"HERE, where can I get something to eat near my hotel at 11 pm? "

Query to Intent-Slot Sequence Model using Multi-Head Attention

NGLS, Query Processing

Soojung Hong

May 27. 2020

Intent-Slot based Semantic Query Parsing & Service Graph Execution

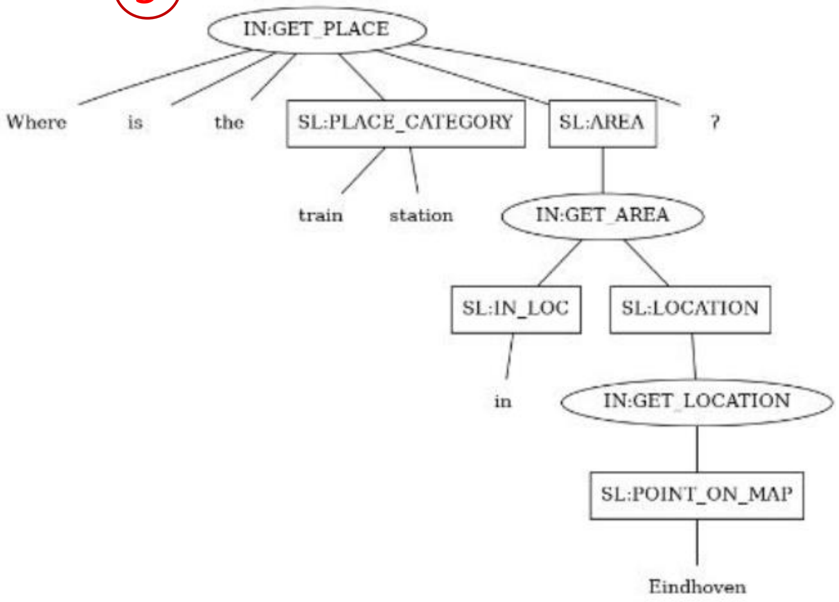
1

“Where is the train station in Eindhoven?”

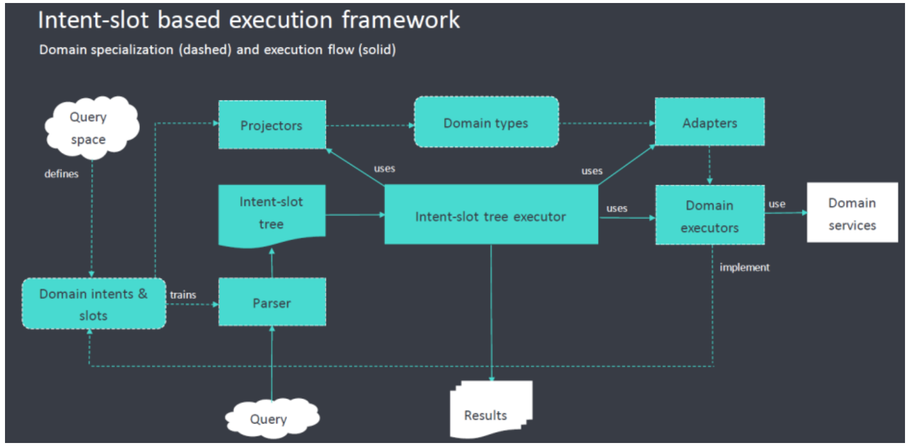
2

NGLS Semantic Query Parsing

3

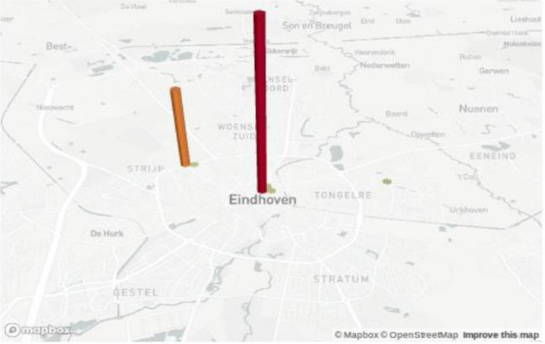


4


Intent-slot based execution framework

Domain specialization (dashed) and execution flow (solid)

5

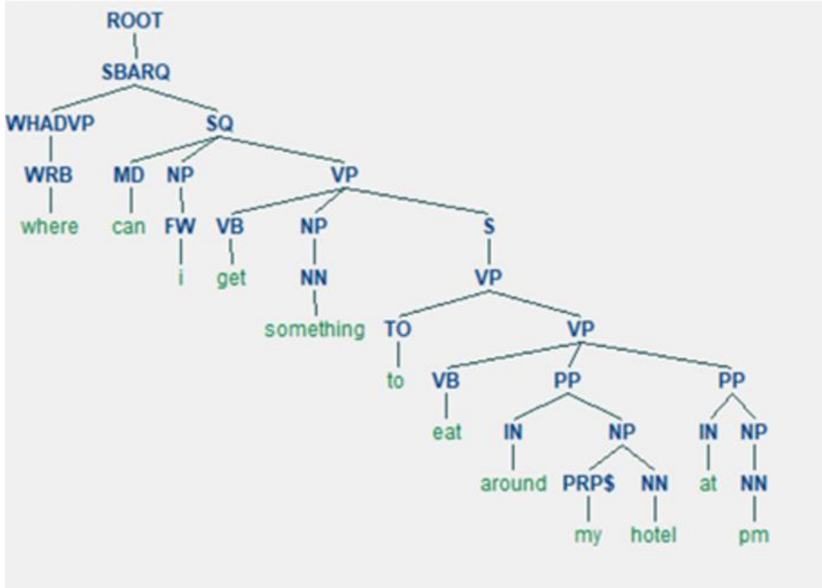


Query results

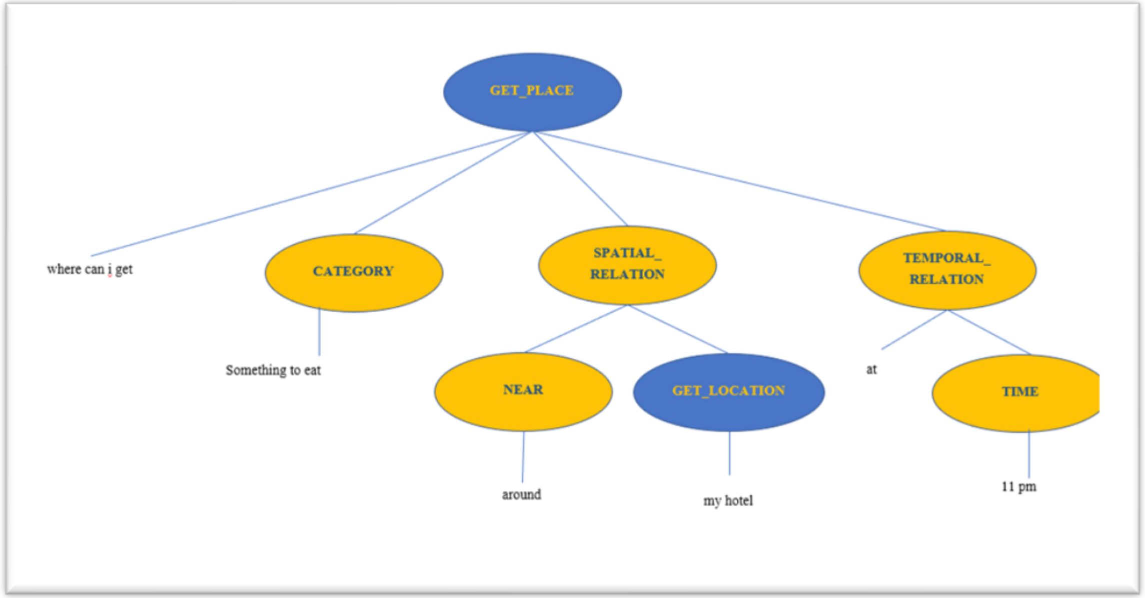
	name	id
0	Station Eindhoven-Beukenlaan	here:pds:place:528u15us-105fe2e50fb34db6ae1f9c9efd740b25
1	Eindhoven Strijp-S	here:pds:place:528u15us-3eba350e3d454ce381cf03134ce4120a
2	Station Eindhoven Strijp-S	here:pds:place:528u15us-c8469e73f7f04ab2b22bcc1240cb53d6

Let’s re-think about parsing

Syntactic : Constituency Parse Tree



Semantic : Intent-Slot based Parse Tree



- Semantic Query Parsing problem as Language Translation problem

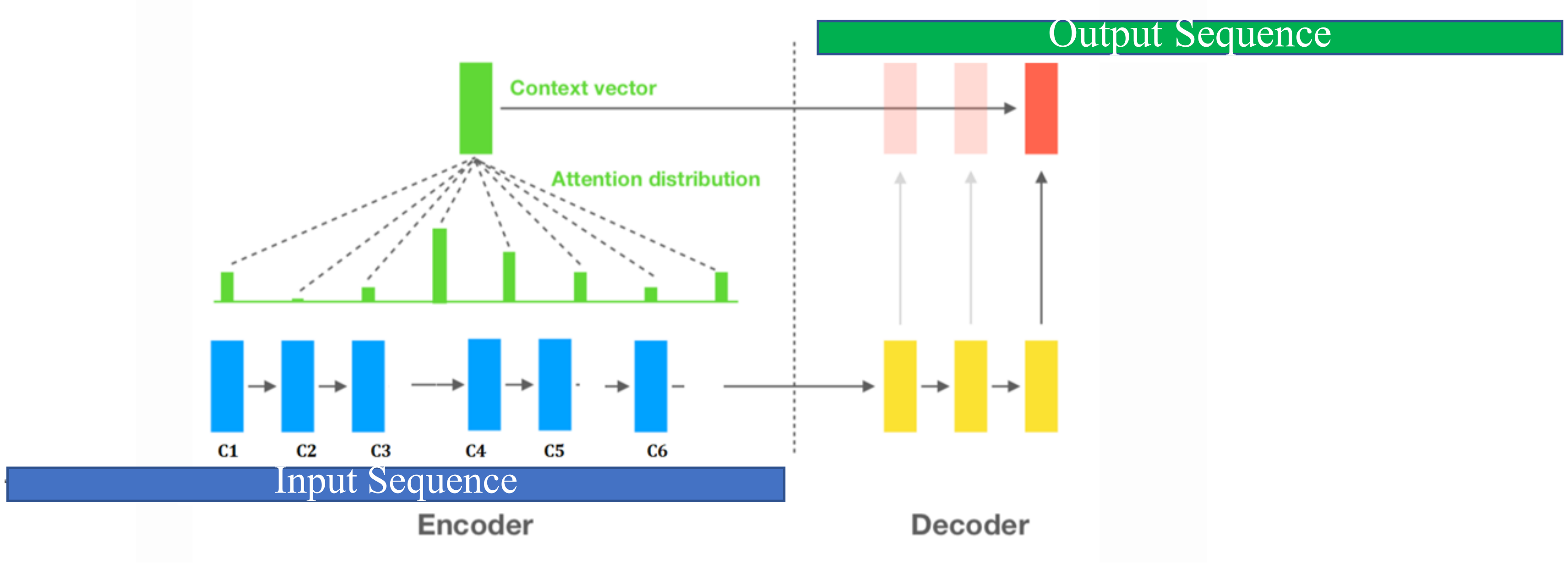
Query (Natural Language)

⬇

Service Graph Language

Query to Intent-Slot Sequence Model

Base model : Sequence to Sequence using Attention



English to Korean : “I” “am” “a” “boy” → “나는” “소년입니다”

Query to Intent-Slot :

Input (Query) :

where can i get something to eat around my hotel at 11 pm ?

Output (Intent-Slot Sequence):

GET_PLACE, CATEGORY, SPATIAL_RELATION, NEAR, GET_LOCATION, TEMPORAL_RELATION, TIME

Input and Output Sequence

Input Sequence :

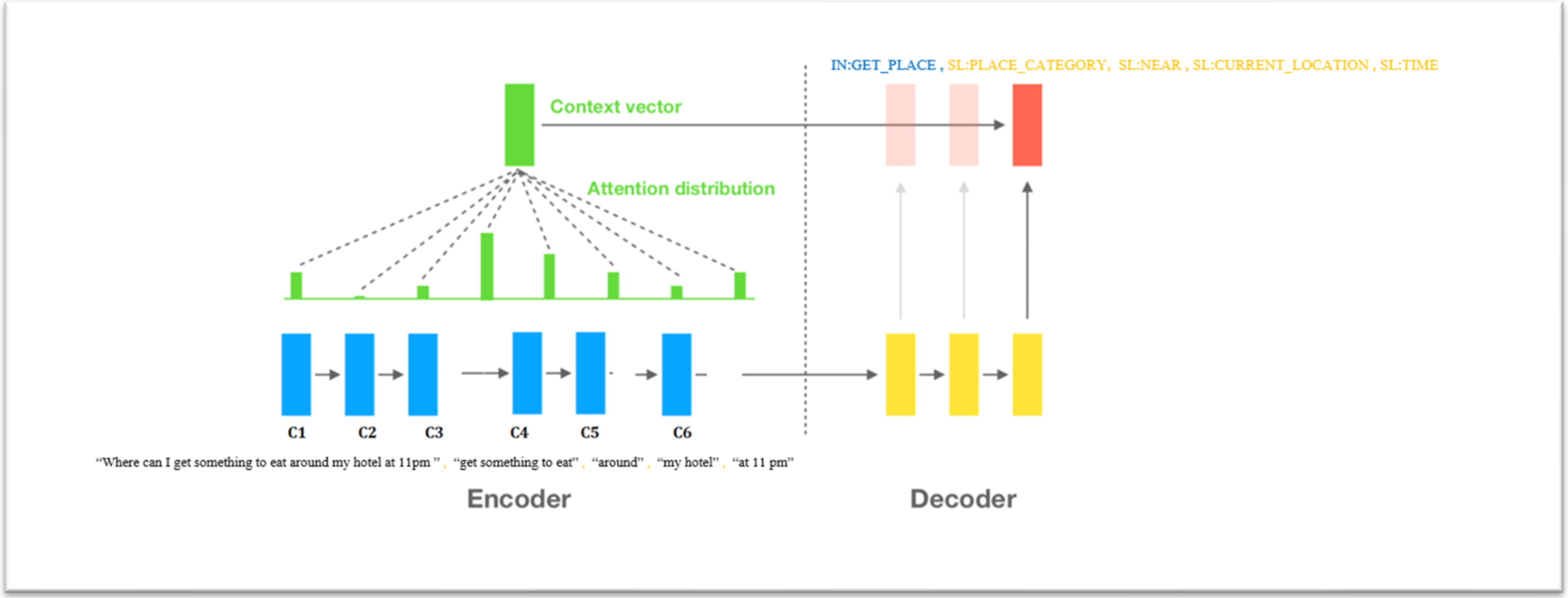
Not a Sequence of Token

~~“where” “can” “I” “get” “something” “to” “eat” “around” “my” “hotel” “at” “11” “pm” “?”~~

“where can i get something to eat around my hotel at 11 pm”, “something to eat”, “around my hotel”, “around”, “my hotel”, “at 11 pm”, “11pm”

Output Sequence :

GET_PLACE, CATEGORY, SPATIAL_RELATION, NEAR, GET_LOCATION, TEMPORAL_RELATION, TIME



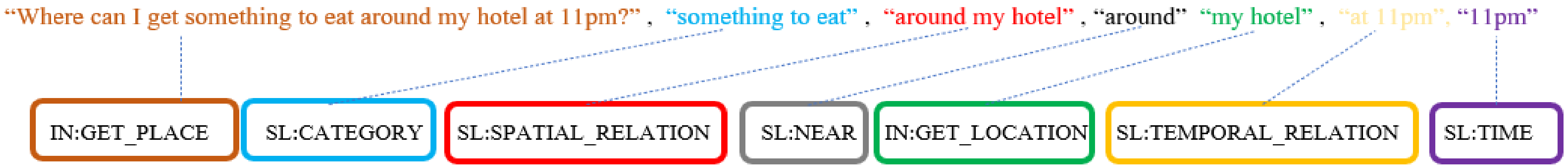
IN:GET_PLACE , SL:PLACE_CATEGORY, SL:NEAR , SL:CURRENT_LOCATION , SL:TIME
Context vector
Attention distribution
C1
C2
C3
C4
C5
C6
“Where can I get something to eat around my hotel at 11pm ” , “get something to eat” , “around” , “my hotel” , “at 11 pm”
Encoder
Decoder

Not a Sequence of Token

[Query]

"Where can I get something to eat around my hotel at 11pm?"

[Query to several phrases]



"Where can I get something to eat around my hotel at 11pm?" , "something to eat" , "around my hotel" , "around" "my hotel" , "at 11pm", "11pm"

IN:GET_PLACE SL:CATEGORY SL:SPATIAL_RELATION SL:NEAR IN:GET_LOCATION SL:TEMPORAL_RELATION SL:TIME

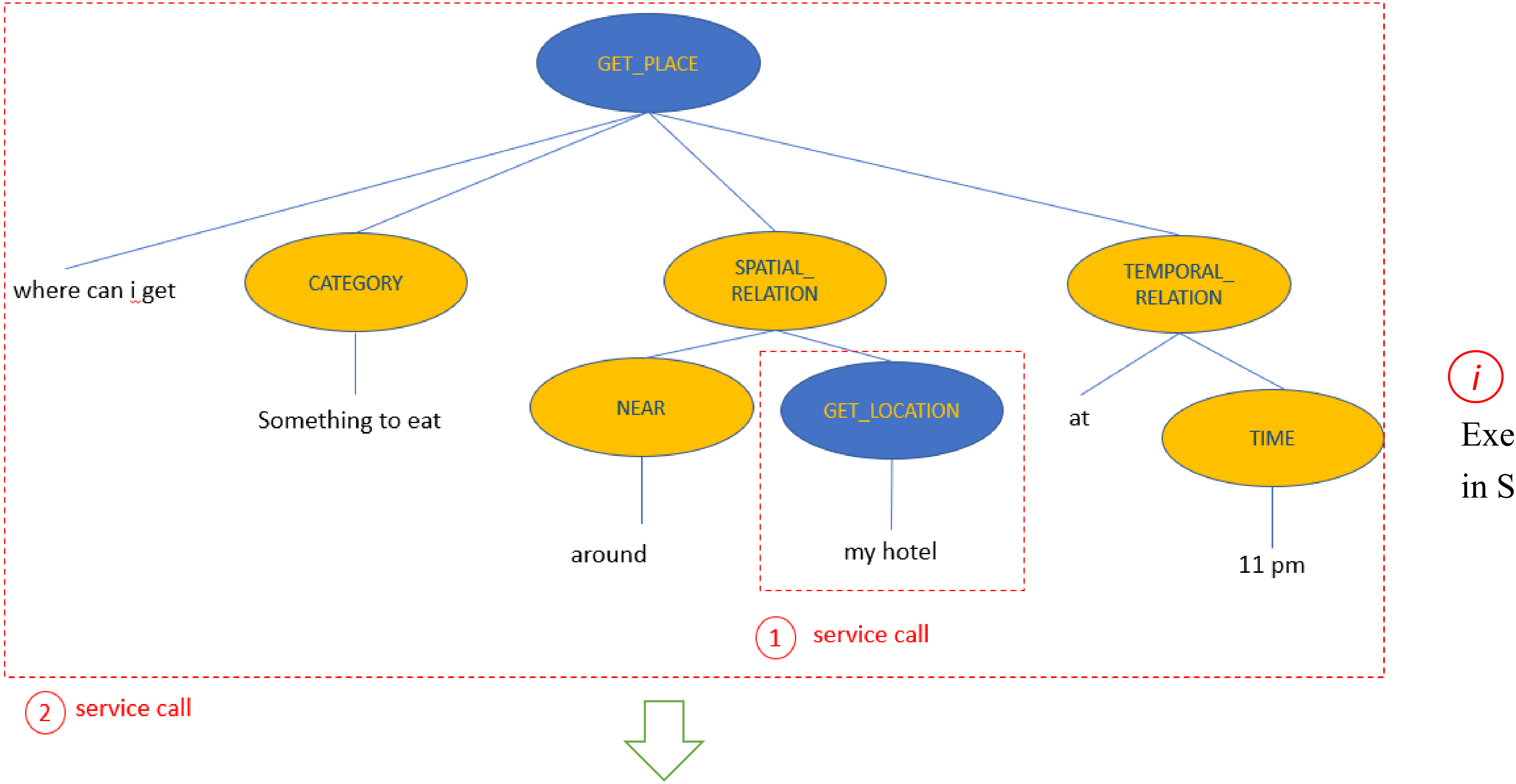
[Intent-Slot Labels]

Query with Intent-Slot Annotation :

GET_PLACE[*where can i get* CATEGORY[*something to eat*] SPATIAL_RELATION[NEAR[*around*] GET_LOCATION[*my hotel*]] TEMPORAL_RELATION[*at* TIME[*11 pm*]?]]

Automatic mapping to Parse tree and Service Graph execution

- Intents 🔵 map to Service type
- Slots 🟡 annotate phrases within query and map to parameters (modifiers) of Service



Execution order in Service graph

{GET_PLACE (CATEGORY: "Food", SPATIAL_RELATION: "Near", { GET_LOCATION(POINT_ON_MAP: Context.Location) }, TEMPORAL_RELATION: "When", { TIME : "11pm"} }

Intents and Slots types

- Intent types (= Service Types)

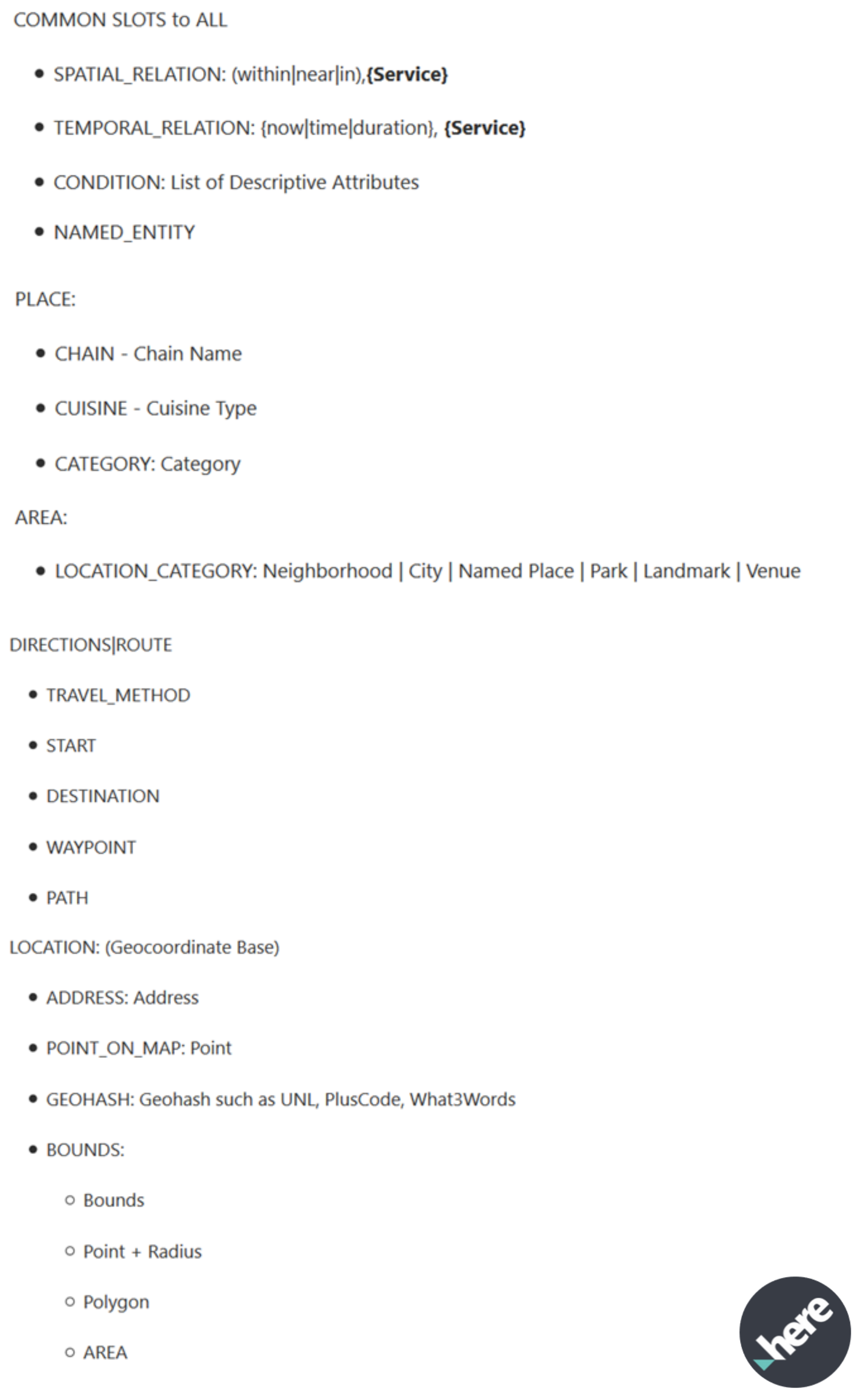
  GET_PLACE (any POI)

  GET_AREA (includes District, Geofence)

  GET_DIRECTIONS (includes Route)

  GET_LOCATION (includes Address, Point on Map, Intersection)

- Slot types (Modifier Types)



COMMON SLOTS to ALL

- SPATIAL_RELATION: (within|near|in),**{Service}**
- TEMPORAL_RELATION: {now|time|duration}, **{Service}**
- CONDITION: List of Descriptive Attributes
- NAMED_ENTITY

PLACE:

- CHAIN - Chain Name
- CUISINE - Cuisine Type
- CATEGORY: Category

AREA:

- LOCATION_CATEGORY: Neighborhood | City | Named Place | Park | Landmark | Venue

DIRECTIONS|ROUTE

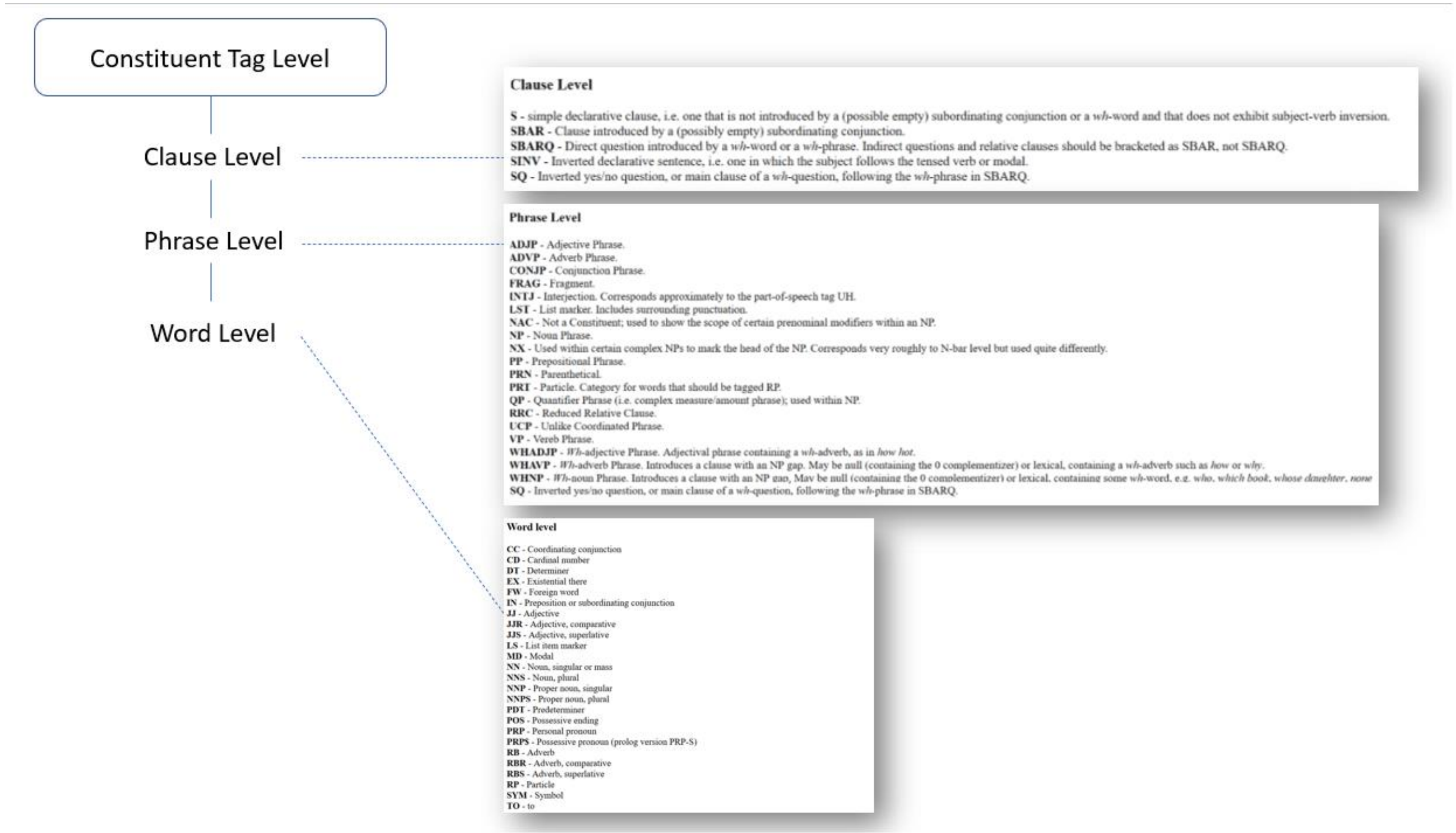
- TRAVEL_METHOD
- START
- DESTINATION
- WAYPOINT
- PATH

LOCATION: (Geocoordinate Base)

- ADDRESS: Address
- POINT_ON_MAP: Point
- GEOHASH: Geohash such as UNL, PlusCode, What3Words
- BOUNDS:
  - Bounds
  - Point + Radius
  - Polygon
  - AREA

Phrase Splitter for Input Sequence

- Extracting phrases to construct the input sequence
- Depth-First Search on *-Phrase node in Syntactic parse tree (* : Verb, Noun, Preposition,Wh-)



Clause Level

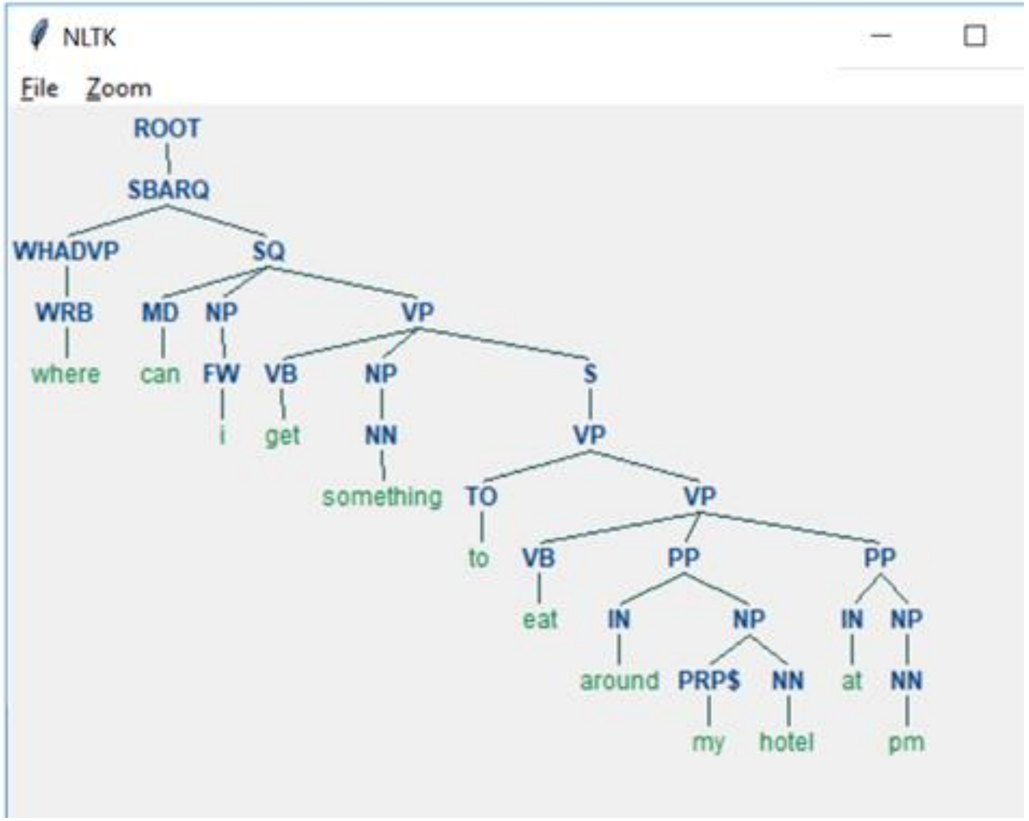
S - simple declarative clause, i.e. one that is not introduced by a (possible empty) subordinating conjunction or a *wh*-word and that does not exhibit subject-verb inversion.
SBAR - Clause introduced by a (possibly empty) subordinating conjunction.
SBARQ - Direct question introduced by a *wh*-word or a *wh*-phrase. Indirect questions and relative clauses should be bracketed as SBAR, not SBARQ.
SINV - Inverted declarative sentence, i.e. one in which the subject follows the tensed verb or modal.
SQ - Inverted yes/no question, or main clause of a *wh*-question, following the *wh*-phrase in SBARQ.

Phrase Level

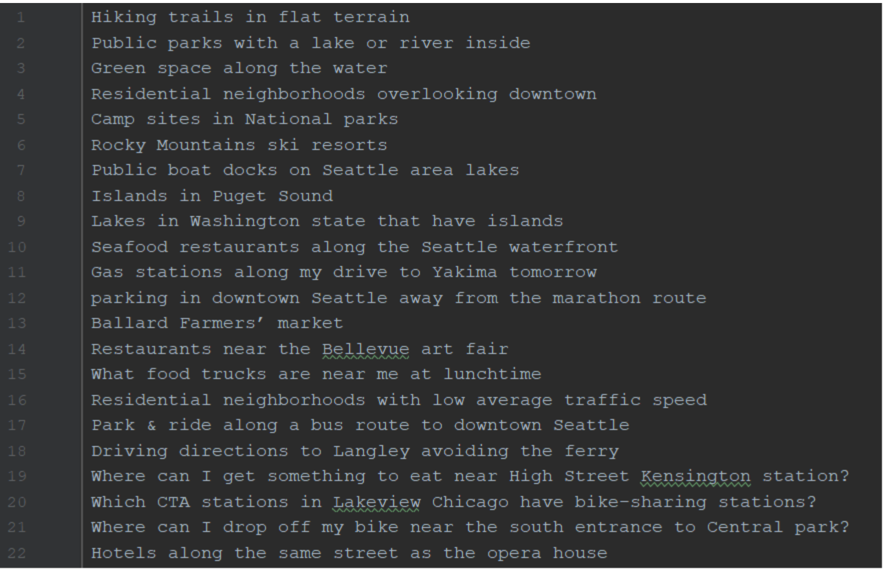
ADJP - Adjective Phrase.
ADVP - Adverb Phrase.
CONJP - Conjunction Phrase.
FRAG - Fragment.
INTJ - Interjection. Corresponds approximately to the part-of-speech tag UH.
LST - List marker. Includes surrounding punctuation.
NAC - Not a Constituent; used to show the scope of certain prenominal modifiers within an NP.
NP - Noun Phrase.
NX - Used within certain complex NPs to mark the head of the NP. Corresponds very roughly to N-bar level but used quite differently.
PP - Prepositional Phrase.
PRN - Parenthetical.
PRT - Particle. Category for words that should be tagged RP.
QP - Quantifier Phrase (i.e. complex measure/amount phrase); used within NP.
RRC - Reduced Relative Clause.
UCP - Unlike Coordinated Phrase.
VP - Vereb Phrase.
WHADJP - *Wh*-adjective Phrase. Adjectival phrase containing a *wh*-adverb, as in *how hot*.
WHAVP - *Wh*-adverb Phrase. Introduces a clause with an NP gap. May be null (containing the 0 complementizer) or lexical, containing a *wh*-adverb such as *how* or *why*.
WHNP - *Wh*-noun Phrase. Introduces a clause with an NP gap, May be null (containing the 0 complementizer) or lexical, containing some *wh*-word, e.g. *who, which book, whose daughter, none*
SQ - Inverted yes/no question, or main clause of a *wh*-question, following the *wh*-phrase in SBARQ.

Word level

CC - Coordinating conjunction
CD - Cardinal number
DT - Determiner
EX - Existential there
FW - Foreign word
IN - Preposition or subordinating conjunction
JJ - Adjective
JJR - Adjective, comparative
JJS - Adjective, superlative
LS - List item marker
MD - Modal
NN - Noun, singular or mass
NNS - Noun, plural
NNP - Proper noun, singular
NNPS - Proper noun, plural
PDT - Predeterminer
POS - Possessive ending
PRP - Personal pronoun
PRP$ - Possessive pronoun (prolog version PRP-S)
RB - Adverb
RBR - Adverb, comparative
RBS - Adverb, superlative
RP - Particle
SYM - Symbol
TO - to

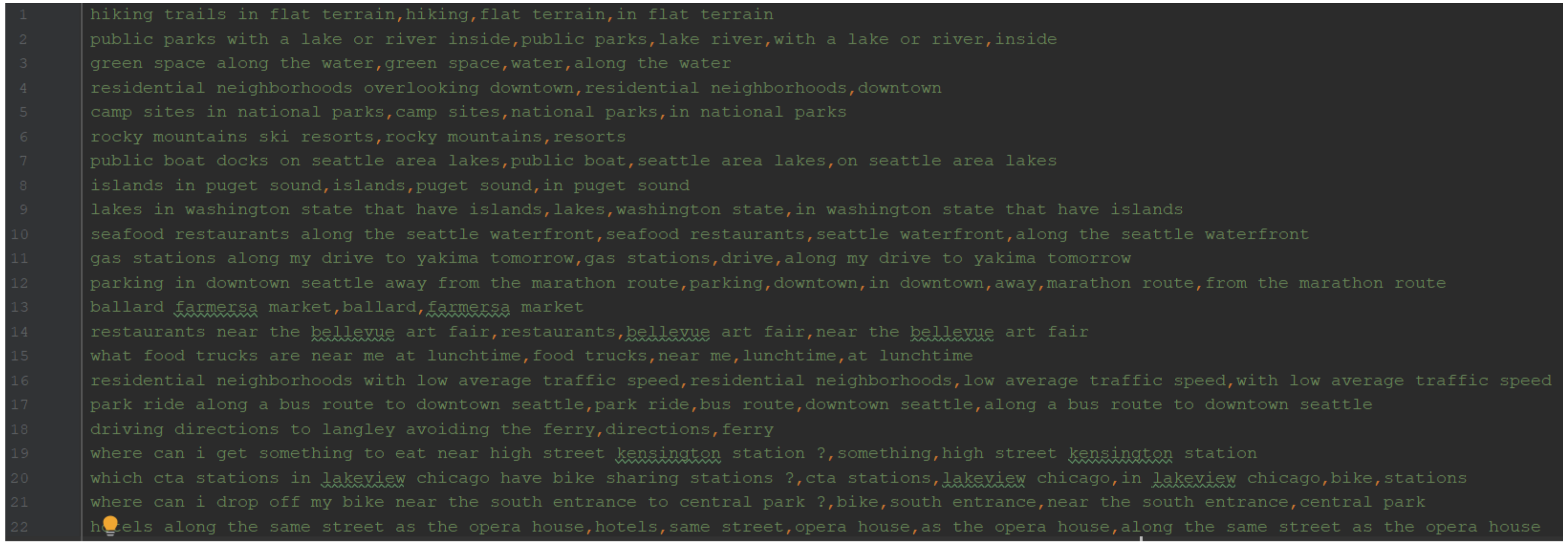


MVP queries & query input sequences



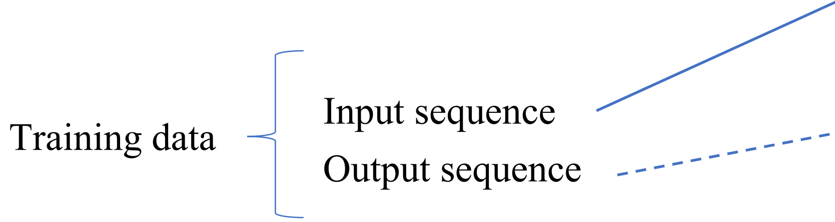
```
Hiking trails in flat terrain
Public parks with a lake or river inside
Green space along the water
Residential neighborhoods overlooking downtown
Camp sites in National parks
Rocky Mountains ski resorts
Public boat docks on Seattle area lakes
Islands in Puget Sound
Lakes in Washington state that have islands
Seafood restaurants along the Seattle waterfront
Gas stations along my drive to Yakima tomorrow
parking in downtown Seattle away from the marathon route
Ballard Farmers’ market
Restaurants near the Bellevue art fair
What food trucks are near me at lunchtime
Residential neighborhoods with low average traffic speed
Park & ride along a bus route to downtown Seattle
Driving directions to Langley avoiding the ferry
Where can I get something to eat near High Street Kensington station?
Which CTA stations in Lakeview Chicago have bike-sharing stations?
Where can I drop off my bike near the south entrance to Central park?
Hotels along the same street as the opera house
```

MVP Queries



```
hiking trails in flat terrain,hiking,flat terrain,in flat terrain
public parks with a lake or river inside,public parks,lake river,with a lake or river,inside
green space along the water,green space,water,along the water
residential neighborhoods overlooking downtown,residential neighborhoods,downtown
camp sites in national parks,camp sites,national parks,in national parks
rocky mountains ski resorts,rocky mountains,resorts
public boat docks on seattle area lakes,public boat,seattle area lakes,on seattle area lakes
islands in puget sound,islands,puget sound,in puget sound
lakes in washington state that have islands,lakes,washington state,in washington state that have islands
seafood restaurants along the seattle waterfront,seafood restaurants,seattle waterfront,along the seattle waterfront
gas stations along my drive to yakima tomorrow,gas stations,drive,along my drive to yakima tomorrow
parking in downtown seattle away from the marathon route,parking,downtown,in downtown,away,marathon route,from the marathon route
ballard farmersa market,ballard,farmersa market
restaurants near the bellevue art fair,restaurants,bellevue art fair,near the bellevue art fair
what food trucks are near me at lunchtime,food trucks,near me,lunchtime,at lunchtime
residential neighborhoods with low average traffic speed,residential neighborhoods,low average traffic speed,with low average traffic speed
park ride along a bus route to downtown seattle,park ride,bus route,downtown seattle,along a bus route to downtown seattle
driving directions to langley avoiding the ferry,directions,ferry
where can i get something to eat near high street kensington station ?,something,high street kensington station
which cta stations in lakeview chicago have bike sharing stations ?,cta stations,lakeview chicago,in lakeview chicago,bike,stations
where can i drop off my bike near the south entrance to central park ?,bike,south entrance,near the south entrance,central park
hotels along the same street as the opera house,hotels,same street,opera house,as the opera house,along the same street as the opera house
```

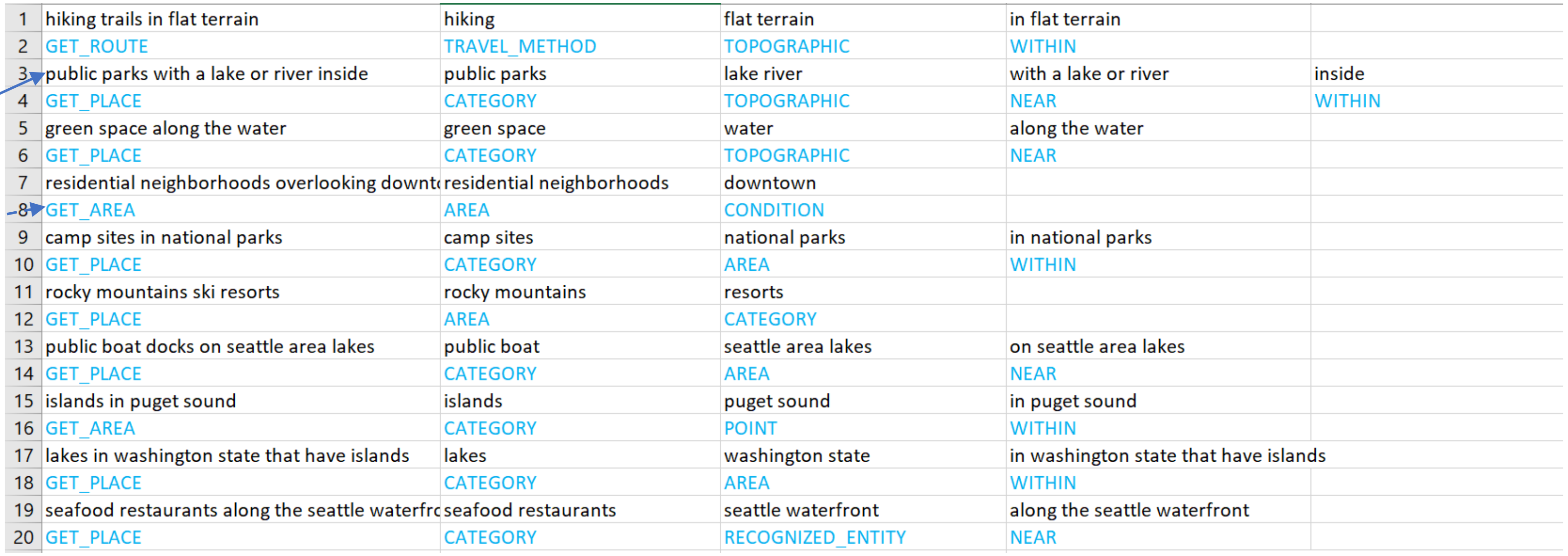
Sequences of Phrases



Training data

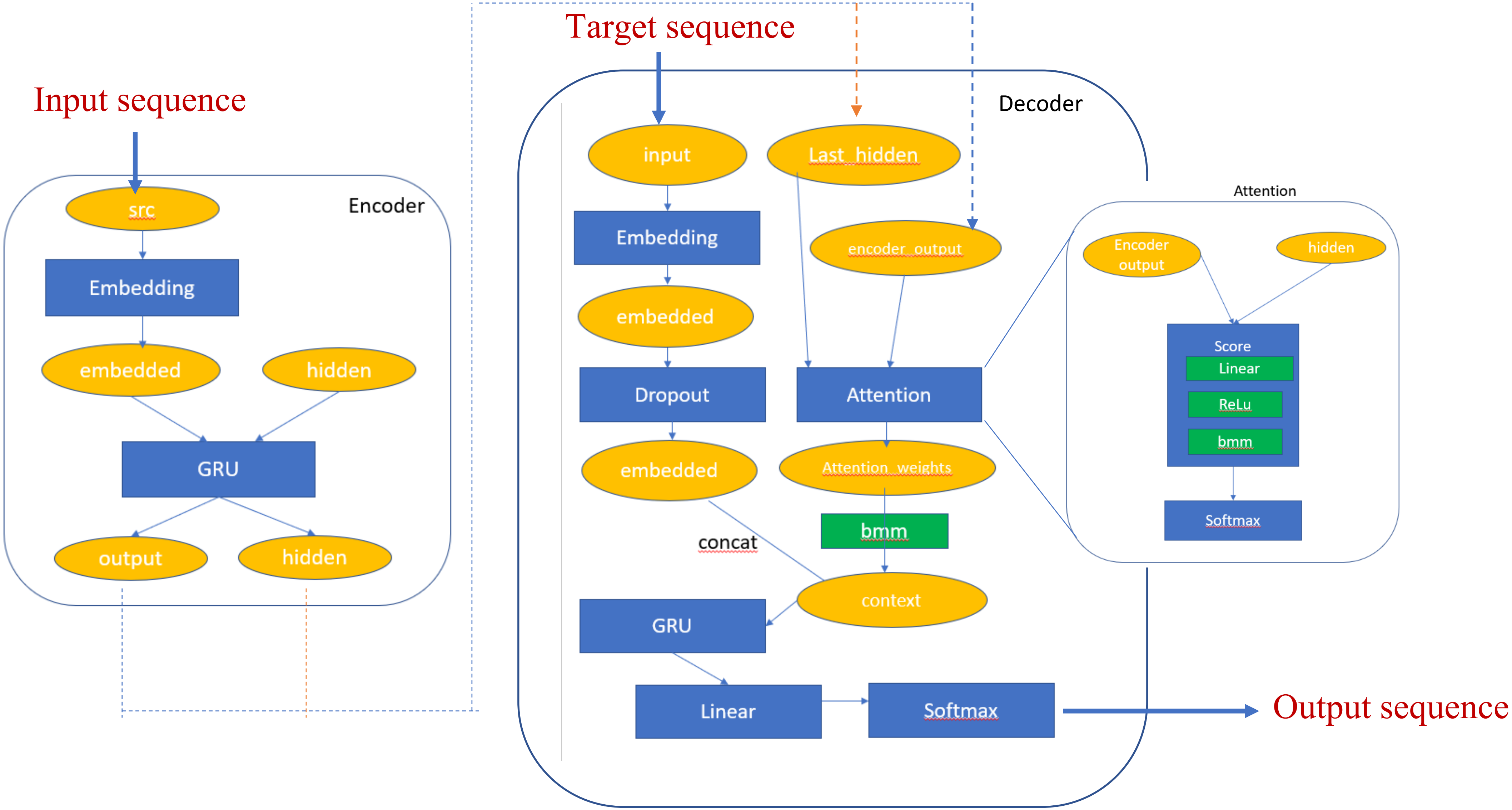
Input sequence

Output sequence

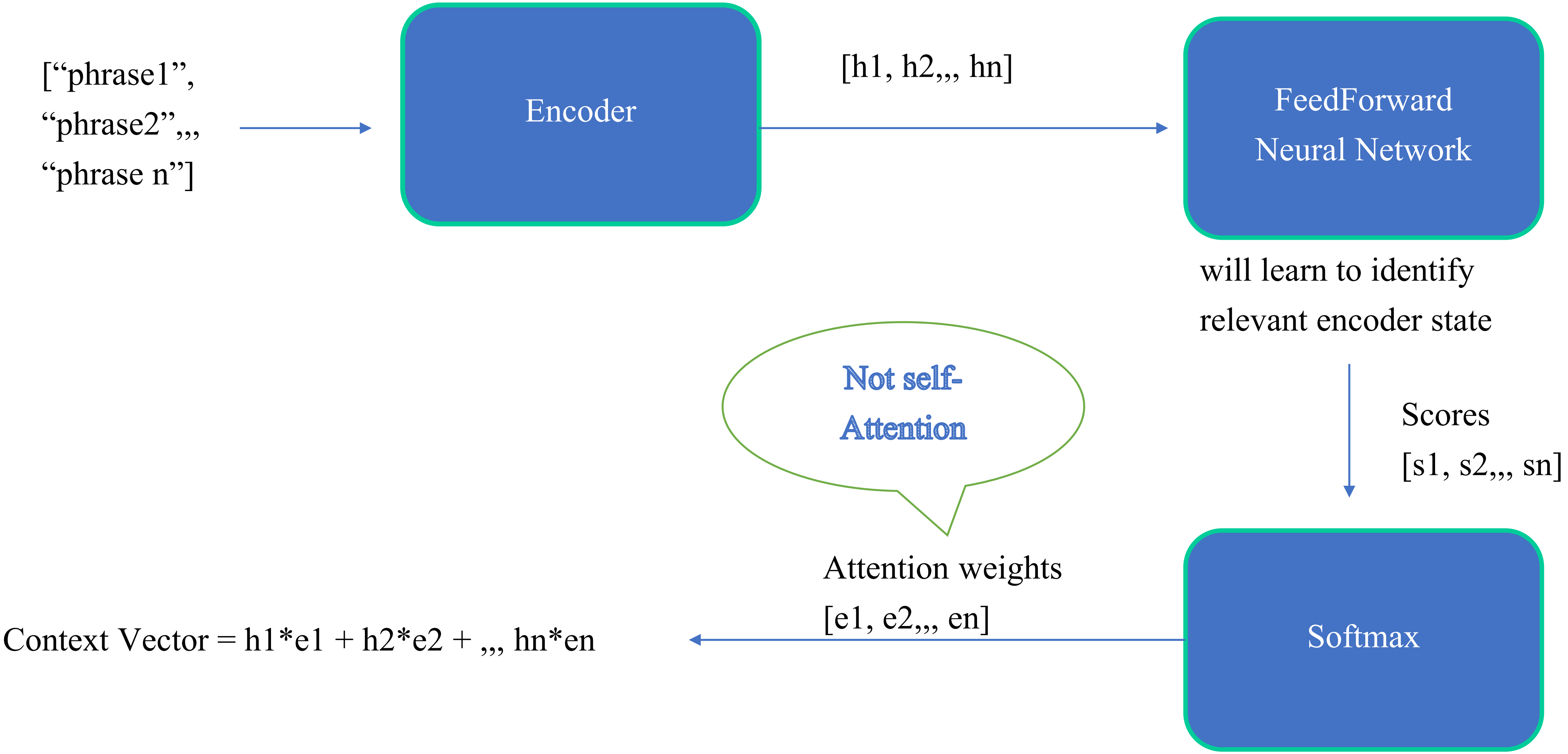


1	hiking trails in flat terrain	hiking	flat terrain	in flat terrain	
2	GET_ROUTE	TRAVEL_METHOD	TOPOGRAPHIC	WITHIN	
3	public parks with a lake or river inside	public parks	lake river	with a lake or river	inside
4	GET_PLACE	CATEGORY	TOPOGRAPHIC	NEAR	WITHIN
5	green space along the water	green space	water	along the water	
6	GET_PLACE	CATEGORY	TOPOGRAPHIC	NEAR	
7	residential neighborhoods overlooking downt	residential neighborhoods	downtown		
8	GET_AREA	AREA	CONDITION		
9	camp sites in national parks	camp sites	national parks	in national parks	
10	GET_PLACE	CATEGORY	AREA	WITHIN	
11	rocky mountains ski resorts	rocky mountains	resorts		
12	GET_PLACE	AREA	CATEGORY		
13	public boat docks on seattle area lakes	public boat	seattle area lakes	on seattle area lakes	
14	GET_PLACE	CATEGORY	AREA	NEAR	
15	islands in puget sound	islands	puget sound	in puget sound	
16	GET_AREA	CATEGORY	POINT	WITHIN	
17	lakes in washington state that have islands	lakes	washington state	in washington state that have islands	
18	GET_PLACE	CATEGORY	AREA	WITHIN	
19	seafood restaurants along the seattle waterfro	seafood restaurants	seattle waterfront	along the seattle waterfront	
20	GET_PLACE	CATEGORY	RECOGNIZED_ENTITY	NEAR	

Seq2Seq model architecture



Attention weights vector



[“phrase1”,
“phrase2”,,,
“phrase n”]

Encoder

[h1, h2,,, hn]

FeedForward
Neural Network

will learn to identify
relevant encoder state

Scores
[s1, s2,,, sn]

Softmax

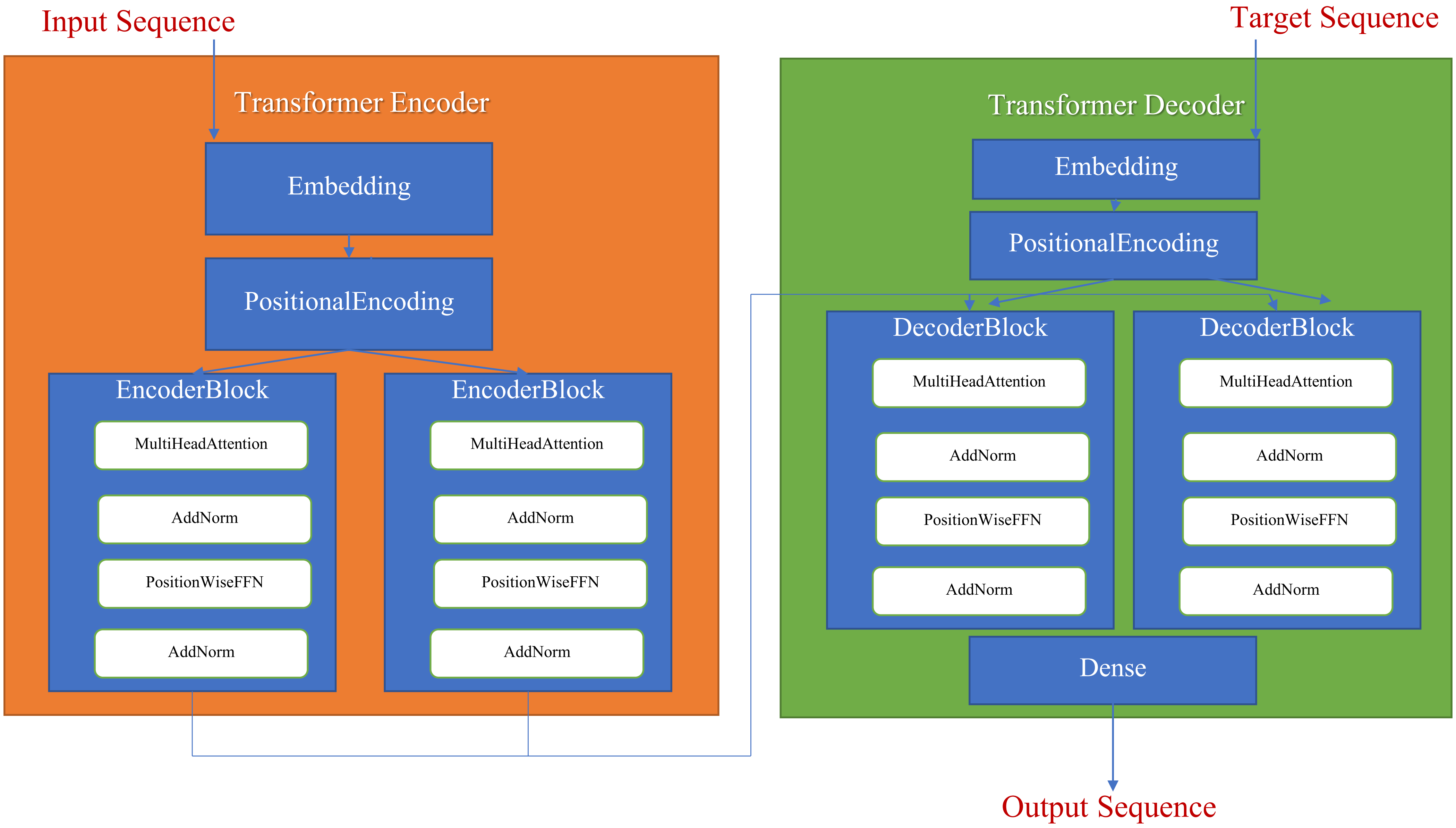
Attention weights
[e1, e2,,, en]

Not self-
Attention

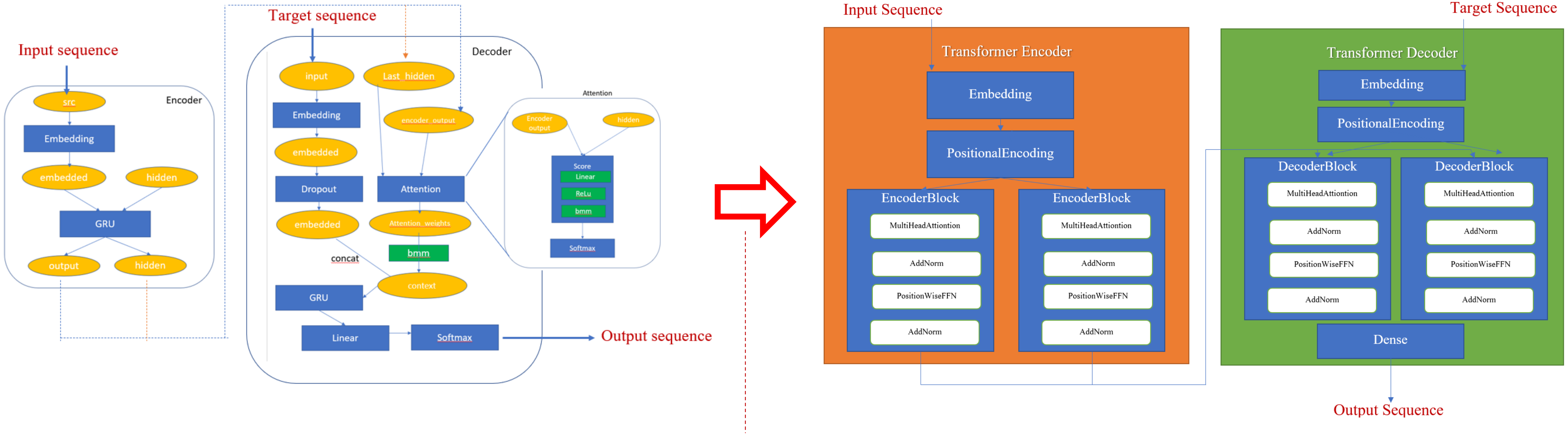
Context Vector = h1*e1 + h2*e2 + ,,, hn*en

Query to Intent-Slot Sequence Model

- The architecture follows the Multi-Head Attention model (a.k.a Transformer)



Query to Intent-Slot Sequence Model



Attention Weights vector → Self-Attention

Single Head → Multi-Head (9 heads)

Non-Positional → Positional Encoding

Query to Intent-Slot Sequence Model Result

- Training data : 300 Queries (it should be min 3000 queries, hopefully max 10k queries)
- Test data : 20 Alexa queries

- Further enhancement

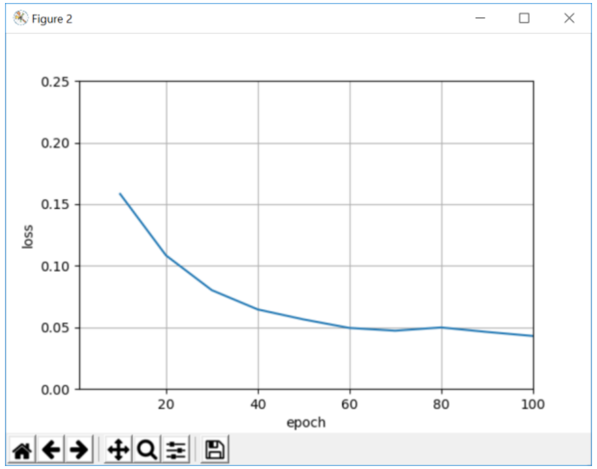
  1. More heads and more transformer blocks

     (cf. BERT 24 transformer blocks, embedding dimension is 1024, 340M parameters)

     (cf. GPT2 48 transformer blocks and sequence length 1024,

     and embedding dimension 1600, 1.5B parameters)

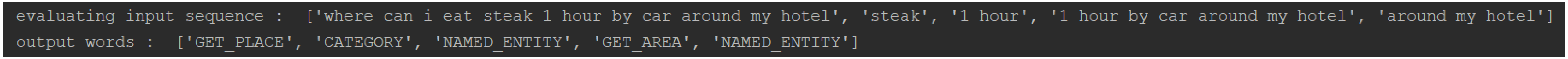
  2. Bidirectional Encoding



Result Comparison

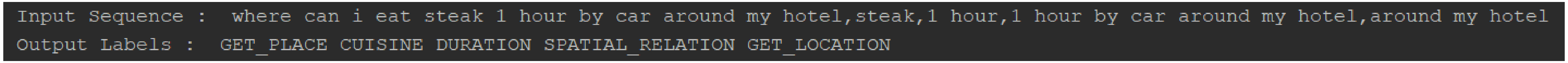
- Query : “Where can I eat steak 1 hour by car around my hotel”
- Expected labels : GET_PLACE, CUISINE, DURATION, SPATIAL_RELATION, GET_LOCATION

- Encoder-Decoder model



```
evaluating input sequence :  ['where can i eat steak 1 hour by car around my hotel', 'steak', '1 hour', '1 hour by car around my hotel', 'around my hotel']
output words :  ['GET_PLACE', 'CATEGORY', 'NAMED_ENTITY', 'GET_AREA', 'NAMED_ENTITY']
```

- Multi-Head Attention

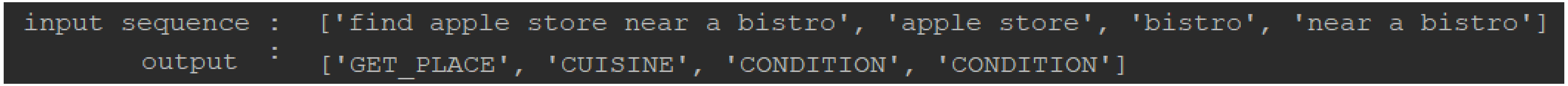


```
Input Sequence :  where can i eat steak 1 hour by car around my hotel,steak,1 hour,1 hour by car around my hotel,around my hotel
Output Labels :  GET_PLACE CUISINE DURATION SPATIAL_RELATION GET_LOCATION
```

Result Comparison

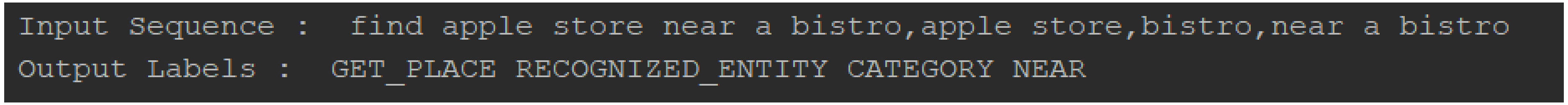
- Query : “Find apple store near a bistro”
- Expected labels : GET_PLACE, RECOGNIZED_ENTITY,CATEGORY,SPATIAL_RELATION

- Encoder-Decoder model



```
input sequence :   ['find apple store near a bistro', 'apple store', 'bistro', 'near a bistro']
        output :   ['GET_PLACE', 'CUISINE', 'CONDITION', 'CONDITION']
```

- Multi-Head Attention



```
Input Sequence :  find apple store near a bistro,apple store,bistro,near a bistro
Output Labels :  GET_PLACE RECOGNIZED_ENTITY CATEGORY NEAR
```

Alexa Query Example :

“HERE, where can I eat hot dogs 10 minutes by public transit around my location”

Raw query :

where can i eat hot dogs 10 minutes by public transit around my location

Input Phrase Sequence (by Phrase Splitter) :

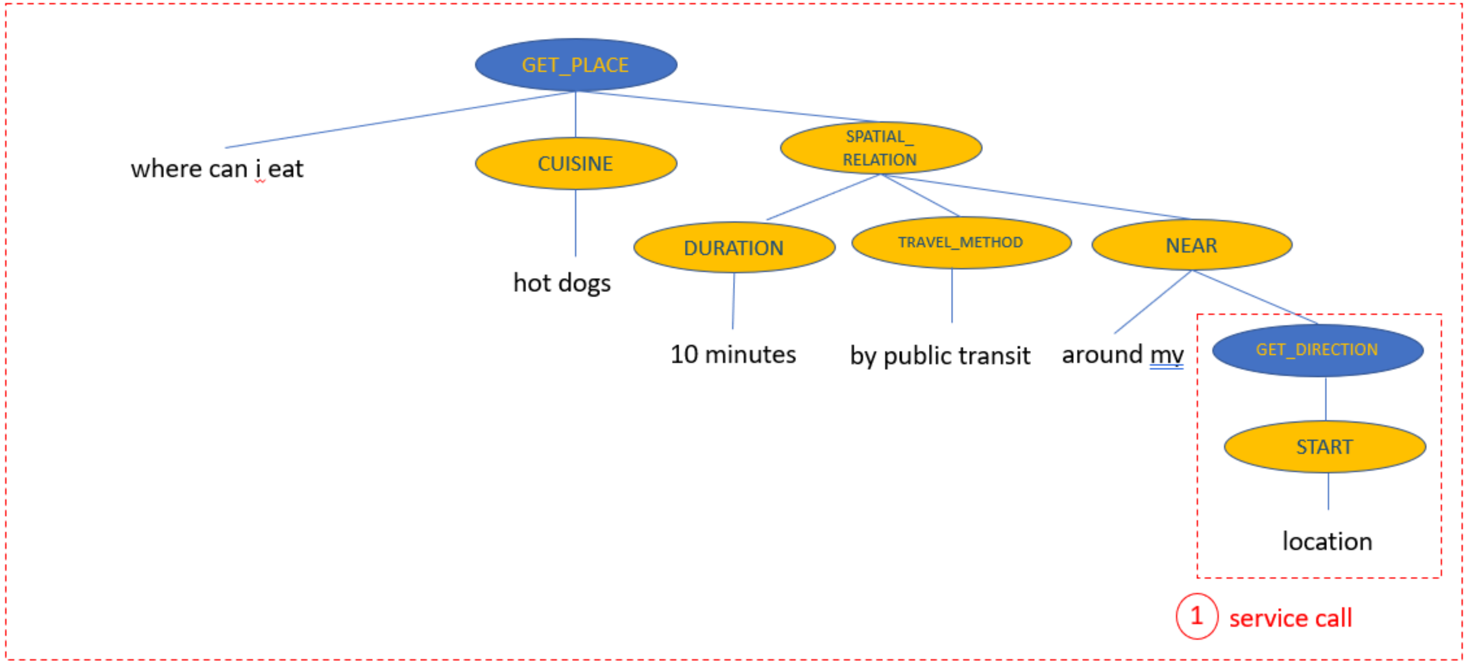
where can i eat hot dogs 10 minutes by public transit around my location, hot dogs, 10 minutes, 10 minutes by public transit around my location, public transit, around my location, location

Labels (Annotations) with Input sequence :

GET_PLACE[where can i eat CUISINE[hot dogs] SPATIAL_RELATION[DURATION[10 minutes] TRAVEL_METHOD[by public transit] NEAR[around my GET_DIRECTION[START[location]]]]

Output Sequence by Seq2Seq model :

GET_PLACE, CUISINE, SPATIAL_RELATION, DURATION, TRAVEL_METHOD, NEAR, GET_DIRECTION, START



2 service call

{ GET_PLACE (CUISINE : "hot dogs") SPATIAL_RELATION: "WITHIN", { GET_DIRECTIONS (START: { GET_LOCATION(SPATIAL_RELATION: "around", Context.Location}, DURATION: "10 minutes"), METHOD_TRAVEL: "public transit" } }

Alexa Query Example :

"HERE, find me an EV charging station close to a Starbucks along my route"

Raw query :

find me an ev charging station close to a starbucks along my route

Input Phrase Sequence (by Phrase Splitter) :

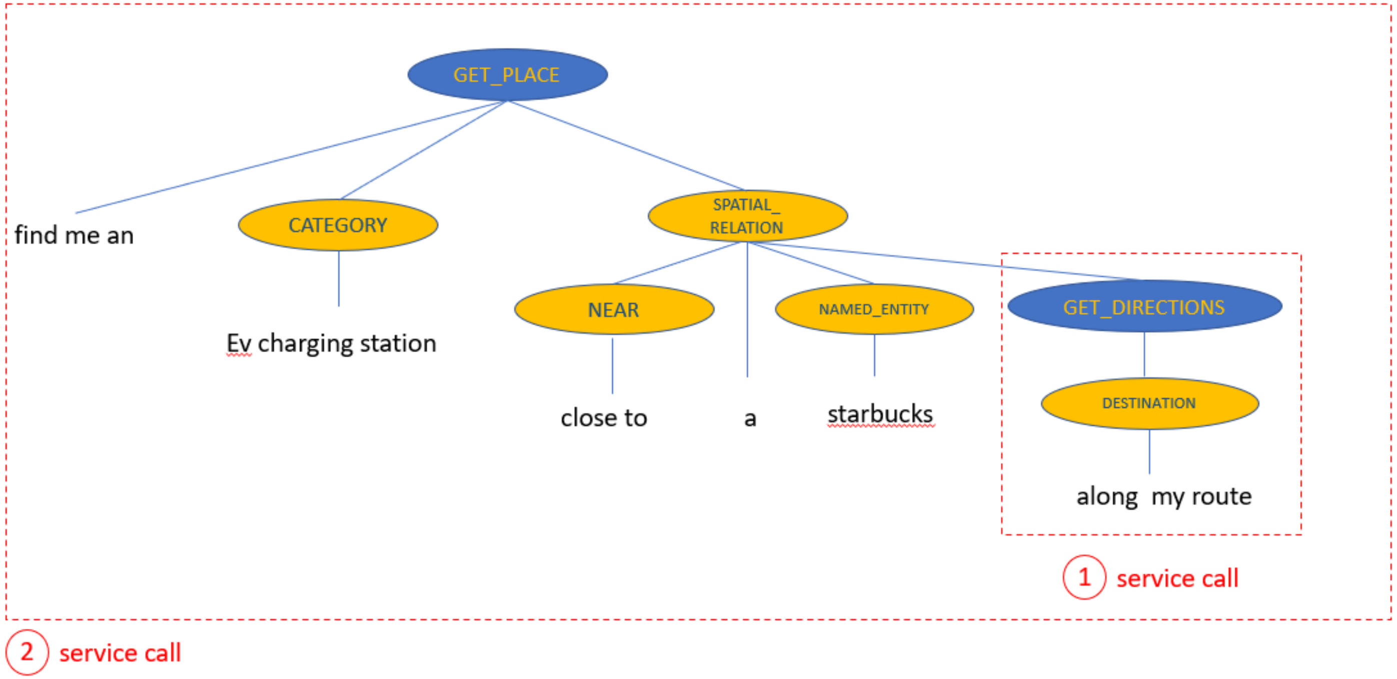
find me an ev charging station close to a starbucks along my route,ev,charging,station close,starbucks,along my route,route

Labels (Annotations) with Input sequence :

GET_PLACE[find me an CATEGORY[ev charging station] SPATIAL_RELATION[NEAR[close to] a NAMED_ENTITY[starbucks] GET_DIRECTIONS[along my DESTINATION[route]]]]

Output Sequence by Seq2Seq model :

GET_PLACE, CATEGORY, SPATIAL_RELATION, NEAR, NAMED_ENTITY, GET_DIRECTIONS, DESTINATION



{ GET_PLACE (PLACE_CATEGORY: "EV charging station"), SPATIAL_RELATION: "close to" { GET_PLACE(CHAIN:"Starbucks") }, SPATIAL_RELATION: "ALONG" { GET_DIRECTIONS(START: Context.Location, DESTINATION: Context.Home) } }

Thank you for your Attention!

Self Attention

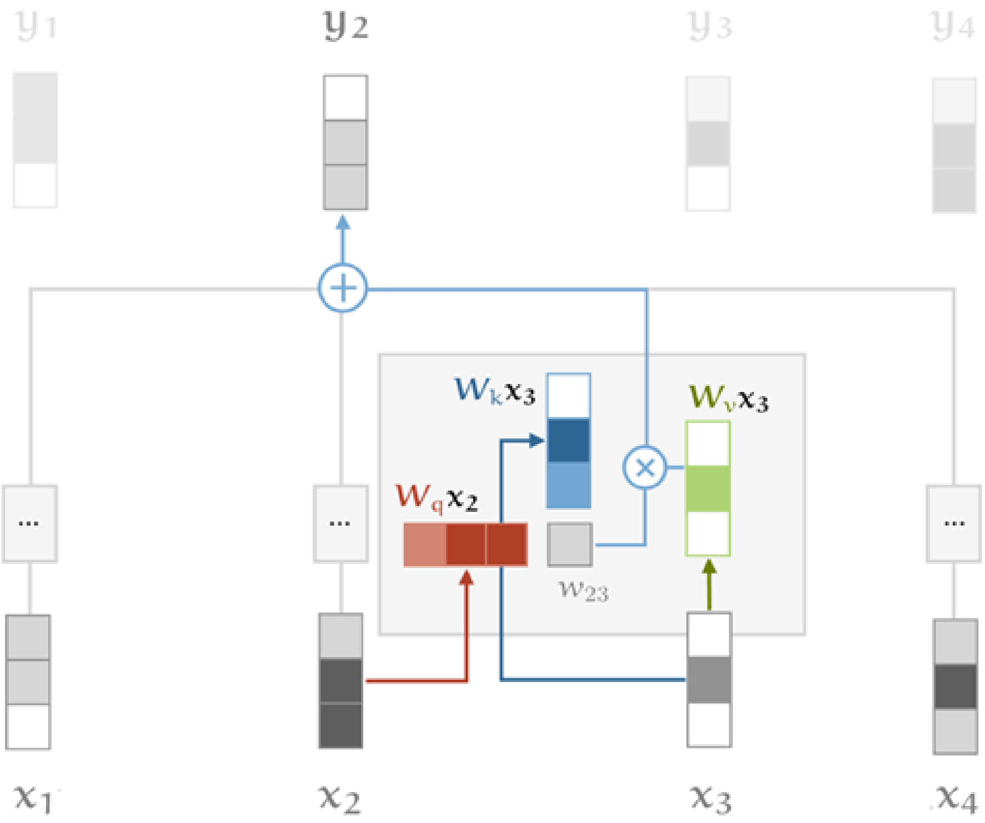
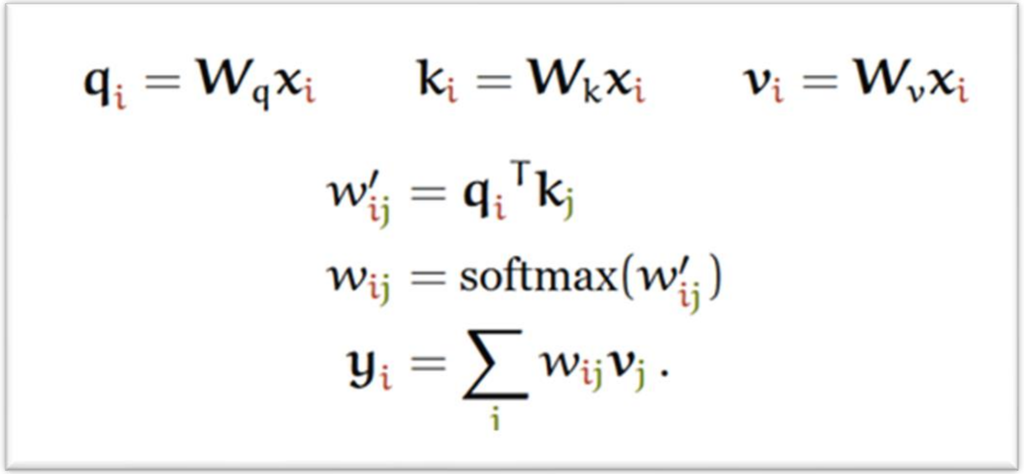


Illustration of the self-attention with key, query and value

K x K matrices W_q, W_k and W_v

Compute three linear transformations of each X_i



$$q_i = W_q x_i \qquad k_i = W_k x_i \qquad v_i = W_v x_i$$

$$w'_{ij} = q_i{}^T k_j$$

$$w_{ij} = \text{softmax}(w'_{ij})$$

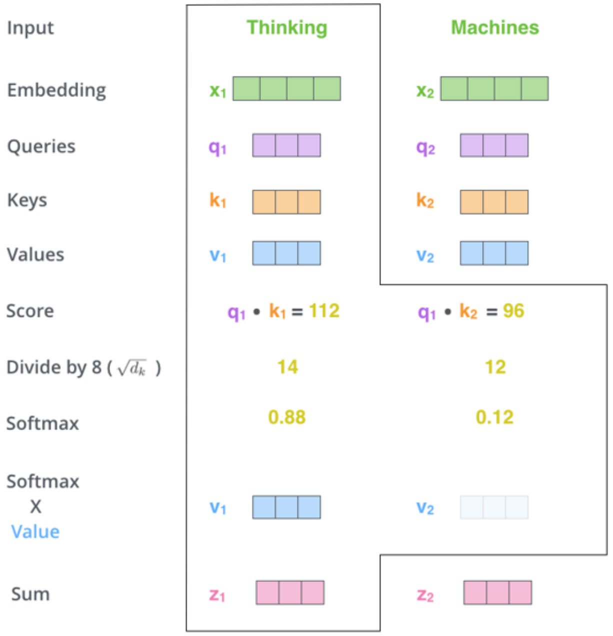
$$y_i = \sum_i w_{ij} v_j \, .$$

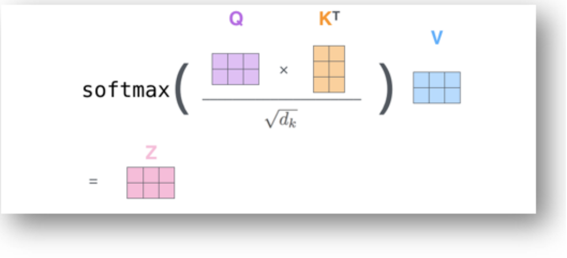
- **Query** : Established weights by comparison between X_i and all other vectors for Y_i (its own output of X_i)
- **Key** : Established weights by comparison between X_i and all other vectors for Y_j (j-th output,i.e. other output)
- **Value** : Used as part of weighted sum to compute each output vector once the weights has been established

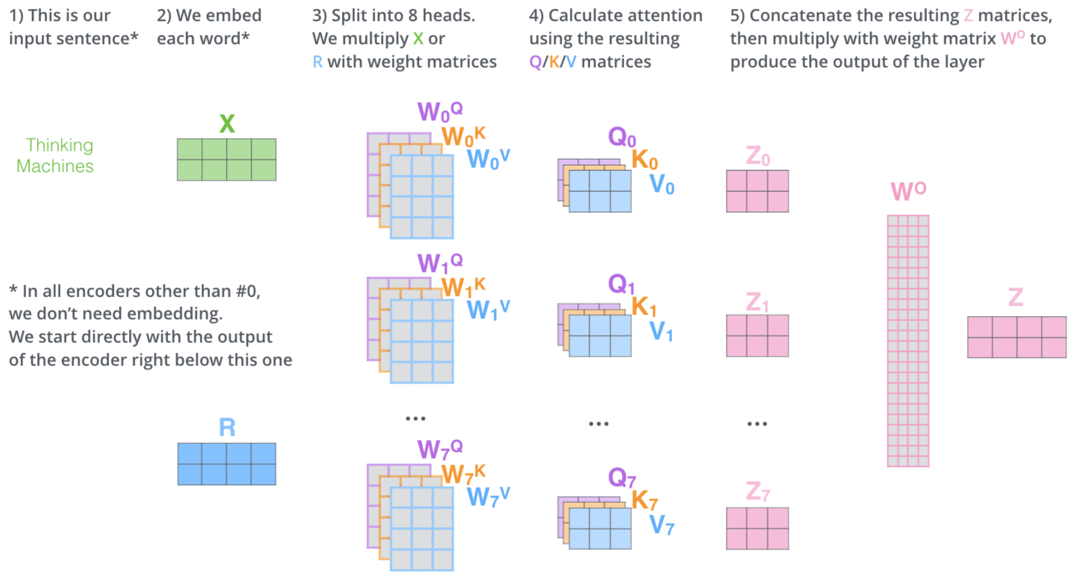
Illustration from http://www.peterbloem.nl/blog/transformers

MultiHead Self Attention & Positional Encoding

- Permutation equivalent : Self-Attention see the input sequence as a set not a sequence.
- the self-attention by itself ignores the sequential nature of the input
- MultiHead Attention works better than Single Attention






1) This is our input sentence*
2) We embed each word*
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix WO to produce the output of the layer
Thinking
Machines
X
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
R
W0Q
W0K
W0V
W1Q
W1K
W1V
W7Q
W7K
W7V
Q0
K0
V0
Q1
K1
V1
Q7
K7
V7
Z0
Z1
Z7
...
WO
Z

https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/