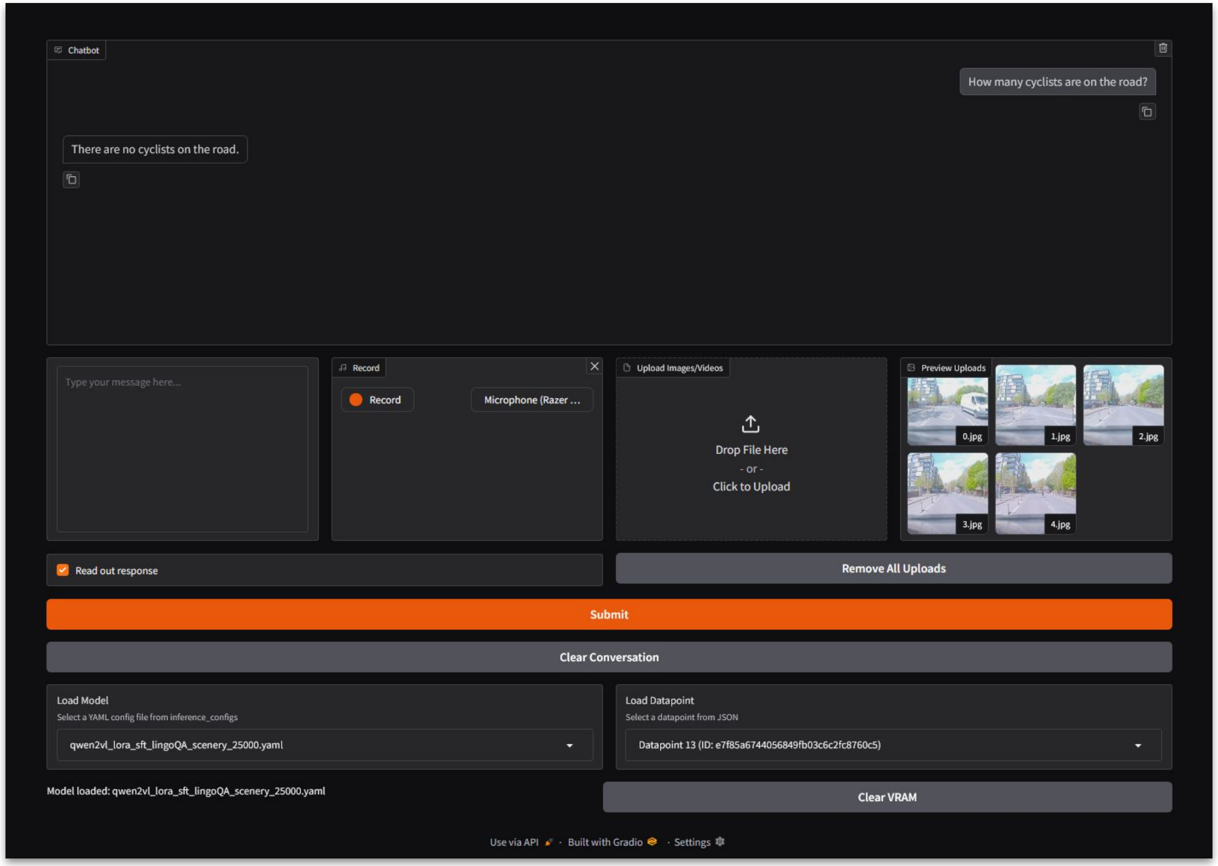
	Category	No. Frames	Human	Lingo-J	BLEU	METEOR	CIDEr
Human	*human study*	5	**93.3**	**96.6**	**81.04**	**52.92**	**361.77**
Human		1	-	81.8	10.64	15.01	64.45
LingoQA	*fine-tuned models*	5	**57.1**	**60.8**	**15.00**	**18.56**	**65.62**
LingoQA		1	-	57.0	14.21	18.40	59.46
LLaVA		1	-	59.0	12.5	18.5	57.0
BLIP-2		1	-	52.2	13.0	17.4	60.1
Vicuna-7B		0	-	38.8	10.1	15.2	51.0
GPT-4V	*zero-shot models*	5	**56.61**	**59.6**	6.30	12.35	**42.82**
LingoQA		5	-	33.6	**8.33**	**14.33**	39.16
LLaVA		1	38.97	49.4	4.23	8.38	38.39
FUYU		1	17.69	45.4	1.90	13.00	12.04

1) Source: “LingoQA: Visual Question Answering for Autonomous Driving”, Table 5

Live Demo with Gradio

Live Demo - Implementation Details

- Built using gradio
- **Multiple Input Options**
  - Custom text input with image upload
  - Video upload (processed at 1 frame/second)
  - LingoQA evaluation dataset samples
- **Model Selection**
  - Loading different fine-tuned model variants
- **Local Text-to-Speech**
  - Kokoro model by hexgrad
- **Local Speech-to-Text**
  - Whisper model by OpenAI



Future Work

Future Work and Research Directions

- **Dataset Metadata**
  Include Metadata that is not incorporated in LingoQA (Lidar, etc.)

- **Larger and/or more current models**
  Utilize larger or more up-to-date models for improved accuracy

- **Higher Image Resolution**
  Increase the resolution of images to capture finer details

- **Utilize Full LingoQA Dataset**
  Use the complete LingoQA dataset to enhance generalization, improve model performance and capture more edge cases

Conclusion

Conclusion

✅ **Successfully fine-tuned a VLM** (Qwen2-VL-7B) for autonomous driving visual QA

✅ Implemented **multiple inference methods** for optimized inference speed

✅ Created a **functional interface** for the fine-tuned models with local TTS and STT capabilities

✅ Evaluated and compared results to **LingoQA benchmark** using LingoJudge
🚀 Best fine-tuned model (Qwen2-VL-7B_Scenery-25k) achieves **SOTA** score of 63.50

References

- Sima, C. *et al.* (2023) ‘DriveLM: Driving with graph visual question answering’, *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/2312.14150.
- Marcu, A.-M. *et al.* (2023) ‘LingoQA: Visual question answering for autonomous driving’, *arXiv [cs.RO]*. Available at: http://arxiv.org/abs/2312.14115.
- Sun, P. et al. (2020) ‘Scalability in Perception for Autonomous Driving: Waymo Open Dataset’, arXiv [cs.CV]. Available at: https://arxiv.org/abs/1912.04838.
- Wang, P. et al. (2024) ‘Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution’, arXiv [cs.CV]. Available at: http://arxiv.org/abs/2409.12191.
- Liu, H. et al. (2023) ‘Visual Instruction Tuning’, arXiv [cs.CV]. Available at: http://arxiv.org/abs/2304.08485.
- Zhang, R. et al. (2023) ‘LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model’, arXiv [cs.CV]. Available at: https://arxiv.org/abs/2304.15010.
- Development Team (2025) ‘uv: A lightweight package manager for Python environments’, GitHub. Available at: https://github.com/astral-sh/uv.
- Hu, E.J. et al. (2021) ‘LoRA: Low-Rank Adaptation of Large Language Models’, arXiv [cs.CL]. Available at: https://arxiv.org/abs/2106.09685.
- Lin, C.-Y. (2004) ‘ROUGE: A Package for Automatic Evaluation of Summaries’, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81. Available at: https://www.aclweb.org/anthology/W04-1013.
- Papineni, K. et al. (2002) ‘BLEU: a Method for Automatic Evaluation of Machine Translation’, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318. Available at: https://www.aclweb.org/anthology/P02-1040.
- Banerjee, S. and Lavie, A. (2005) ‘METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments’, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72. Available at: https://www.aclweb.org/anthology/W05-0909.
- Vedantam, R., Lawrence Zitnick, C. and Parikh, D. (2015) ‘CIDEr: Consensus-based Image Description Evaluation’, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575. Available at: https://doi.org/10.1109/CVPR.2015.7299087.

Q&A

Appendix

LingoQA Dataset

	Scenarios	QA pairs	QA per scenario
Action	24.5k	267.8k	≈10.9
Scenery	3.5k	152.5k	≈43.6
Eval. Dataset	100	1000	10

Lingo-Judge

- DeBerta-V3 with a linear head on top of the class token output (Fine-tuned with LoRA)
- Initial dataset for fine-tuning:
  Q&A form evaluation dataset & model predictions (on evaluation dataset)
  Correctness target is labeled by human annotations
- Iterative improvement through active learning (correction wrong predictions & adding them into the training dataset)

Fine-tuning Setup

Model
- Qwen2_VL 7B Instruct
- Multimodal Vision Language Model
- No image resolution restrictions

Frameworks
- LLaMA-Factory Library
- Parameter Efficient Fine-tuning with LoRA and QLoRA

Key Hyperparameters
- Learning Rate: $1.0e^{-4}$
- Epochs: 3
- Optimizer: Adam
- dtype: bf16
- LR Scheduler: cosine

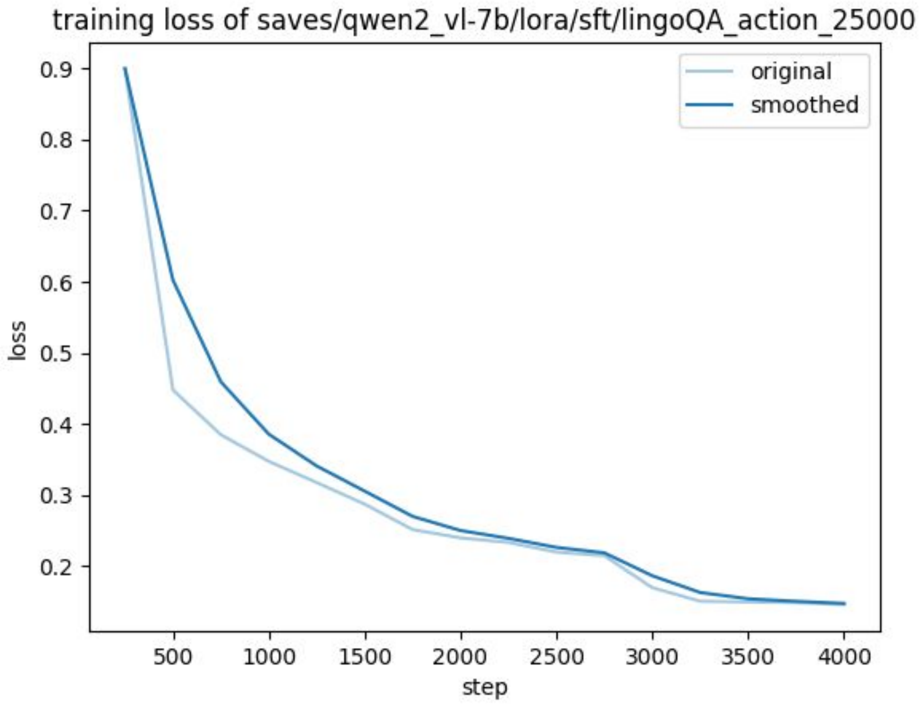
Lingo-Judge Code

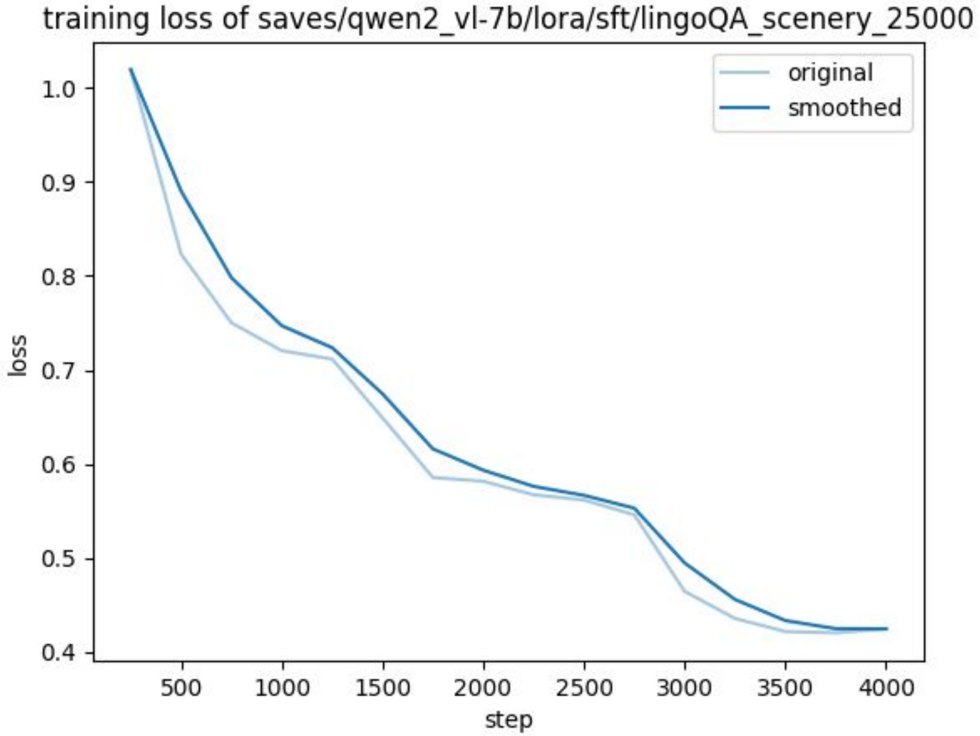
```python
class LingoJudge(nn.Module):
    """
    LingoJudge is a textual classifier that evaluates the truthfulness of an answer on the LingoQA benchmark.
    """
    def __init__(self, pretrained_model=LINGO_JUDGE):
        super().__init__()
        self.tokenizer = AutoTokenizer.from_pretrained(pretrained_model, use_fast=True)
        self.model = AutoModelForSequenceClassification.from_pretrained(pretrained_model).eval()
```

```python
@torch.inference_mode()
def forward(self, question: str, references: List[str], prediction: str):
    """
    Inference function for textual classifier with multiple reference answers.
    Args:
        question: Input question.
        references: List of references.
        prediction: Model prediction.
    Output:
        scores: Score indicating truthfulness.
    """
    device = next(self.parameters()).device
    texts = [
        f"{self.tokenizer.cls_token}\nQuestion: {question}\nAnswer: {a_gt}\nStudent: {prediction}"
        for a_gt in references
    ]

    encoded_input = self.tokenizer(texts, return_tensors='pt', padding=True, truncation=True, max_length=128)
    encoded_input = {k: v.to(device) for k, v in encoded_input.items()}
    output = self.model(**encoded_input)
    scores = output.logits.squeeze(-1)
    return scores
```

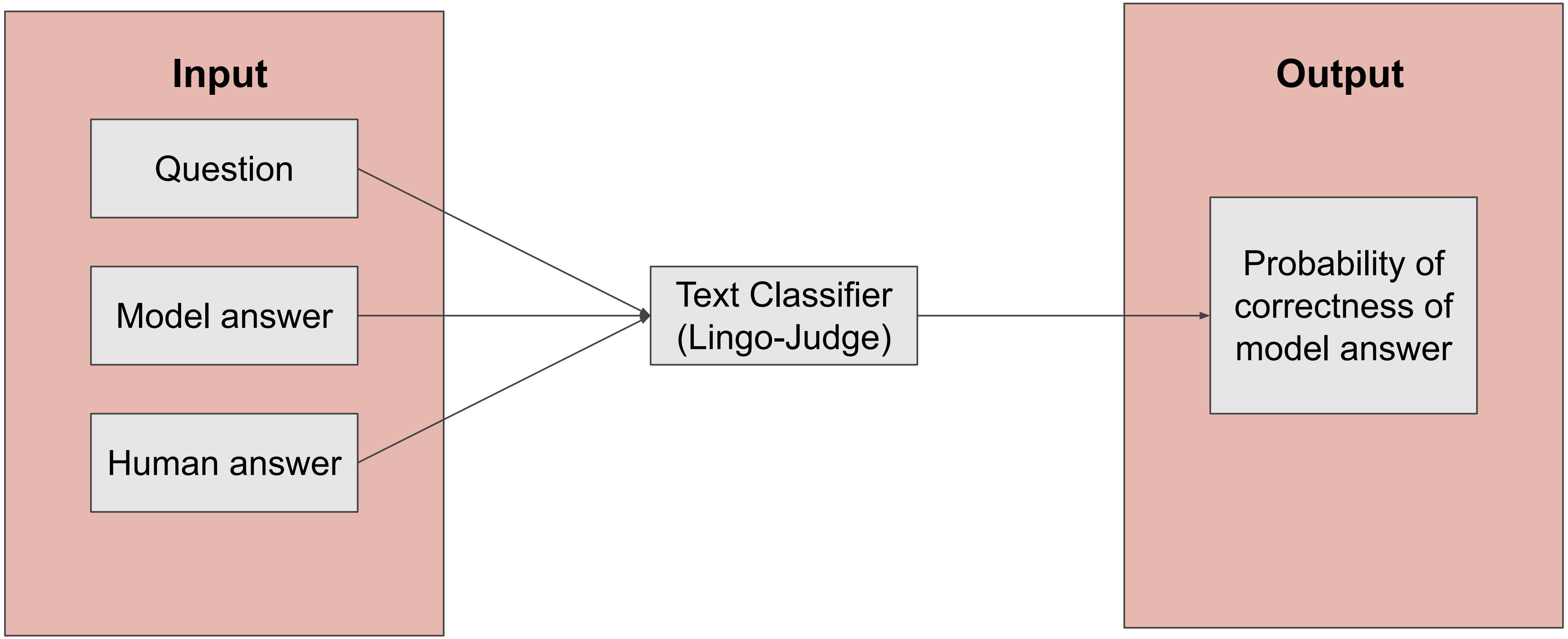
```python
def compute(self, questions: List[str], references: List[List[str]], predictions: List[str]):
    """
    Compute maximum classifier metric. For multiple reference answers, selects the highest one.
    Args:
        questions: List of input questions.
        references: List of lists, with multiple references per question supported.
        predictions: List of model predictions.
    Output:
        scores: Score indicating truthfulness.
    """
    max_scores = []

    for index, question in enumerate(questions):
        references_preprocessed = [self.preprocess(reference) for reference in references[index]]
        prediction_preprocessed = self.preprocess(predictions[index])
        scores = self.forward(question, references_preprocessed, prediction_preprocessed)
        max_score = [max(scores)]
        max_scores.extend(max_score)
    return torch.Tensor(max_scores)
```

Training Loss





LingoQA Benchmark consists of evaluation metric & dataset

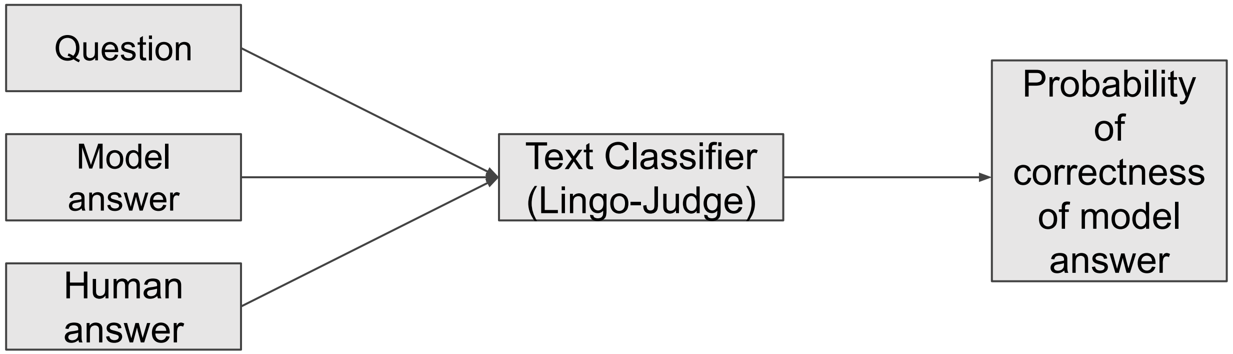


LingoQA Benchmark consists of evaluation metric & dataset

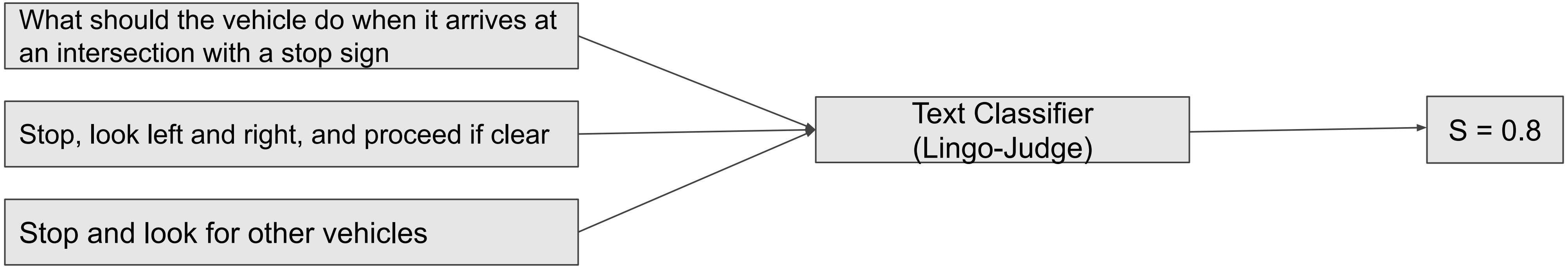
Question:
What should the car do when it arrives at an intersection with a stop sign

Model answer:
Stop, look left and right and proceed if clear

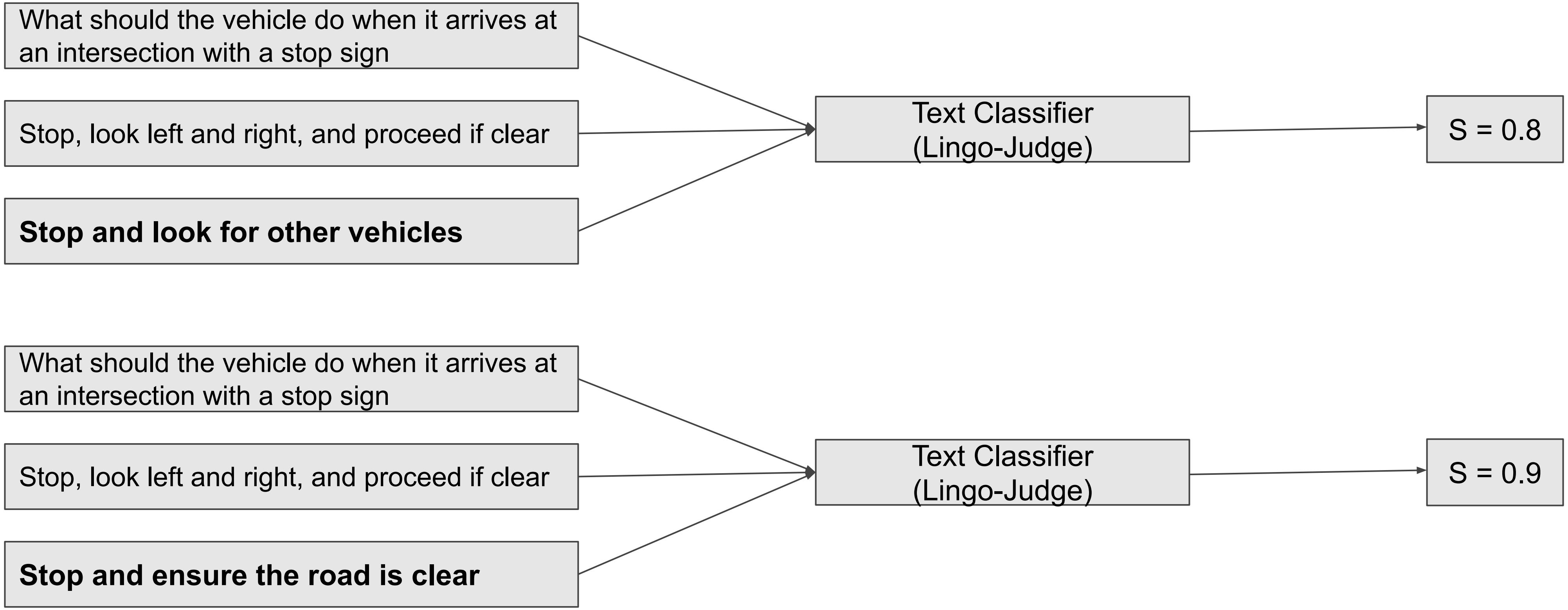
Human answer:
- Stop and look for other vehicles
- Stop and ensure the road is clear



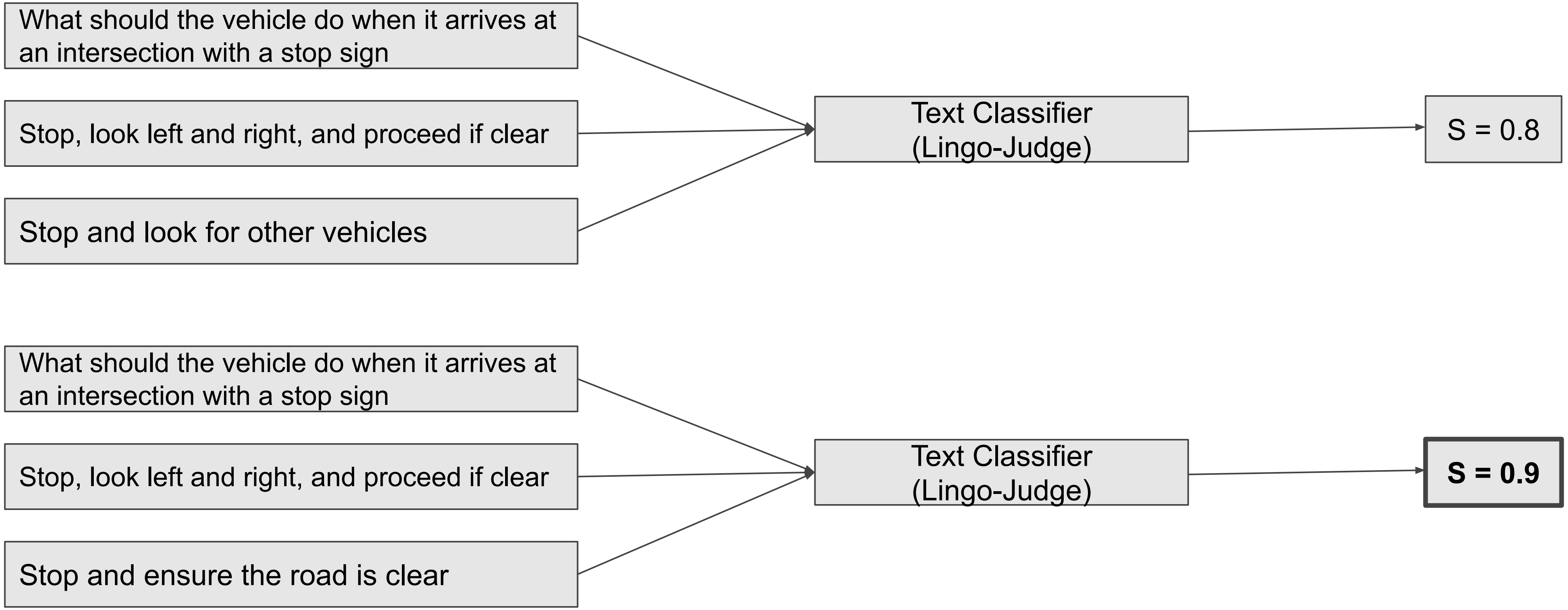
LingoQA Benchmark consists of evaluation metric & dataset



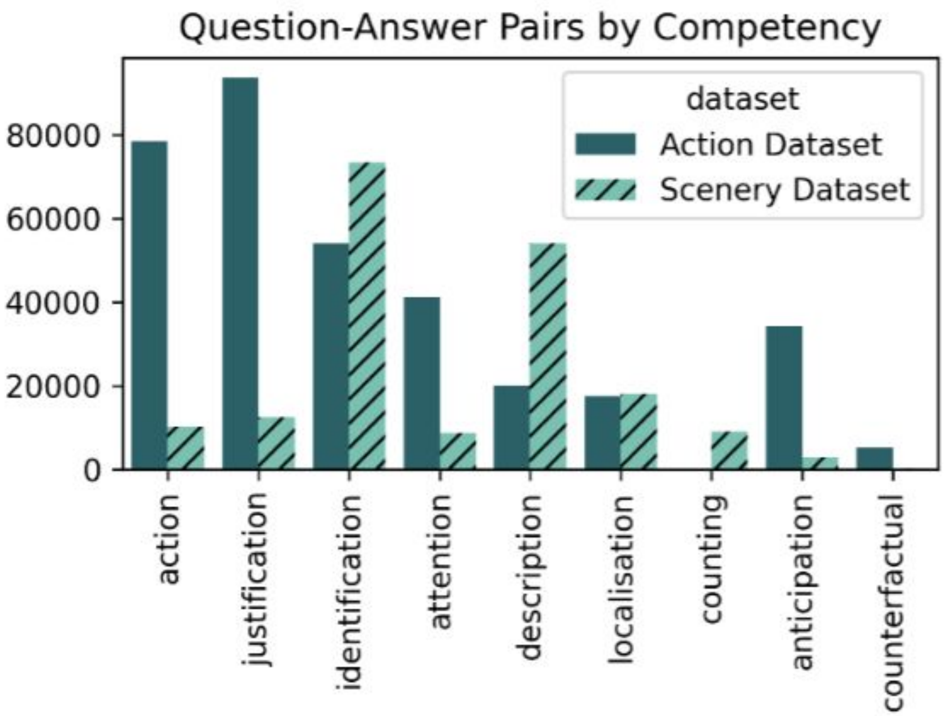
LingoQA Benchmark consists of evaluation metric & dataset

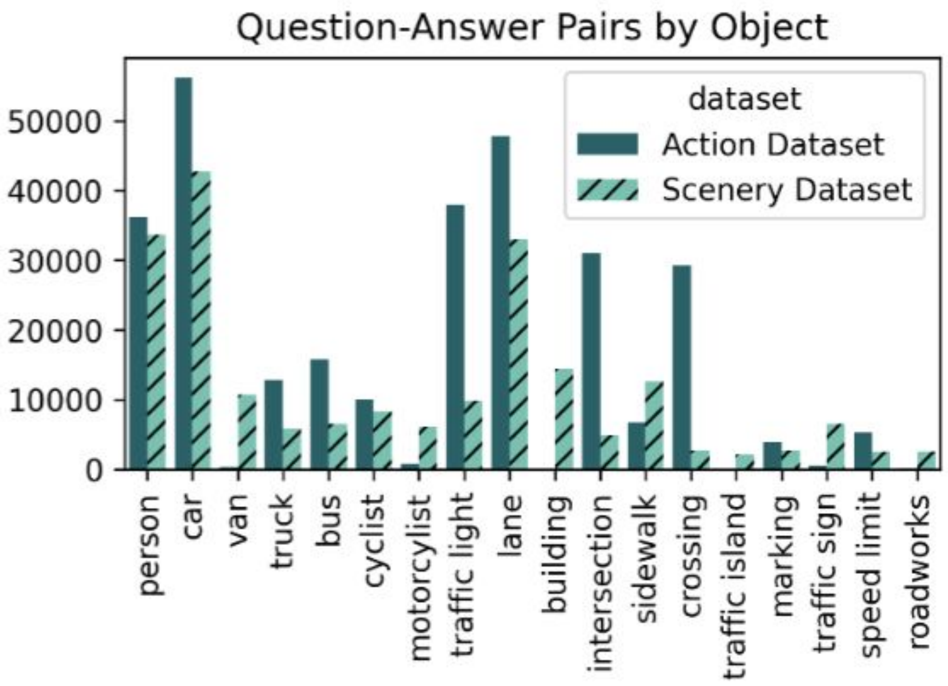


LingoQA Benchmark consists of evaluation metric & dataset



LingoQA Training Dataset Distribution





LingoQA Evaluation Dataset Distribution