Grounding Natural Language Processing in Geodata

Raphael Schumann
PhD Student in Computational Linguistics
Heidelberg University, Germany

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Geodata

OpenStreetMap

- Natural Language Interface
- Generating Navigation Instructions

Google Street View

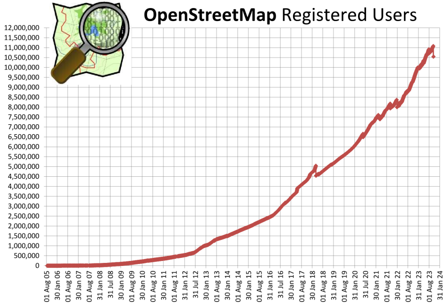
- Vision and Language Navigation
- Embodied AI

Introduction OpenStreetMap

OpenStreetMap (OSM) is a free and **open geographic database** started in 2006.

Applications:

- Map Rendering
- Route Planning
- Geocoding
- Geographic Information Analysis
- and many, many more...



Also serves as a data source for geospatial services of Facebook, Amazon or Apple.

Introduction OpenStreetMap

OpenStreetMap is one gigantic XML file called *planet.osm* (1.8 TB, gzip: 134 GB).

There are three basic components:

Nodes:

- point in space defined by latitude and longitude

Ways:

- composed of multiple nodes
- used for: roads, rivers, railway lines, building outlines

Relations:

- group of nodes, ways and other relations
- used for: bus routes, major highways, multipolygons

Any element can be tagged with **key-value pairs** that specify semantics and metadata.

Overpass Query Language

The Overpass Query Language (OverpassQL) allows to extract elements from OSM:

```
{{geocodeArea:"Troms"}}->.searchArea;
(
    node["natural"="peak"](area.searchArea);
)->.peaks;
way["highway"="cycleway"](area.searchArea)(around.peaks:500);
out center;
```

Desired elements are specified by their **type, tags and spatial relationship**.

OverpassQL offers **extensive syntax features** to query, combine and filter geodata.

Motivation Natural Language Interface

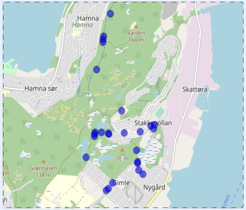
Natural Language Input:

Bike lanes 500 meters around the top of a hill or mountain in Troms.

Overpass Query Language:

```
{{geocodeArea:"Troms"}}->.searchArea;
(
    node["natural"="peak"](area.searchArea);
)->.peaks;
way["highway"="cycleway"](area.searchArea)(around.peaks:500);
out center;
```

Query Execution Results:




```
{"elements": [
 {"type":    "way",
   "id":        4013565,
    "center":  {"lat": 69.6900047,
                  "lon": 18.9971581},
    "tags":    {"highway": "cycleway",
                  "surface": "asphalt",
                  "segregated": "no"},
  },
    ...
```

Natural Language Interface

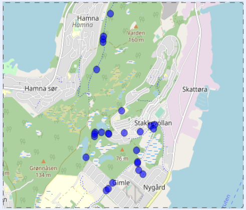
Natural Language Input:

Bike lanes 500 meters around the top of a hill or mountain in Troms.

Overpass Query Language:

```
{{geocodeArea:"Troms"}}->.searchArea;
(
    node["natural"="peak"](area.searchArea);
)->.peaks;
way["highway"="cycleway"](area.searchArea)(around.peaks:500);
out center;
```

Query Execution Results:

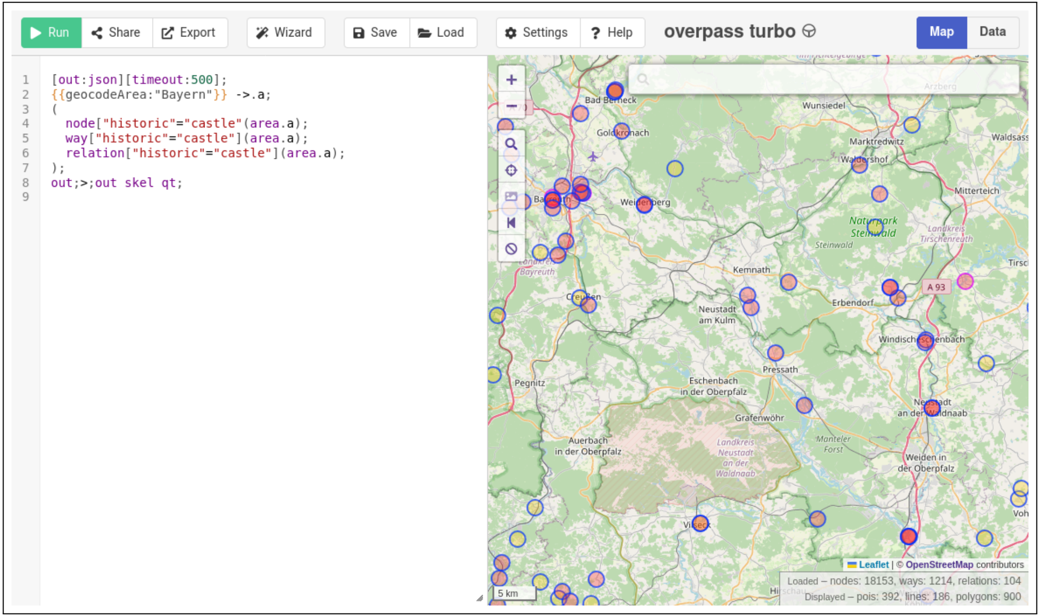


{"elements": [
 {"type": "way",
  "id": 4013565,
  "center": {"lat": 69.6900047,
    "lon": 18.9971581},
  "tags": {"highway": "cycleway",
    "surface": "asphalt",
    "segregated": "no"},
 },
 ...

Text-to-OverpassQL:
A Natural Language Interface for Complex Geodata Querying of OpenStreetMap

Michael Staniek, Raphael Schumann, Maike Züfle, Stefan Riezler *TACL 2024*

Data Acquisition: https://overpass-turbo.eu



Data Acquisition: Annotation

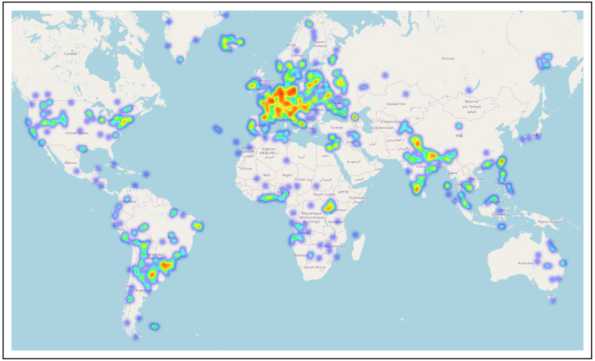
We show the Overpass queries to trained students and ask them to write text inputs.

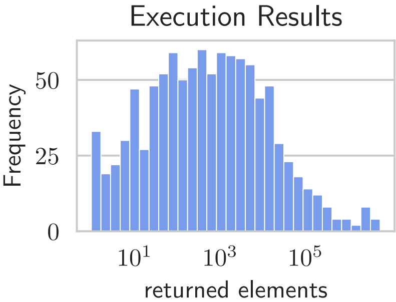
This has several advantages in comparison to write the queries from scratch:

- Easier to train students to understand Overpass than to write it
- The queries were written to satisfy legitimate information need
- High feature coverage because written by proficient users

OverpassNL Dataset: Statistics

8.5k Overpass queries with annotated natural language input

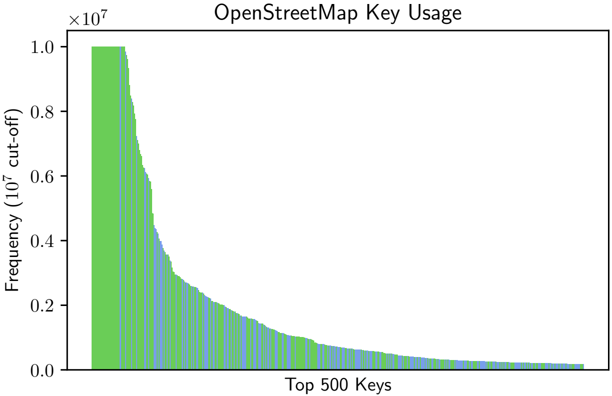




Large location coverage and up to 10^7 elements returned when executing a query.

OverpassNL Dataset: Statistics

- 1k unique keys
- 90% of key usage in OSM
- 3.9k unique values
- 4.9k unique key-value pairs



OverpassNL Dataset: Comparison

Dataset	**Query Language**	**Named Identifiers**		**Extractable Elements**		**Results**
		total	per database	total	per database	per query
Spider	SQL	4,669 & 876	58 & 5	1.6M	9.6k	30
OverpassNL	OverpassQL	1,046	1,046	9B	9B	10k

In OverpassNL, each query can utilize any key-value pair (named identifier) and retrieve elements from the entire OSM database.

In Text-to-SQL, each individual database is muss smaller which reduces the number of relevant table/column names (named identifier) significantly.

Text-to-OverpassQL: Results

Model	String Match	Execution Accuracy
OverpassT5	22.0	36.7
GPT-3	19.5	34.1
GPT-4	**23.4**	**40.4**

Evaluated on 1,000 dev instances.

Finetuned T5 Model

- 6.5k training instances
- +8k from code comments
- ByT5 is better than CodeT5

In-Context Learned GPT Models

- 5-Shot In-Context Learning
- retrieval by text similarity
- sBERT cosine is better than BLEU

Text-to-OverpassQL: Results

Model	chrF	KVS	TreeS	Execution Accuracy
OverpassT5	75.5	66.0	73.7	36.7
GPT-4	**75.7**	**69.9**	**74.0**	**40.4**

- **chrF**: Character F-score [Popović 2015]
- **KVS**: Key-Value Similarity - Normalized overlap of key-value
- **TreeS**: XML-Tree Similarity - Subtree overlap of query in XML representation

Text-to-OverpassQL: Results

Model	chrF	KVS	TreeS	Execution Accuracy
OverpassT5	75.5	66.0	73.7	36.7
GPT-4	**75.7**	**69.9**	**74.0**	**40.4**

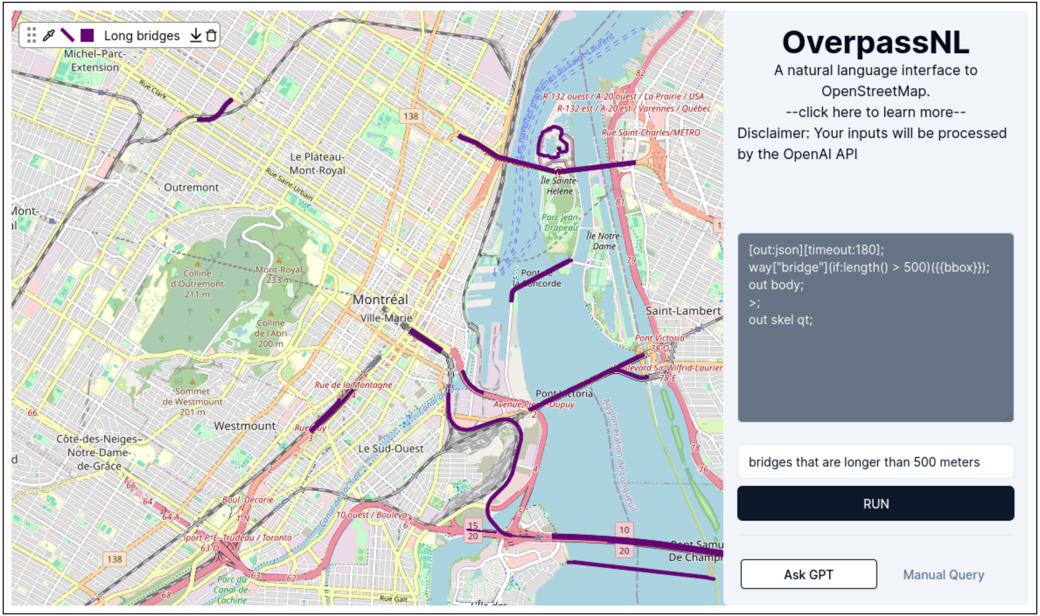
- **chrF**: Character F-score [Popović 2015]
- **KVS**: Key-Value Similarity - Normalized overlap of key-value
- **TreeS**: XML-Tree Similarity - Subtree overlap of query in XML representation

Text-to-OverpassQL: Self-Refinement

Model	# Syntax Errors	Execution Accuracy
GPT-4	24	40.4
refine w/o feedback	31	39.6
refine with feedback	26	**41.4**

Feedback is the error message or a sample of the returned results.

Demo: https://overpassnl.schumann.pub



Generating Navigation Instructions

Generating Human-like Navigation Instructions

If you ask a person on the street for directions, they give you a different type of navigation instructions than a GPS system or routing app.

GPS system:

- turn-by-turn with exact distances
- street names

Human-like:

- colloquially
- landmark based
- easier to memorize

Generating Human-like Navigation Instructions

Existing systems that generate landmark-based navigation instructions:

- rule-based landmark selection
- hand-crafted salience scores
- template-based text generation

→ **End-to-end system that learns to select landmarks and generate instructions**

Generating Landmark Navigation Instructions from Maps as a Graph-to-Text Problem
Raphael Schumann and Stefan Riezler *ACL 2021*

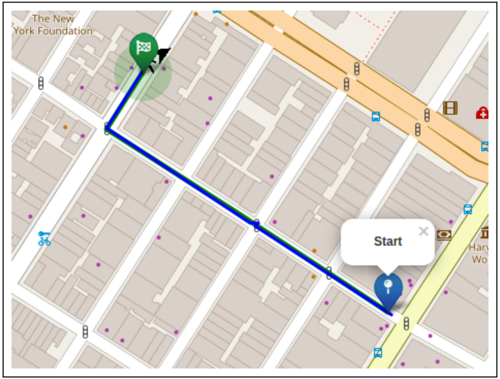
Navigation Instructions - Data Collection

Turn left at the light with Starbucks on the left corner. Go through the next light with the park on your left. Stop at the next T-intersection with the park on the near left corner and a hospital on the far left corner.

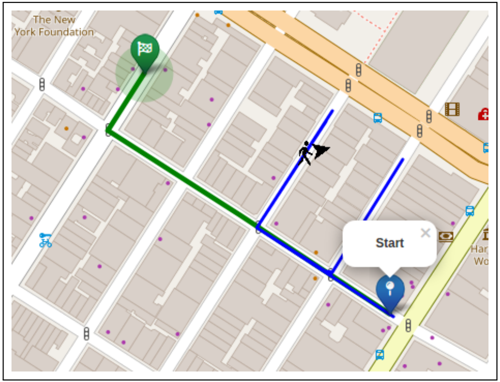
Submit

Navigation Instructions - Data Collection

Navigation Instructions - Data Collection



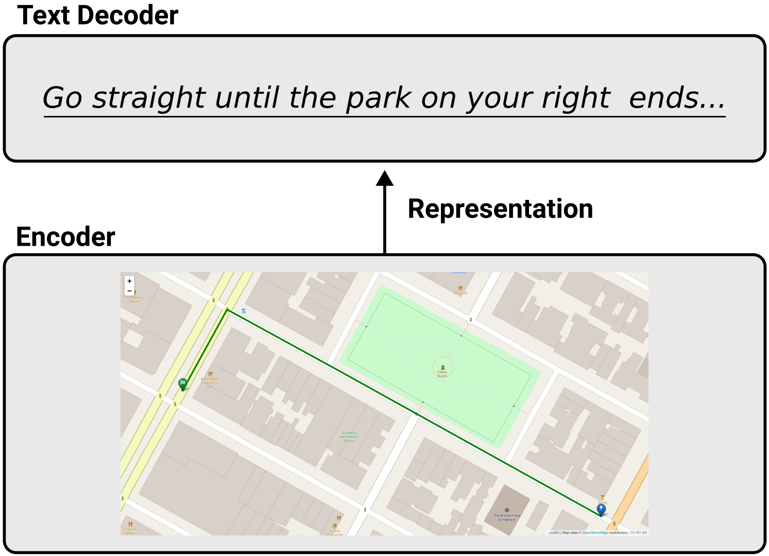
verified



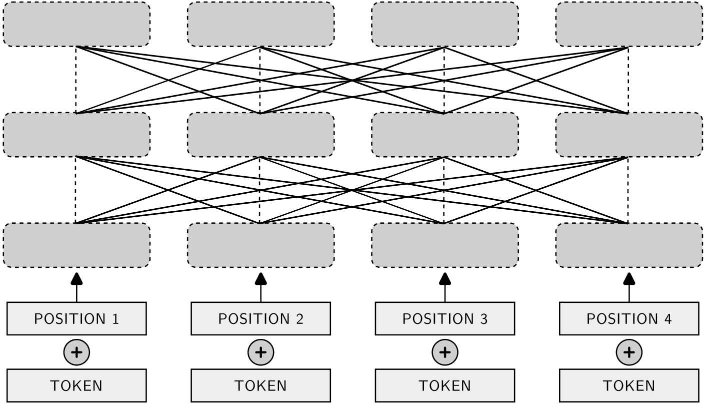
failed

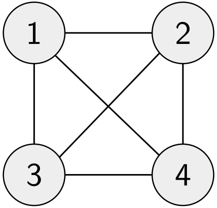
- Paid \$0.35 (+\$0.25 bonus) per writing task and \$0.20 (+\$0.15 bonus) per navigation task
- Collected **7.5k verified navigation instructions** for around \$10k in total

Route Representation



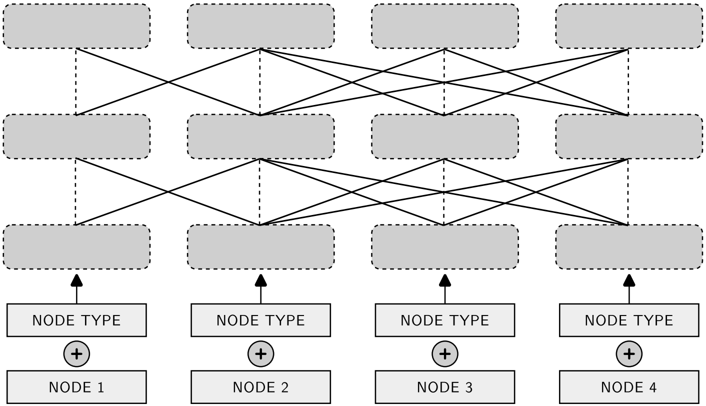
Graph Encoder

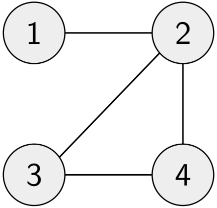




Vanilla Transformer Encoder [Vaswani et al. 2017]

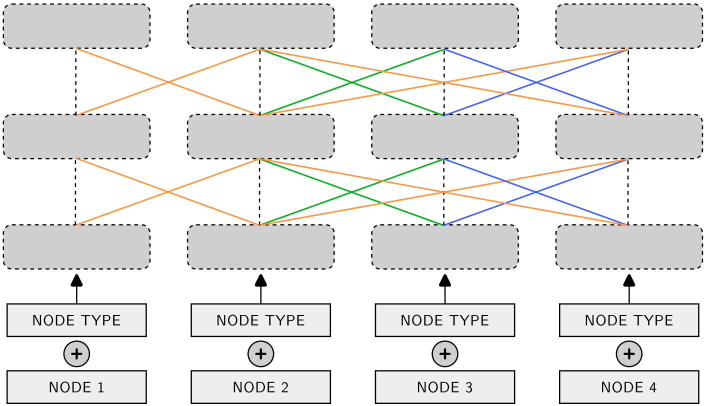
Graph Encoder

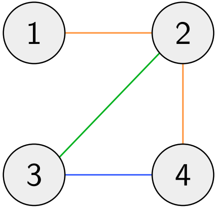




Graph Attention Networks [Veličković et al. 2018]

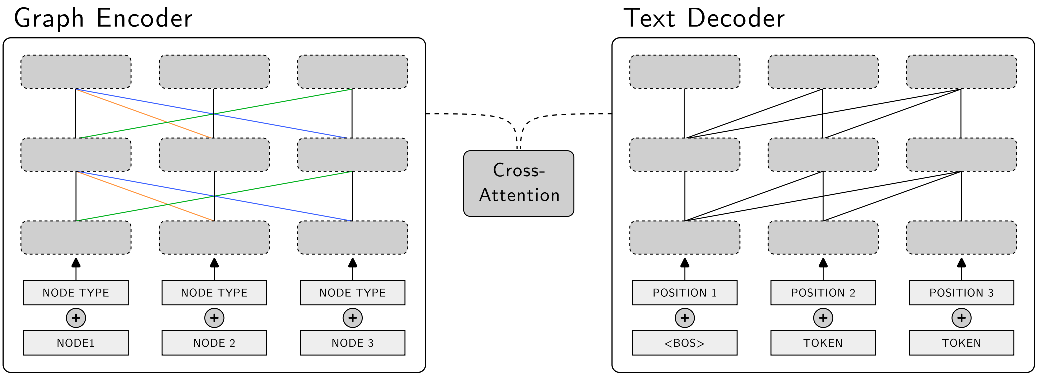
Graph Encoder



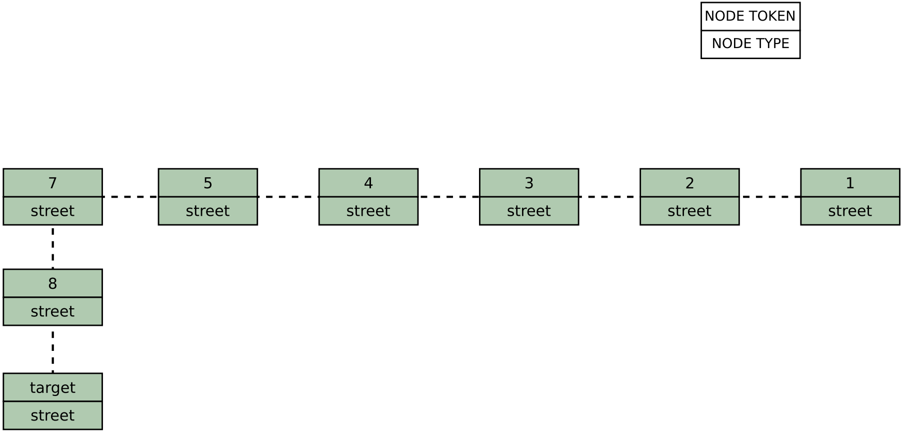


Labeled Edges [Schlichtkrull et al. 2018]

Graph-to-Text

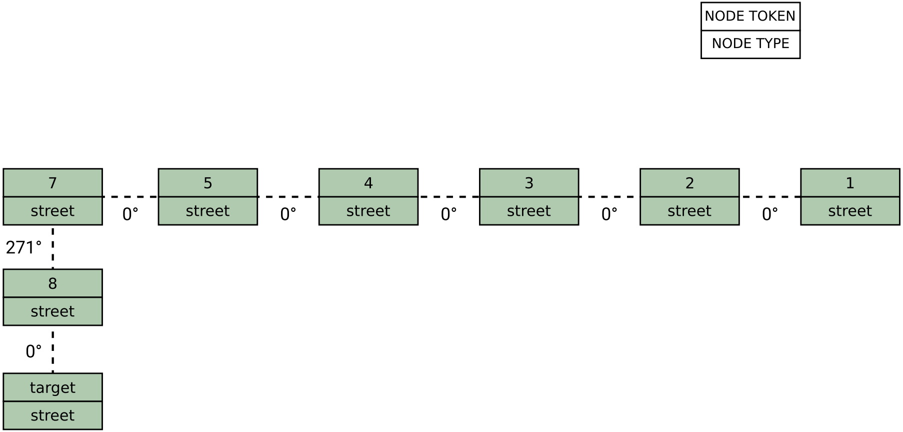


Route Graph Representation



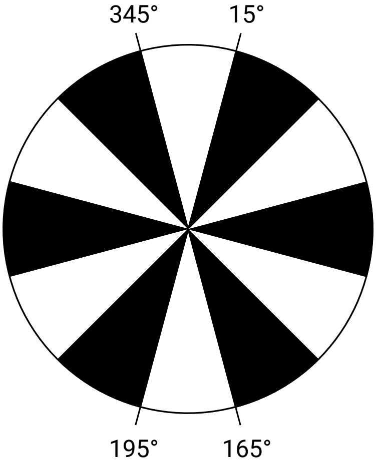
Go straight until the park on your right ends and turn left. Continue and stop with Starbucks on your left.

Route Graph Representation



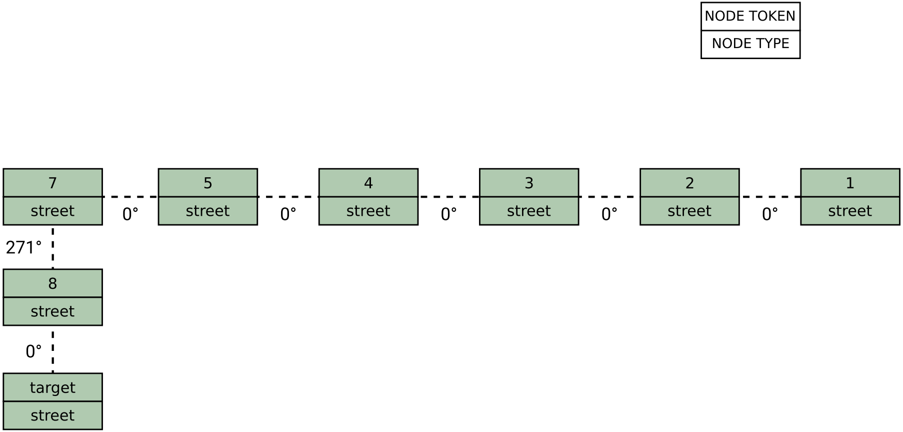
Go straight until the park on your right ends and turn left. Continue and stop with Starbucks on your left.

Route Graph Representation



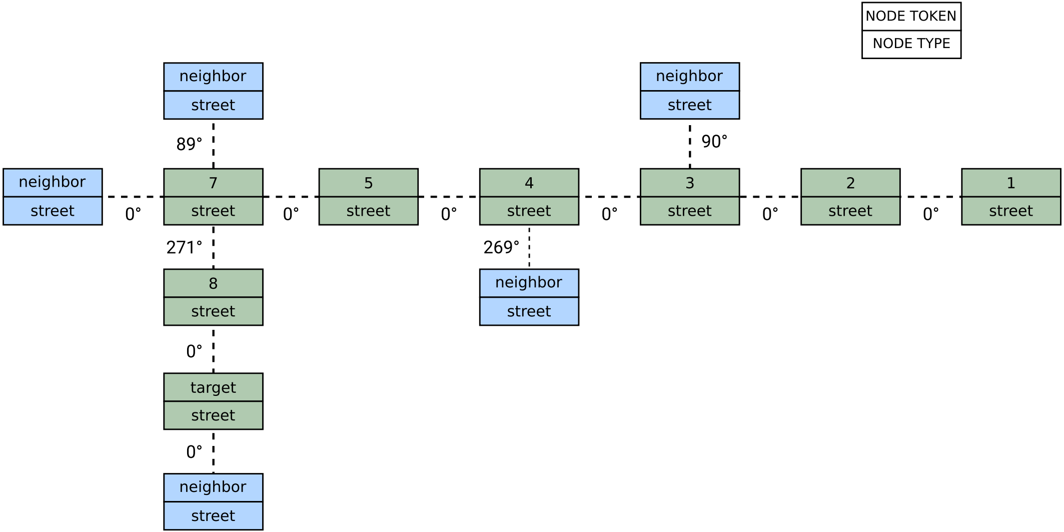
- discretize angles into 12 bins with 30° each
- label edges with angles according to angle bin

Route Graph Representation



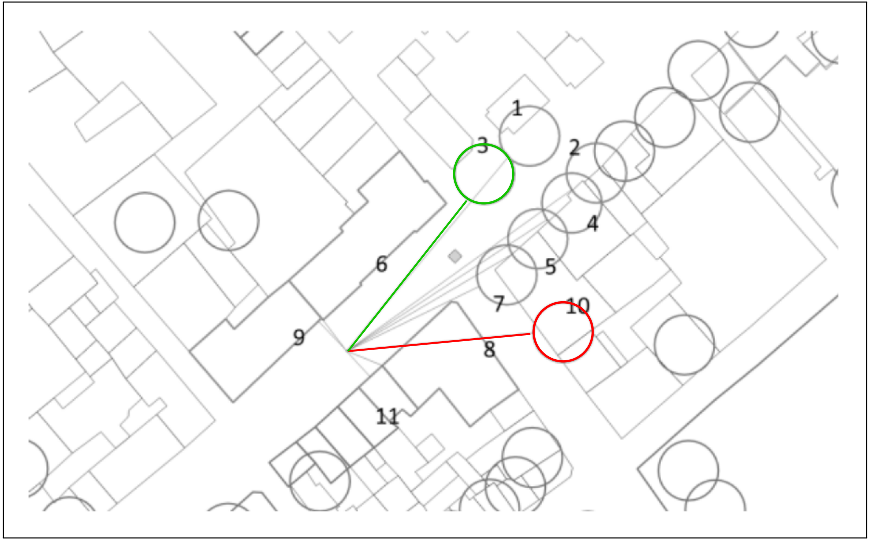
Go straight until the park on your right ends and turn left. Continue and stop with Starbucks on your left.

Route Graph Representation



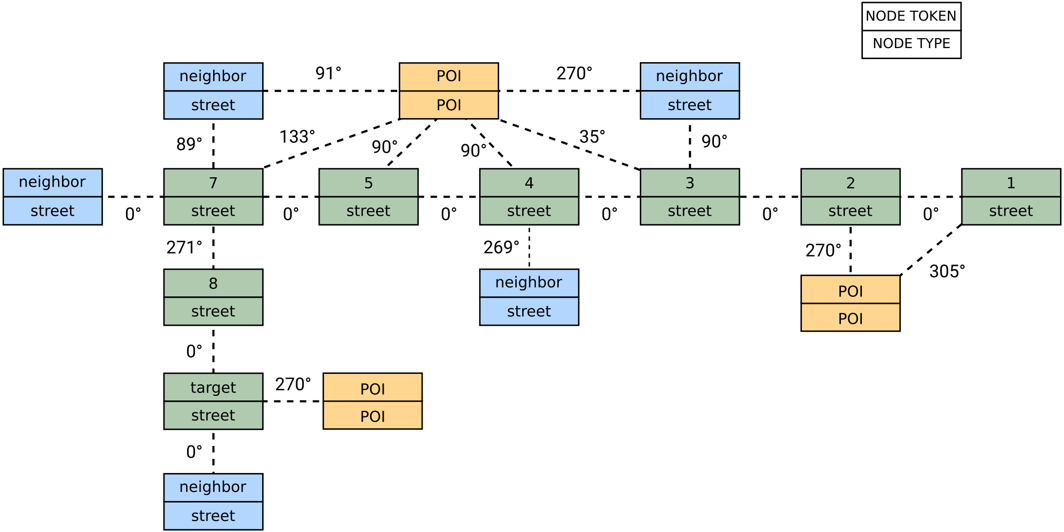
Go straight until the park on your right ends and turn left. Continue and stop with Starbucks on your left.

POI Visibility



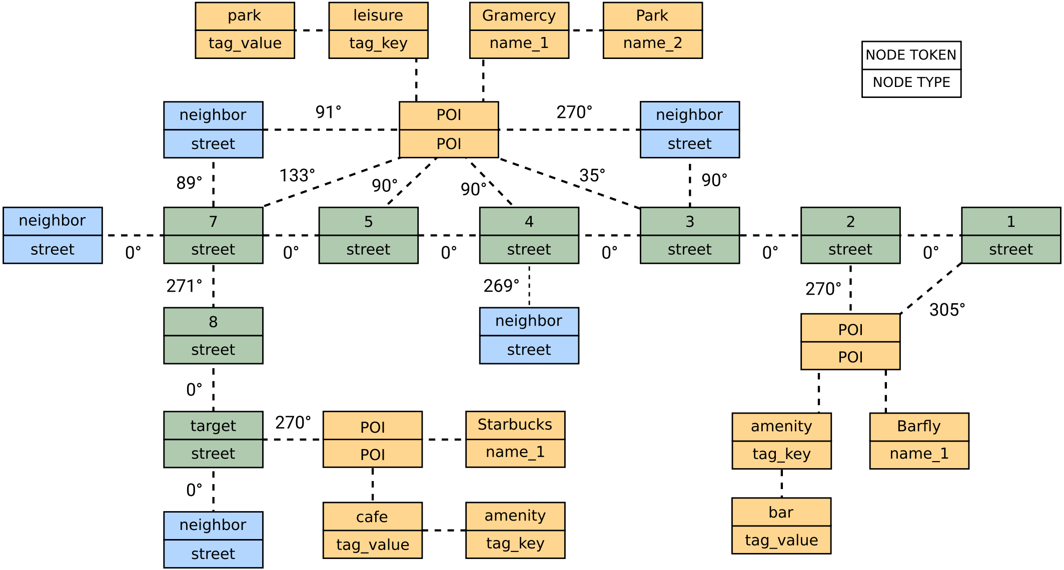
[Rousell et al. 2015]

Route Graph Representation



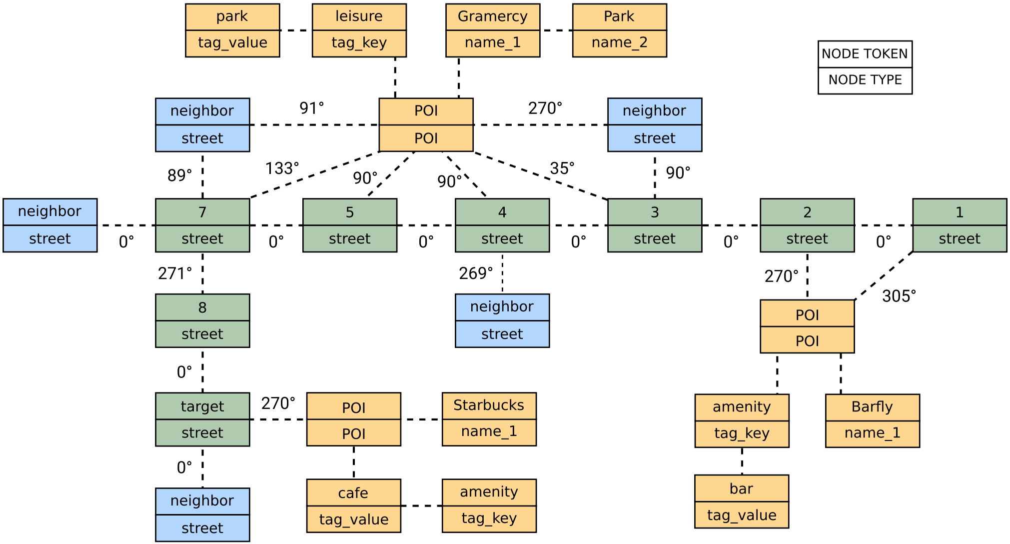
Go straight until the park on your right ends and turn left. Continue and stop with Starbucks on your left.

Route Graph Representation



Go straight until the park on your right ends and turn left. Continue and stop with Starbucks on your left.

Route Graph Representation

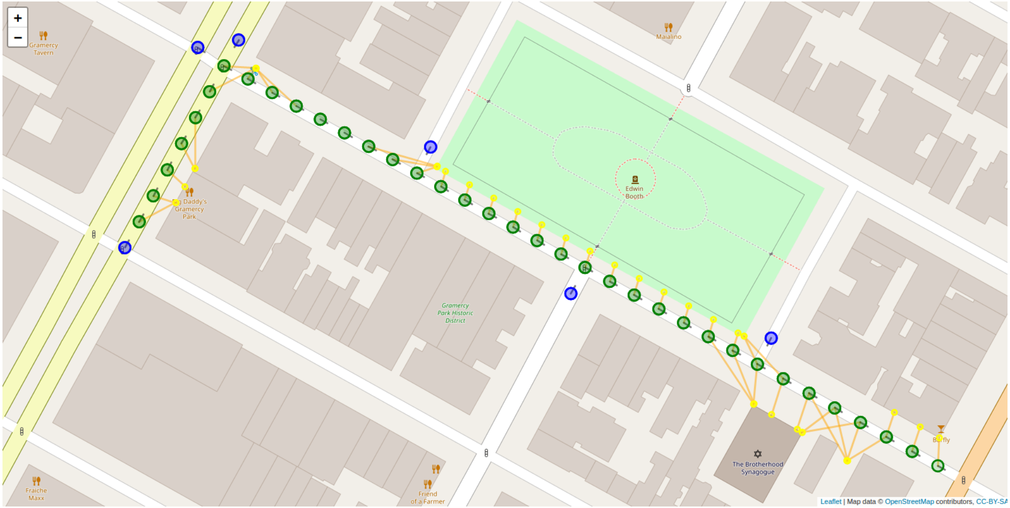


translation and rotation invariant $\rightarrow$ helps generalization

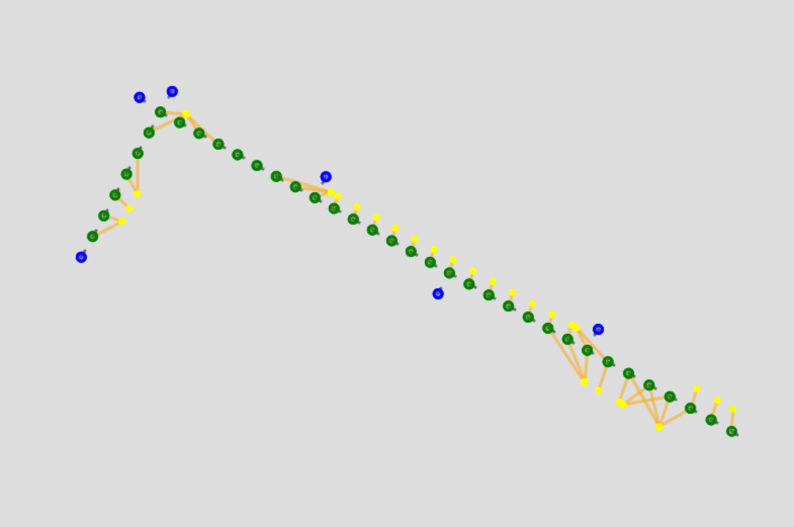
Route Graph Representation

Route Graph Representation

Route Graph Representation

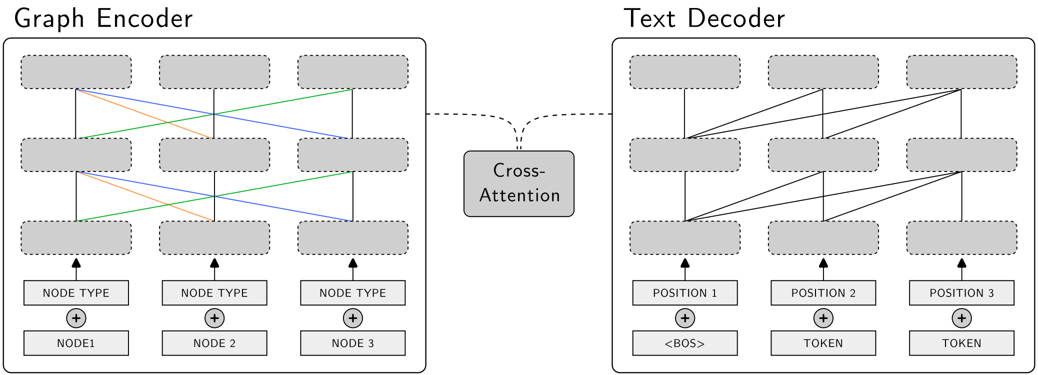


Route Graph Representation



On average, 144 nodes per graph and 4.3 edges per node.

Graph-to-Text

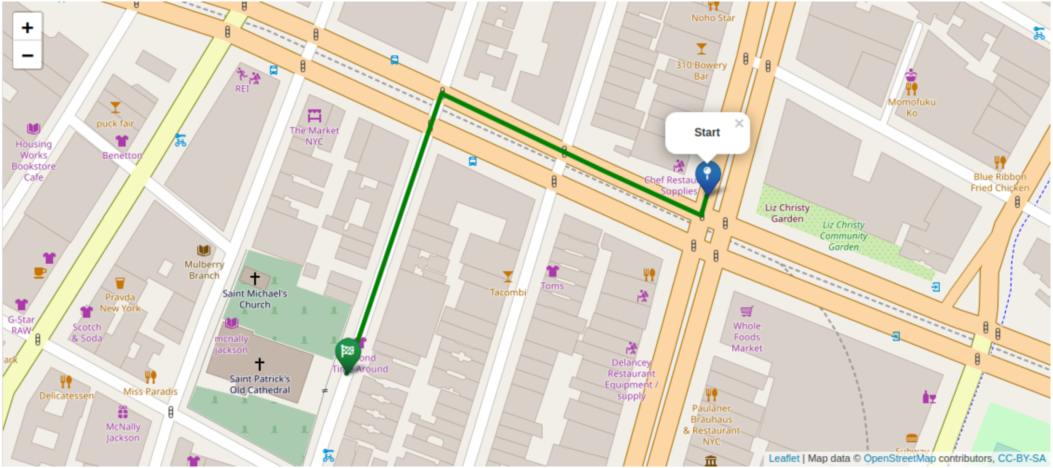


Generating Human-like Navigation Instructions

	Human Navigation Success
rule-based	46%
seq2seq	16%
graph2text	54%

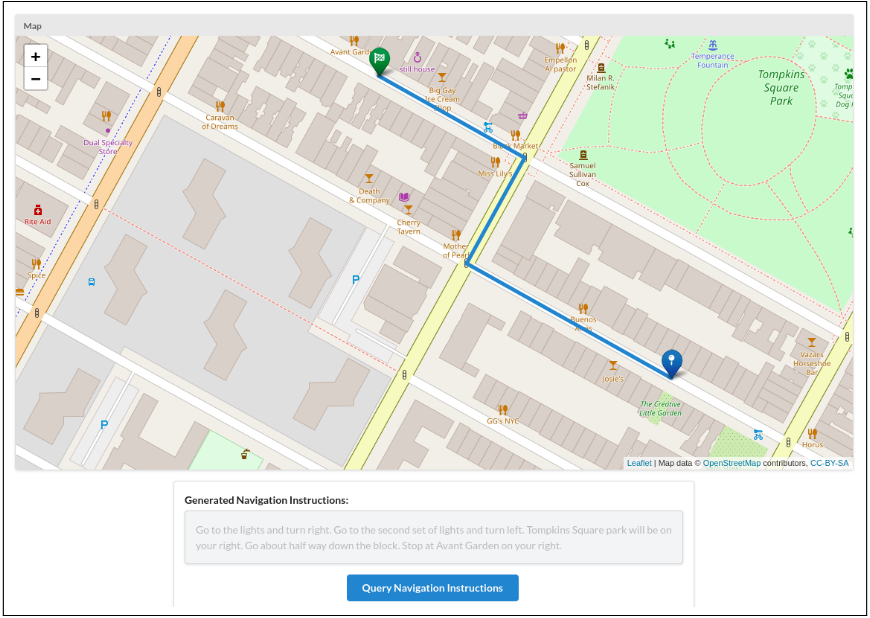
We also analyze the landmarks selected by the graph2text model and find that they closely follow the distribution of reference landmarks.

Example of Generated Instructions



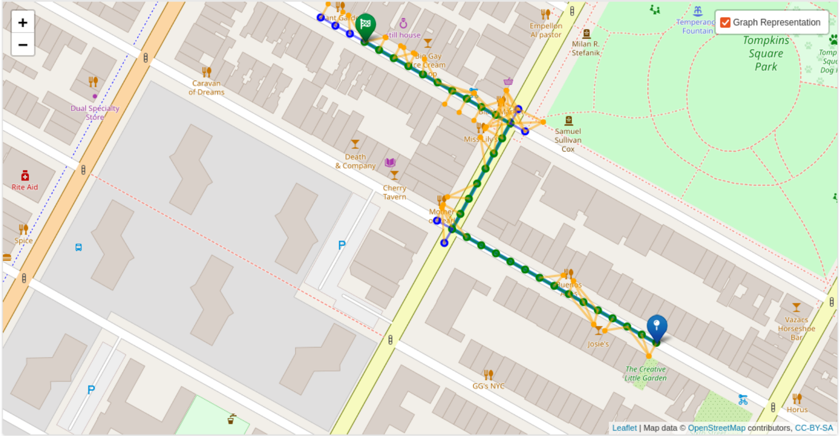
Go to the light and turn right. Go straight through 1 light and at the following light, there should be a bus stop on the far left corner. Turn left and go about 1/2 to the next light, stopping in front of Saint Patrick's church on the right and graveyard Memorial's on the right.

Demo: https://map2seq.schumann.pub/nllni/demo/



Demo: https://map2seq.schumann.pub/nllni/demo/

Map



Generated Navigation Instructions:

Go to the lights and turn right. At the next set of lights turn left. Pass Big Gay Ice Cream on the right and stop at the second building on the right. It is before Avant Garden.

move markers to select new route

Vision and Language Navigation

Any question so far?

Vision and Language Navigation in Street View

Agent Embodiment:

Action Space:

FORWARD
LEFT
RIGHT
STOP

Navigation Instructions:

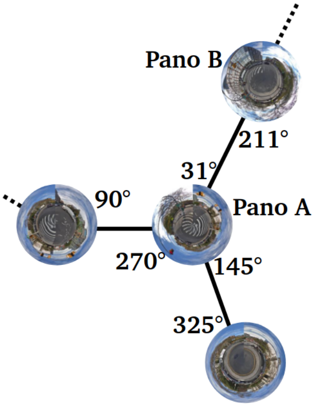
"Pass the bike rental and turn left at the lights. Go to the third set of lights with HSBC on the left and turn left. Stop just after Maison Kayser on the left."

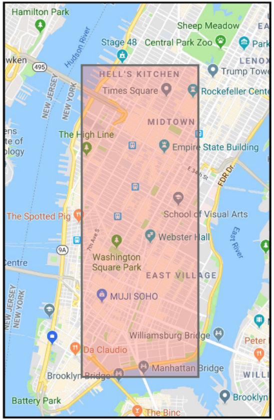
Vision and Language Navigation in Street View

Touchdown Environment:

[Chen et al. 2019]

- 30k panorama images
- Connectivity graph
- Lower Manhattan
- 10k navigation instructions





Touchdown vs. Map2seq

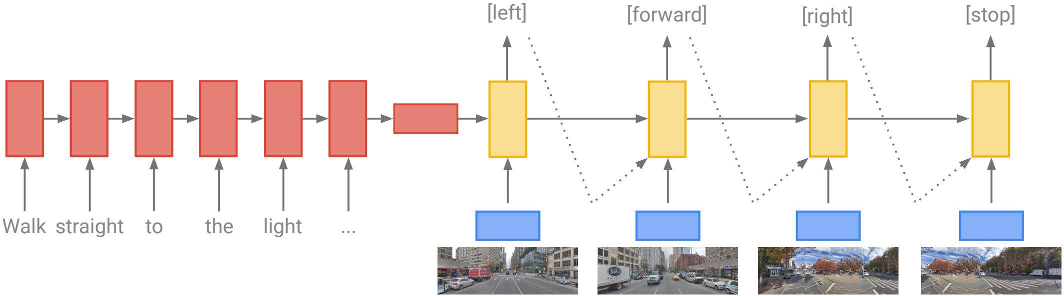
Touchdown:
Orient yourself with the flow of traffic, a clear awning to your left and green door to the right. Travel down this street all the way to the traffic light ahead and turn right. Stop at the store with a green and white sign to your right.

- path length: 35-45 nodes (with shorter outlier)
- instructions length: 89 tokens

Map2seq:
Head all the way down the block and with a Chase on the far right corner, make a right. Head through the next two lights, passing Banh Mi Saigon on the right and then stop in the middle of the next light.

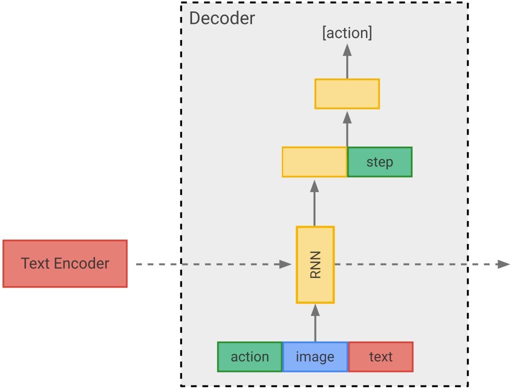
- shortest path
- path length: 35-45 nodes
- instructions length: 55 tokens

Vision and Language Navigation in Street View

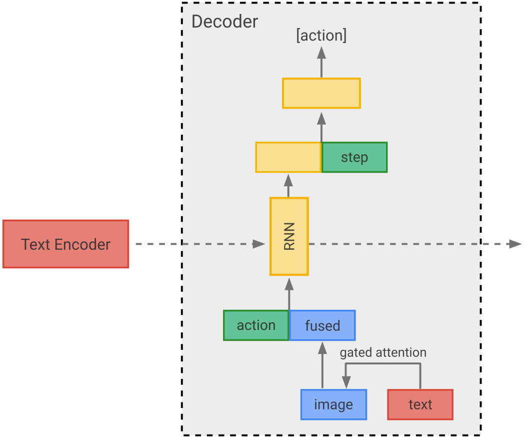


VLN as sequence-to-sequence model with RNN encoder and decocer.

Vision and Language Navigation in Street View

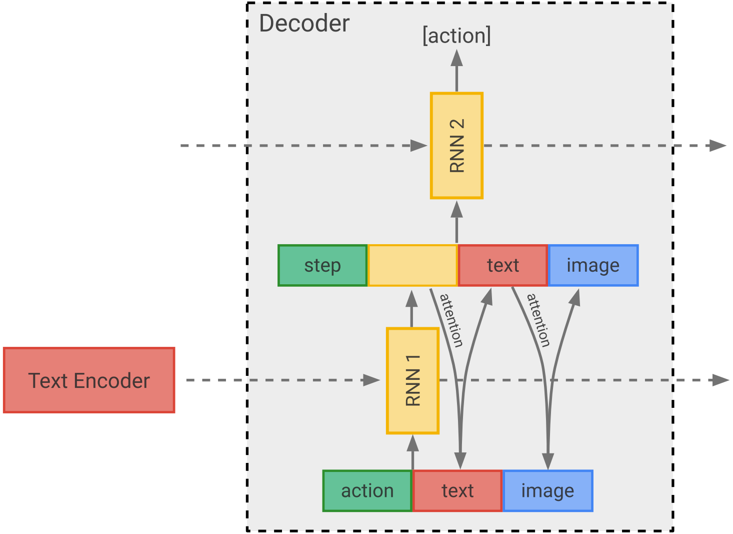


RConcat Model [Chen et al. 2019]



GatedAttention Model [Chaplot et al. 2018]

Vision and Language Navigation in Street View



ORAR Model [Schumann and Riezler 2022]

Vision and Language Navigation in Street View

Model	TC
RConcat [Chen et al. 2019]	13.3
GatedAttention [Chaplot et al. 2018]	14.5
VLN Transformer [Zhu et al. 2021]	16.0
ORAR [Schumann and Riezler 2022]	**34.7**

Task completion (TC) measures weather the agent stopped at the target location.

In the slides, I report the mean TC on Touchdown and map2seq navigation instances.

Vision and Language Navigation in Street View

Analyzing Generalization of Vision and Language Navigation to Unseen Outdoor Areas
Raphael Schumann and Stefan Riezler *ACL 2022*

Vision and Language Navigation in Street View

Model	TC (Seen)	TC (Unseen)
RConcat [Chen et al. 2019]	13.3	2.0
Gated-Attention [Chaplot et al. 2018]	14.5	2.0
VLN Transformer [Zhu et al. 2021]	16.0	3.3
ORAR [Schumann and Riezler 2022]	**34.7**	**4.6**

The task completion rate in the unseen scenario collapses for all models.

In that scenario, the evaluation area is geographically separated from the training area.

Vision and Language Navigation in Street View

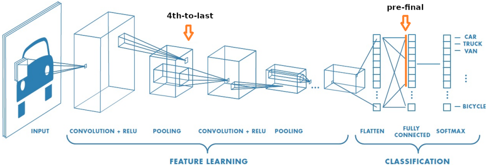
Vision and Language Navigation in Street View

Model	**TC (Seen)**	**TC (Unseen)**
ORAR	34.7	4.6
+ junction	37.1	19.2

Model	**TC (Seen)**	**TC (Unseen)**
ORAR	34.7	4.6
+ junction	37.1	19.2
+ heading delta	37.9	22.6

Vision and Language Navigation in Street View

ResNet Features



Extract **low-level or high-level** image feature from a pretrained ResNet.

Semantic Segmentation

$$f_{\text{DETECT}}=[a_{c_1}, a_{c_2}, \ldots, a_{c_n}];$$

$$a_{c_i}=\sum_{\text{obj is } c_i} \text{Area(obj)} \cdot \text{Conf(obj)}$$

Image representation as a vector, where each component is the **prevalence of an object class**.

Image	**TC (Seen)**	**TC (Unseen)**
no image	26.0	20.0
semantic segmentation	33.2	21.1
ResNet (high level)	**36.0**	16.7
ResNet (low level)	35.4	**22.6**

Image	**TC (Seen)**	**TC (Unseen)**
no image	26.0	20.0
semantic segmentation	33.2	21.1
ResNet (high level)	**36.0**	16.7
ResNet (low level)	35.4	**22.6**
fixed random vector	32.5	18.2

Fixed random vector as representation of each panorama image.

Verbalization Embodiment

Motivation: Use LLMs as the reasoning engine for embodied agents in Street View.

Challenges:

- Integrate visual observations into text interface

Advantages:

- unlocks few-shot learning
- no over-fitting to image features
- evaluation task for egocentric spatial-reasoning of LLMs

VELMA: Verbalization Embodiment of LLM Agents for VLN in Street View
Raphael Schumann, Wanrong Zhu, Weixi Feng, Tsu-Jui Fu,
Stefan Riezler, William Yang Wang *AAAI 2024*

Verbalization Embodiment - Text Interface

Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:

"*Orientate yourself such that **a blue bench** is on your right, go to the end of the block and make a right. Follow the **park** on your left and make a right at the intersection. Pass the **black fire hydrant** on your right and stop when you get to a **gray door on the brown building**.*"

Action Sequence:
There is a blue bench on your left.
1. **turn_around**
There is a blue bench on your right.
2. **forward**
There is a 3-way intersection.
3. **right**
4. **forward**
There is a park on your left.
5. **forward**
There is a park on your left.
6. **forward**
There is a 4-way intersection.
7. ***<next word prediction>***

Verbalization Embodiment - Pipeline

Write a list of visible landmarks mentioned in the navigation instructions:

Go straight down the road and turn right at the next intersection. Go straight until there is a *Starbucks* on your right and turn left at the following intersection with the *bicycle rental* at the corner. Continue down the block and stop when *a church* is on your left.

1. Starbucks
2. a bike rental
3. a church

Prompt to extract landmarks mentioned in the navigation instructions.
The list in blue is generated by the LLM.

Verbalization Embodiment - Pipeline

Prompt Sequence

Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:
"*Go straight down the road and turn right at the next intersection. Go straight until there is a **Starbucks** on your right and turn left at the following intersection with the **bicycle rental** at the corner. Continue down the block and stop when **a church** is on your left.*"
Action Sequence:
1. forward
2. forward

There is a 4-way intersection.

3. right
4. forward
5. forward
6. forward

Action

Environment

Extracted Landmarks

- Starbucks
- a bicycle rental
- a church

Verbalization Embodiment - Pipeline

Prompt Sequence

Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:
"*Go straight down the road and turn right at the next intersection. Go straight until there is a* ***Starbucks*** *on your right and turn left at the following intersection with the* ***bicycle rental*** *at the corner. Continue down the block and stop when* ***a church*** *is on your left.*"
Action Sequence:
1. forward
2. forward

There is a 4-way intersection.

3. right
4. forward
5. forward
6. forward

Environment

Action

Panorama & Heading

Extracted Landmarks

- Starbucks
- a bicycle rental
- a church

Verbalization Embodiment - Pipeline

Prompt Sequence

Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:
"*Go straight down the road and turn right at the next intersection. Go straight until there is a **Starbucks** on your right and turn left at the following intersection with the **bicycle rental** at the corner. Continue down the block and stop when **a church** is on your left.*"
Action Sequence:
1. forward
2. forward

There is a 4-way intersection.

3. right
4. forward
5. forward
6. forward

Action

Environment

Panorama & Heading

Extracted Landmarks

- Starbucks
- a bicycle rental
- a church

Landmark Detection

Verbalization Embodiment - Pipeline

Prompt Sequence

Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:
"*Go straight down the road and turn right at the next intersection. Go straight until there is a* ***Starbucks*** *on your right and turn left at the following intersection with the* ***bicycle rental*** *at the corner. Continue down the block and stop when* ***a church*** *is on your left.*"
Action Sequence:
1. forward
2. forward

There is a 4-way intersection.

3. right
4. forward
5. forward
6. forward

Environment

Action

Panorama & Heading

Extracted Landmarks

- Starbucks
- a bicycle rental
- a church

Landmark Detection

{"*Starbucks*": "*right*" }

Verbalization Embodiment - Pipeline

Prompt Sequence

Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:
"*Go straight down the road and turn right at the next intersection. Go straight until there is a **Starbucks** on your right and turn left at the following intersection with the **bicycle rental** at the corner. Continue down the block and stop when **a church** is on your left.*"
Action Sequence:
1. forward
2. forward

There is a 4-way intersection.

3. right
4. forward
5. forward
6. forward

There is a Starbucks on your right.

Environment

Action

Panorama & Heading

Extracted Landmarks

- Starbucks
- a bicycle rental
- a church

Landmark Detection

{"*Starbucks*": "*right*" }

Verbalized Observation

Verbalization Embodiment - Pipeline

Prompt Sequence

Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:
"*Go straight down the road and turn right at the next intersection. Go straight until there is a* ***Starbucks*** *on your right and turn left at the following intersection with the* ***bicycle rental*** *at the corner. Continue down the block and stop when* ***a church*** *is on your left.*"
Action Sequence:
1. forward
2. forward

There is a 4-way intersection.

3. right
4. forward
5. forward
6. forward

There is a Starbucks on your right.

7. *<next word prediction>*

Environment

Action

Panorama & Heading

Extracted Landmarks

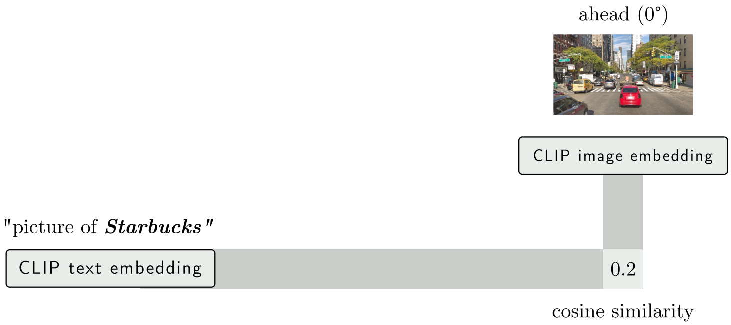
- Starbucks
- a bicycle rental
- a church

Landmark Detection

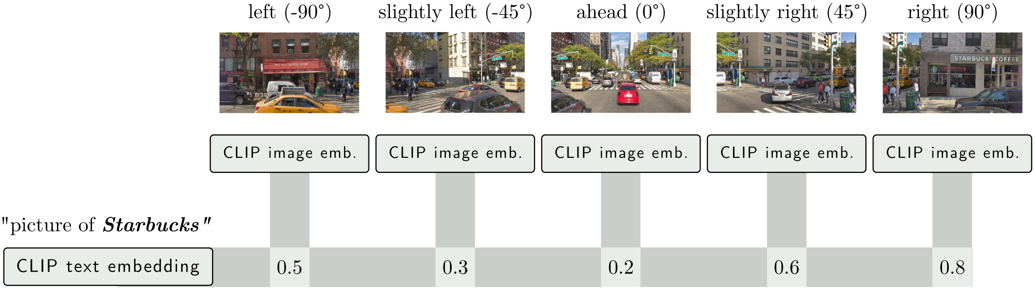
{"*Starbucks*": "*right*" }

Verbalized Observation

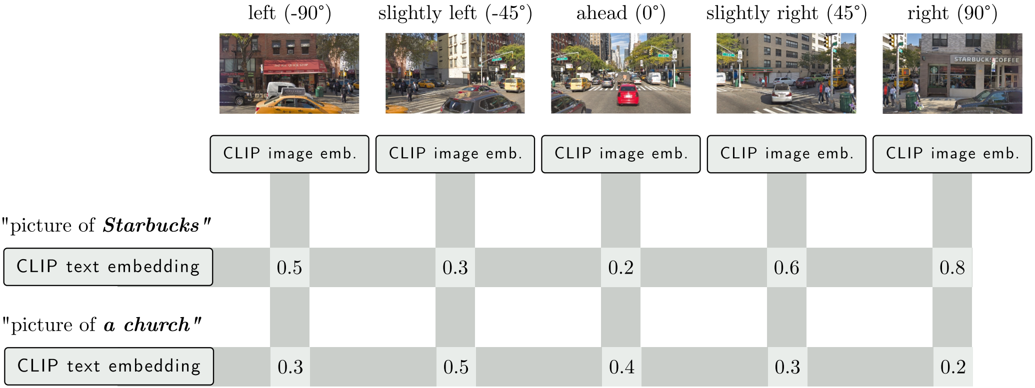
Landmark Detection



Landmark Detection



Landmark Detection



Landmark Detection

	left (-90°)	slightly left (-45°)	ahead (0°)	slightly right (45°)	right (90°)

	CLIP image emb.	CLIP image emb.	CLIP image emb.	CLIP image emb.	CLIP image emb.
"picture of ***Starbucks***" — CLIP text embedding	0.5	0.3	0.2	0.6	0.8
"picture of ***a church***" — CLIP text embedding	0.3	0.5	0.4	0.3	0.2
"picture of ***a bike rental***" — CLIP text embedding	0.2	0.3	0.2	0.4	0.6

Landmark Detection

	left (-90°)	slightly left (-45°)	ahead (0°)	slightly right (45°)	right (90°)

	CLIP image emb.	CLIP image emb.	CLIP image emb.	CLIP image emb.	CLIP image emb.
"picture of ***Starbucks***" CLIP text embedding	0.5	0.3	0.2	0.6	**0.8**
"picture of ***a church***" CLIP text embedding	0.3	0.5	0.4	0.3	0.2
"picture of ***a bike rental***" CLIP text embedding	0.2	0.3	0.2	0.4	0.6

Verbalization Embodiment - Pipeline

Prompt Sequence

Navigate to the described target location!
Action Space: forward, left, right, turn_around, stop
Navigation Instructions:
"*Go straight down the road and turn right at the next intersection. Go straight until there is a **Starbucks** on your right and turn left at the following intersection with the **bicycle rental** at the corner. Continue down the block and stop when **a church** is on your left.*"
Action Sequence:
1. forward
2. forward
There is a 4-way intersection.
3. right
4. forward
5. forward
6. forward
There is a Starbucks on your right.
7. *<next word prediction>*

Environment

Action

Panorama & Heading

Extracted Landmarks

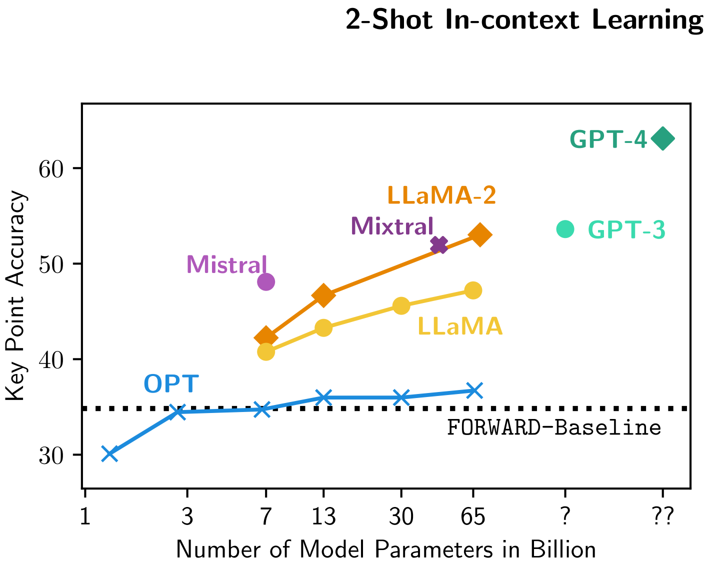
- Starbucks
- a bicycle rental
- a church

Landmark Detection

{"*Starbucks*": "*right*" }

Verbalized Observation

Verbalization Embodiment - Results



Model	TC (unseen)
Mixtral	7.3
GPT-3	8.0
GPT-4	**16.6**

Verbalization Embodiment - Results

Model	**TC (unseen)**
ORAR	25.2
VELMA-FT	**31.8**

VELMA-FT is LLaMA-7b finetuned on the full training set. Die finetuning text is the full verbalized trajectory prompt along the gold path.

Verbalization Embodiment - Results

Model	**TC (unseen)**
ORAR	25.2
VELMA-FT	**31.8**
VELMA-FT (no image)	26.4

VELMA-FT (no image) does not receive verbalized visual observations.

Verbalization Embodiment - Results

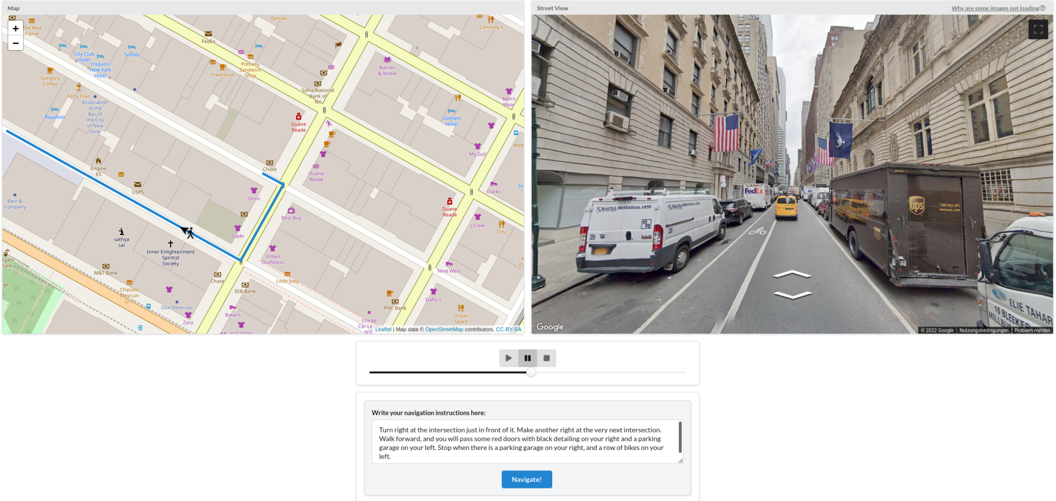
Model	TC (unseen)
ORAR	25.2
VELMA-FT	31.8
VELMA-RBL	**37.0**

Algorithm 1: RBL Optimization of Task Completion

Require: mixing ratio λ, training step j, model weights θ_j, gold action sequence $\hat{\mathbf{a}}$, prompt x_1

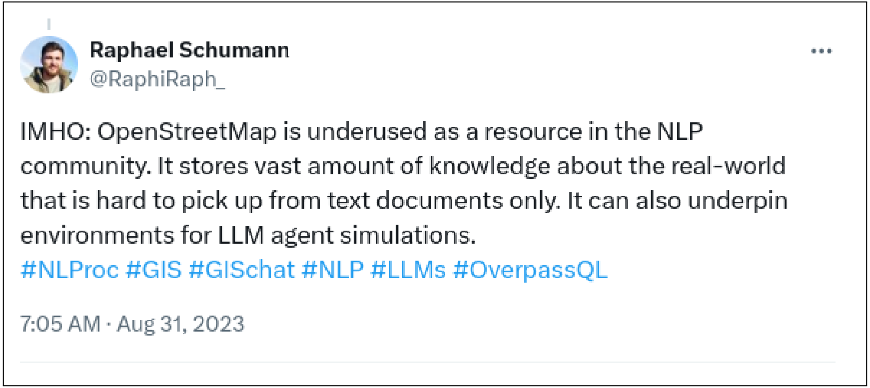
if $random(0, 1) < \lambda$ **then**
  $\mathbf{a}_{\theta_j} = StudentForcing(\theta_j, x_1)$
  $\mathbf{a}_j = \arg\max \mathbf{a}_{\theta_j}$
  if $TaskCompletion(\mathbf{a}_j) = 1$ **then**
    $loss_j = \mathcal{L}_{CE}(\mathbf{a}_{\theta_j}, \mathbf{a}_j)$
  else
    $\mathbf{a}_j^* = Oracle_{stepwise}(\mathbf{a}_j)$
    $loss_j = \mathcal{L}_{CE}(\mathbf{a}_{\theta_j}, \mathbf{a}_j^*)$
  end if
else
  $\mathbf{a}_{\theta_j} = TeacherForcing(\theta_j, x_1, \hat{\mathbf{a}})$
  $loss_j = \mathcal{L}_{CE}(\mathbf{a}_{\theta_j}, \hat{\mathbf{a}})$
end if

Demo: https://velma.schumann.pub



What's Next?

What's Next?



Raphael Schumann
@RaphiRaph_

IMHO: OpenStreetMap is underused as a resource in the NLP community. It stores vast amount of knowledge about the real-world that is hard to pick up from text documents only. It can also underpin environments for LLM agent simulations.
#NLProc #GIS #GIschat #NLP #LLMs #OverpassQL

7:05 AM · Aug 31, 2023

What's Next?

Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Joseph C. O'Brien
Stanford University
Stanford, USA
jobrien3@stanford.edu

Carrie J. Cai
Google Research
Mountain View, CA, USA
cjcai@google.com

Meredith Ringel Morris
Google DeepMind
Seattle, WA, USA
merrie@google.com

Percy Liang
Stanford University
Stanford, USA
pliang@cs.stanford.edu

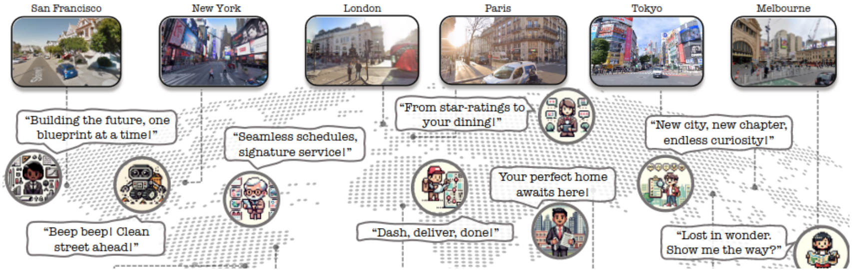
Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu

What's Next?

V-IRL: Grounding Virtual Intelligence in Real Life

Jihan Yang[1*] Runyu Ding[1] Ellis Brown[2] Xiaojuan Qi[1] Saining Xie[2]
[1]The University of Hong Kong [2]New York University

https://virl-platform.github.io



End

Chaplot, Devendra Singh et al. (2018). "Gated-Attention Architectures for Task-Oriented Language Grounding". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. New Orleans, Louisiana.

Chen, Howard et al. (2019). "TOUCHDOWN: Natural Language Navigation and Spatial Reasoning in Visual Street Environments". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, California.

Popović, Maja (Sept. 2015). "chrF: character n-gram F-score for automatic MT evaluation". In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, pp. 392–395.

Rousell, Adam et al. (June 2015). "Extraction of landmarks from OpenStreetMap for use in navigational instructions". In.

Schlichtkrull, Michael et al. (2018). "Modeling Relational Data with Graph Convolutional Networks". In: *The Semantic Web*. Ed. by Aldo Gangemi et al. Cham: Springer International Publishing, pp. 593–607.

Bibliography II

Schumann, Raphael and Stefan Riezler (May 2022). “Analyzing Generalization of Vision and Language Navigation to Unseen Outdoor Areas”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 7519–7532.

Vaswani, Ashish et al. (2017). “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc.

Veličković, Petar et al. (2018). “Graph Attention Networks”. In: *International Conference on Learning Representations*.

Zhu, Wanrong et al. (Apr. 2021). “Multimodal Text Style Transfer for Outdoor Vision-and-Language Navigation”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, pp. 1207–1221.