

The logo for 'here' is a dark blue circle containing the word 'here' in white lowercase letters, with a small teal triangle pointing upwards to the left of the 'h'.

here

*“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

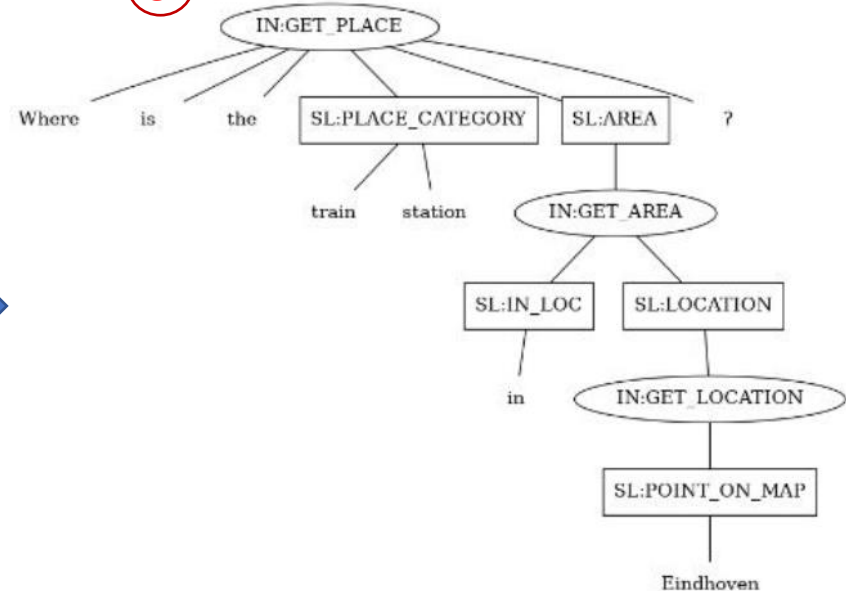
1

“Where is the train station in Eindhoven?”

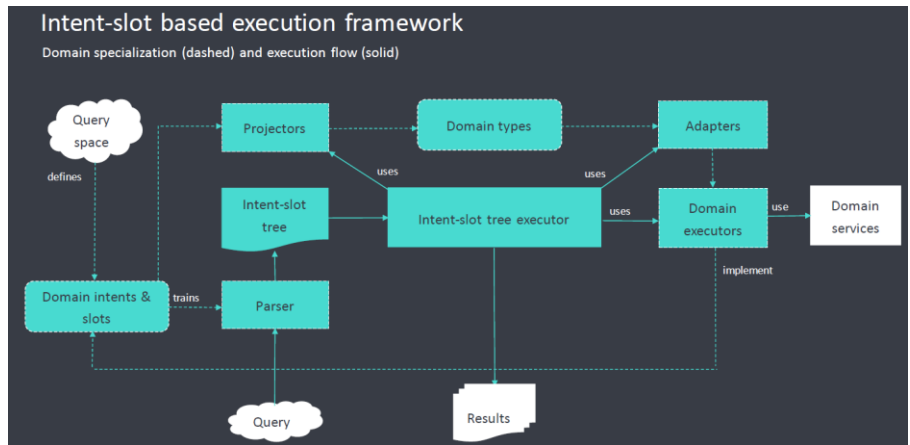
2



3



4



5



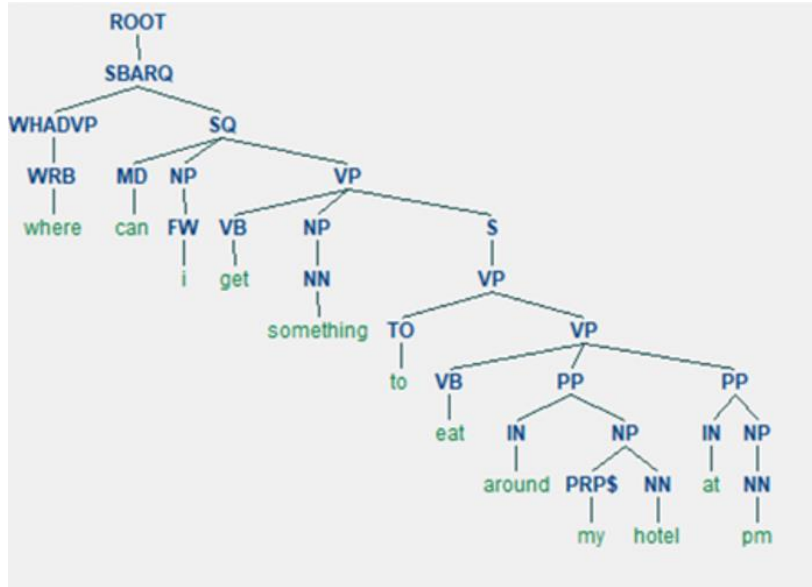
Query results

	name	id
0	Station Eindhoven-Beukenlaan	here:pds:place:528u15us-105fe2e50fb34db6ae1f9c9efd748b25
1	Eindhoven Strijp-5	here:pds:place:528u15us-3eba350e3d454ce381cf03134ce4120a
2	Station Eindhoven Strijp-5	here:pds:place:528u15us-c8469e73f7f94ab2b2bcc1240cb530e

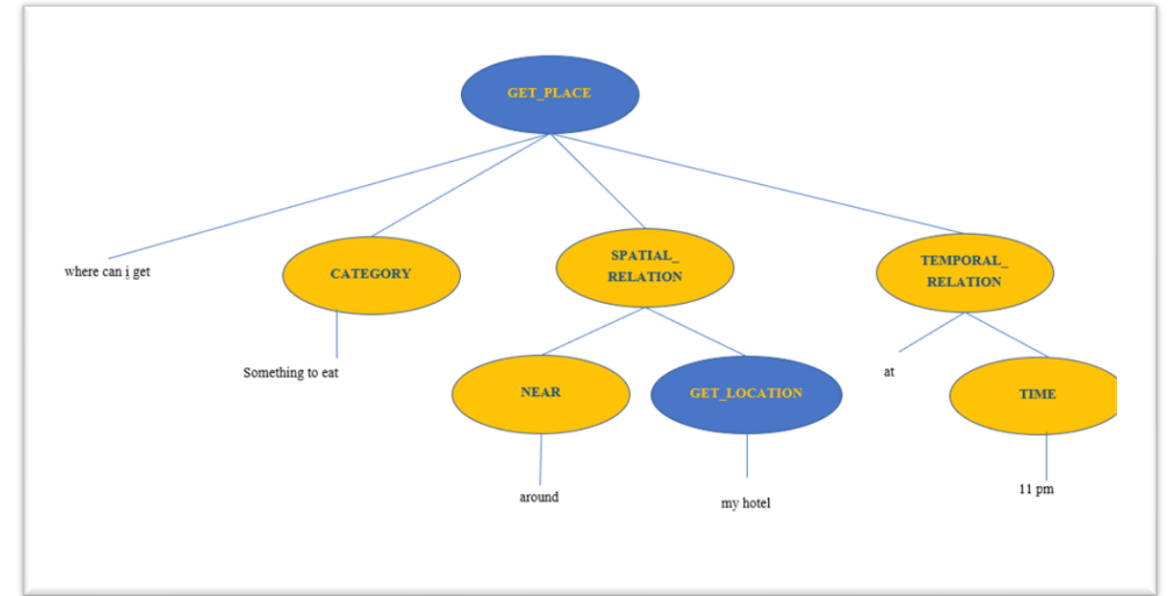


Let's re-think about parsing

Syntactic : Constituency Parse Tree



Semantic : Intent-Slot based Parse Tree



- Semantic Query Parsing problem as Language Translation problem

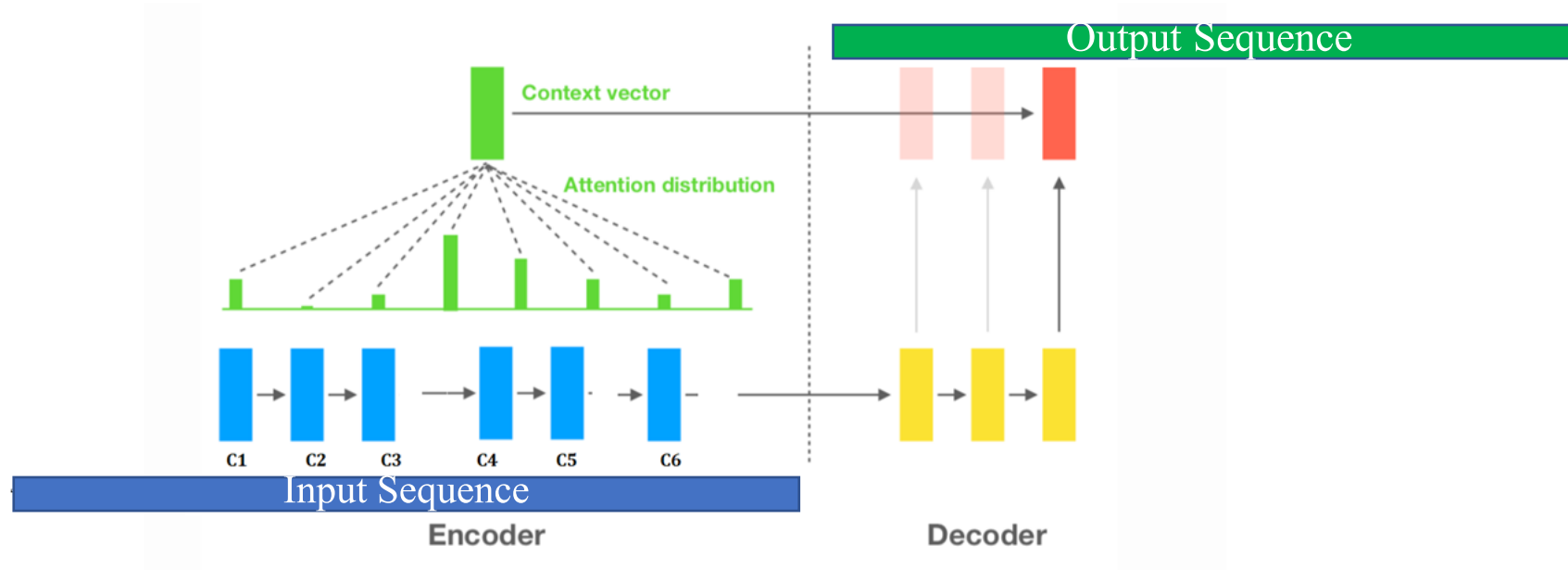
Query (Natural Language)



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 :

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

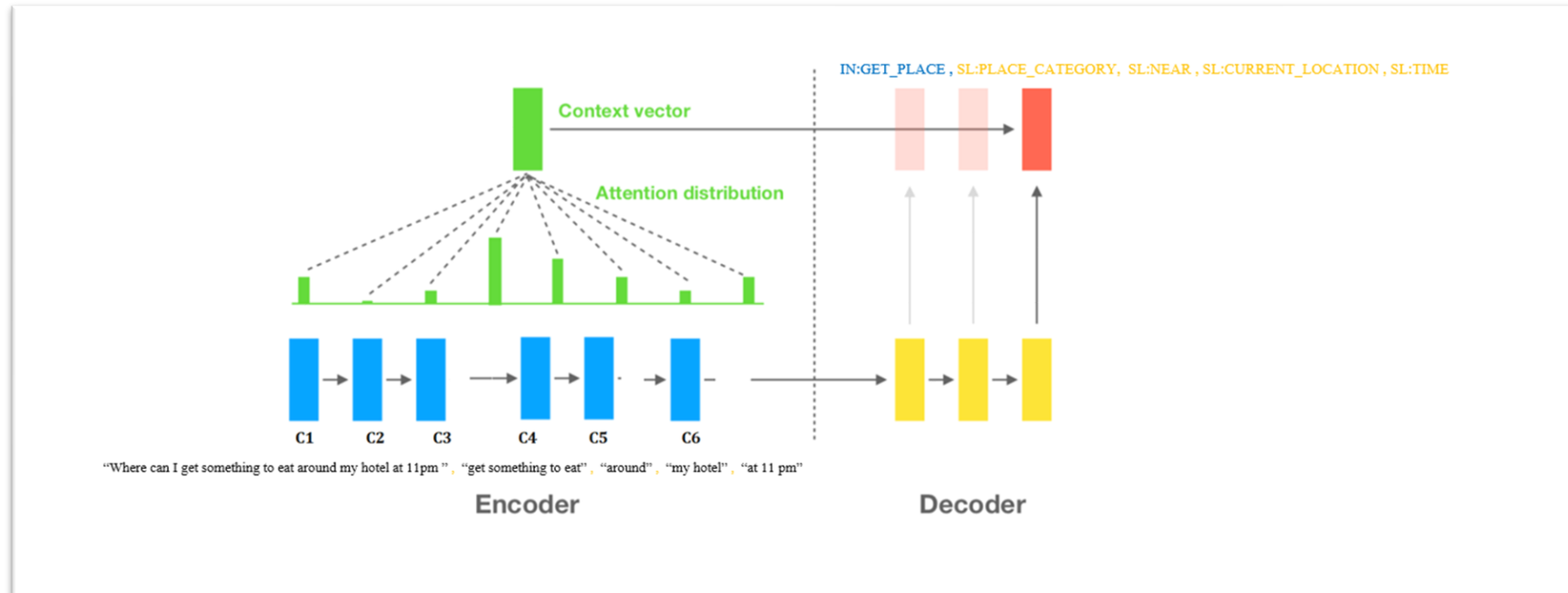
“where can i get something to eat around my hotel at 11 pm”, “something to eat”, “around my hotel”, “around”, “my hotel”, “at 11 pm”, “11pm”

Output Sequence :

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



Not a Sequence of Token



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”



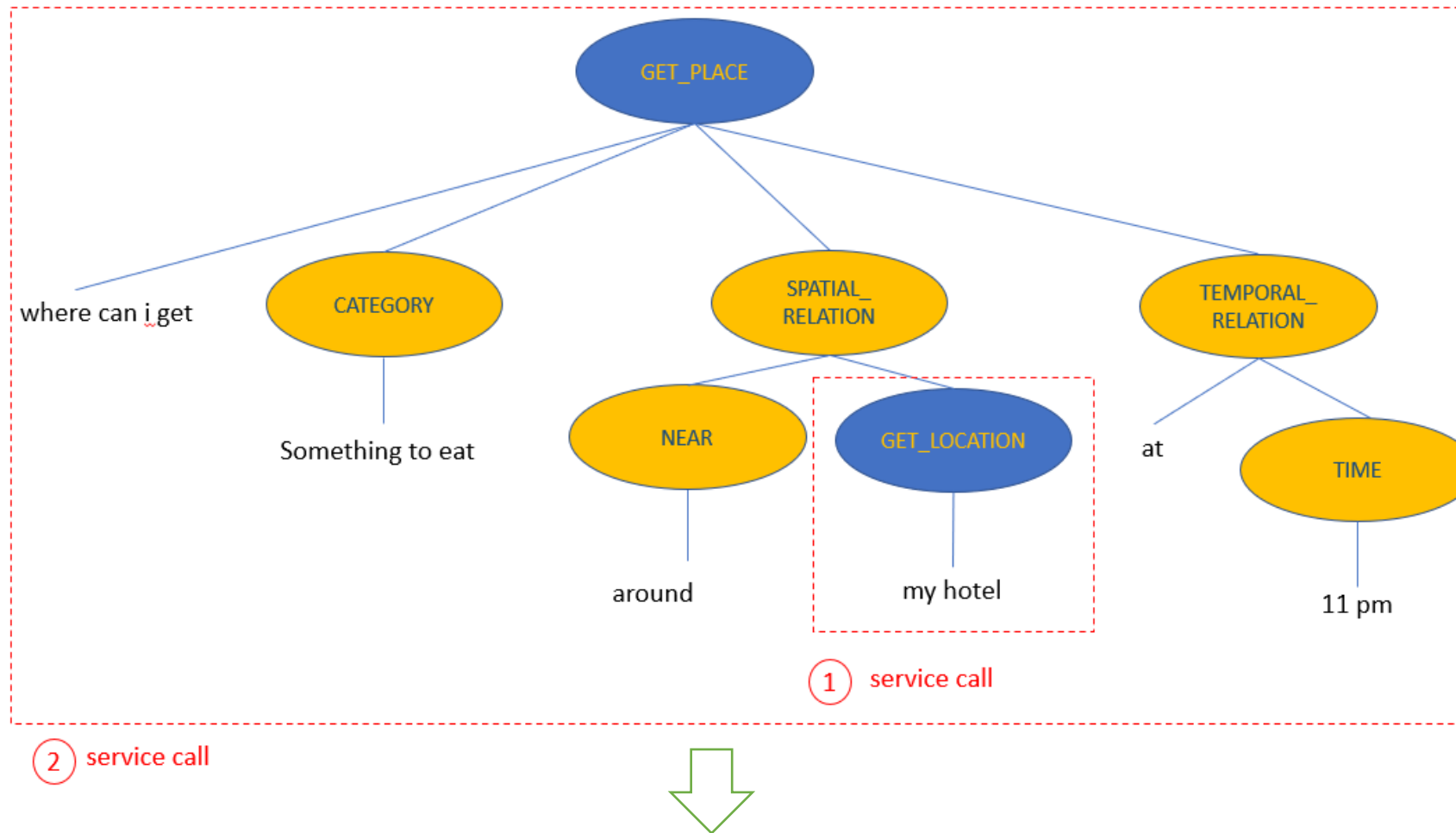
[Intent-Slot Labels]

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 (blue oval) map to Service type
- Slots (yellow oval) annotate phrases within query and map to parameters (modifiers) of Service



i 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-)

Constituent Tag Level

Clause Level

Phrase Level

Word Level

Clause Level

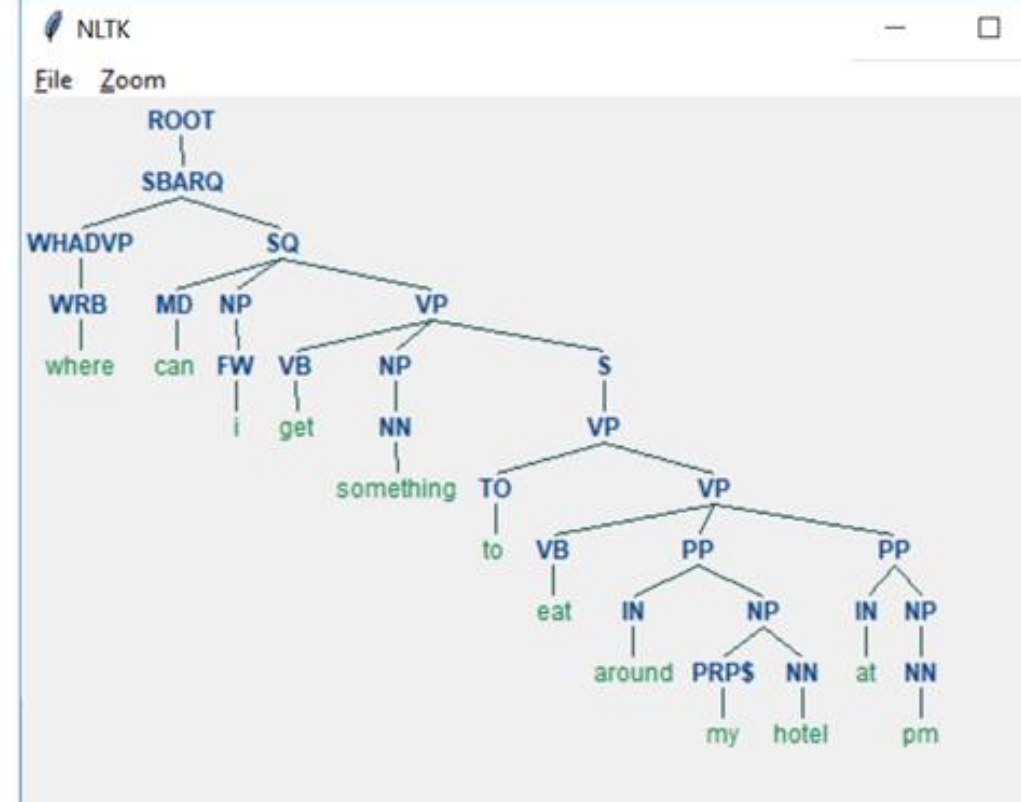
S - simple declarative clause, i.e. one that is not introduced by a (possibly empty) subordinating conjunction or a *wh*-word and that does not exhibit subject-verb inversion.
 SBAR - Clause introduced by a (possibly empty) subordinating conjunction.
 SBARQ - Direct question introduced by a *wh*-word 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 pronominal 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 - Verb 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
 JKR - Adjective, comparative
 JNS - 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
 PRPS - 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



```

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

```

1 hiking trails in flat terrain,hiking,flat terrain,in flat terrain
2 public parks with a lake or river inside,public parks,lake river,with a lake or river,inside
3 green space along the water,green space,water,along the water
4 residential neighborhoods overlooking downtown,residential neighborhoods,downtown
5 camp sites in national parks,camp sites,national parks,in national parks
6 rocky mountains ski resorts,rocky mountains,resorts
7 public boat docks on seattle area lakes,public boat,seattle area lakes,on seattle area lakes
8 islands in puget sound,islands,puget sound,in puget sound
9 lakes in washington state that have islands,lakes,washington state,in washington state that have islands
10 seafood restaurants along the seattle waterfront,seafood restaurants,seattle waterfront,along the seattle waterfront
11 gas stations along my drive to yakima tomorrow,gas stations,drive,along my drive to yakima tomorrow
12 parking in downtown seattle away from the marathon route,parking,downtown,in downtown,away,marathon route,from the marathon route
13 ballard farmersa market,ballard,farmersa market
14 restaurants near the bellevue art fair,restaurants,bellevue art fair,near the bellevue art fair
15 what food trucks are near me at lunchtime,food trucks,near me,lunchtime,at lunchtime
16 residential neighborhoods with low average traffic speed,residential neighborhoods,low average traffic speed,with low average traffic speed
17 park ride along a bus route to downtown seattle,park ride,bus route,downtown seattle,along a bus route to downtown seattle
18 driving directions to langley avoiding the ferry,directions,ferry
19 where can i get something to eat near high street kensington station ?,something,high street kensington station
20 which cta stations in lakeview chicago have bike sharing stations ?,cta stations,lakeview chicago,in lakeview chicago,bike,stations
21 where can i drop off my bike near the south entrance to central park ?,bike,south entrance,near the south entrance,central park
22 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
    
```

MVP Queries

Sequences of Phrases

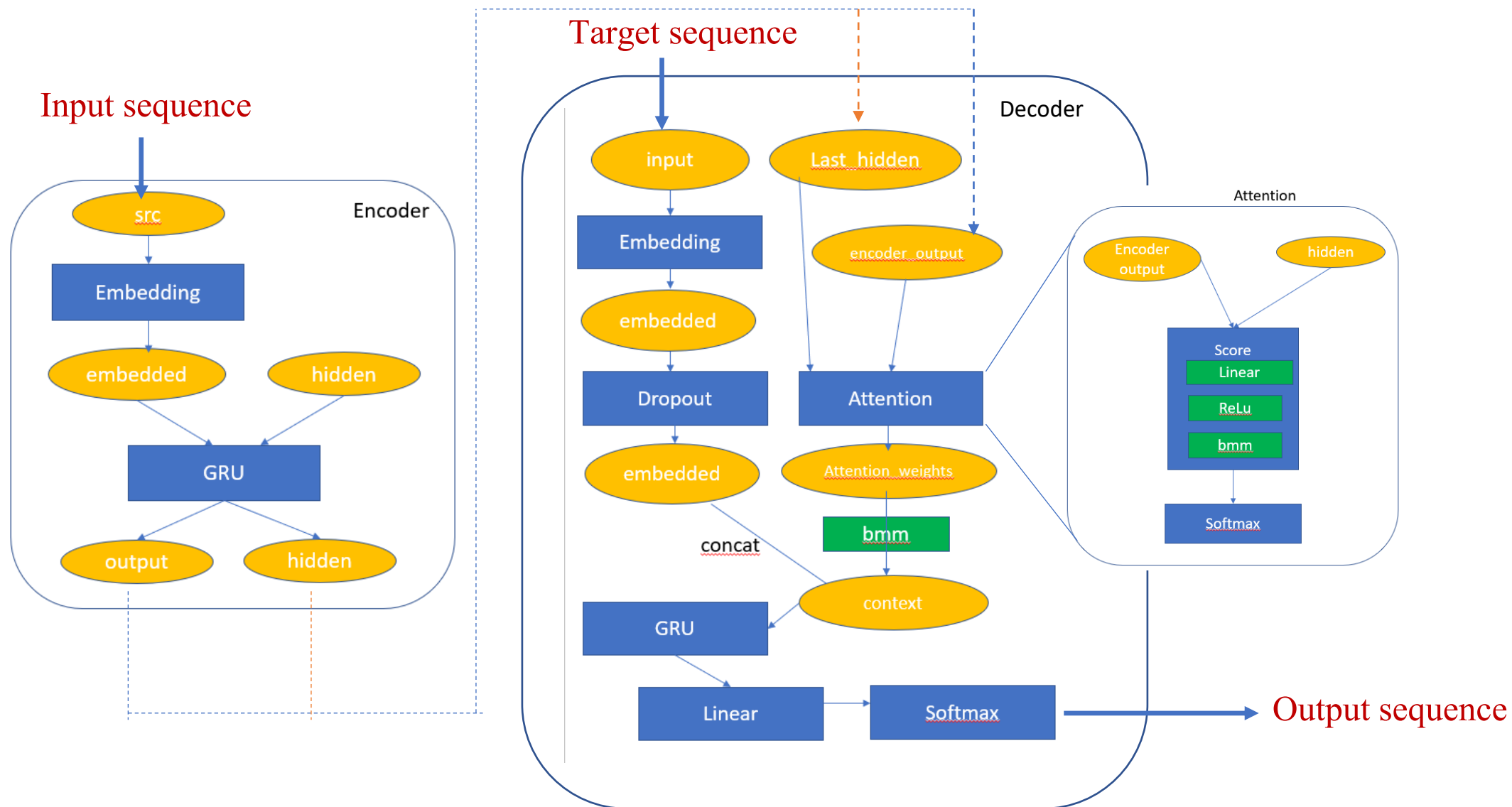
Training data

Input sequence

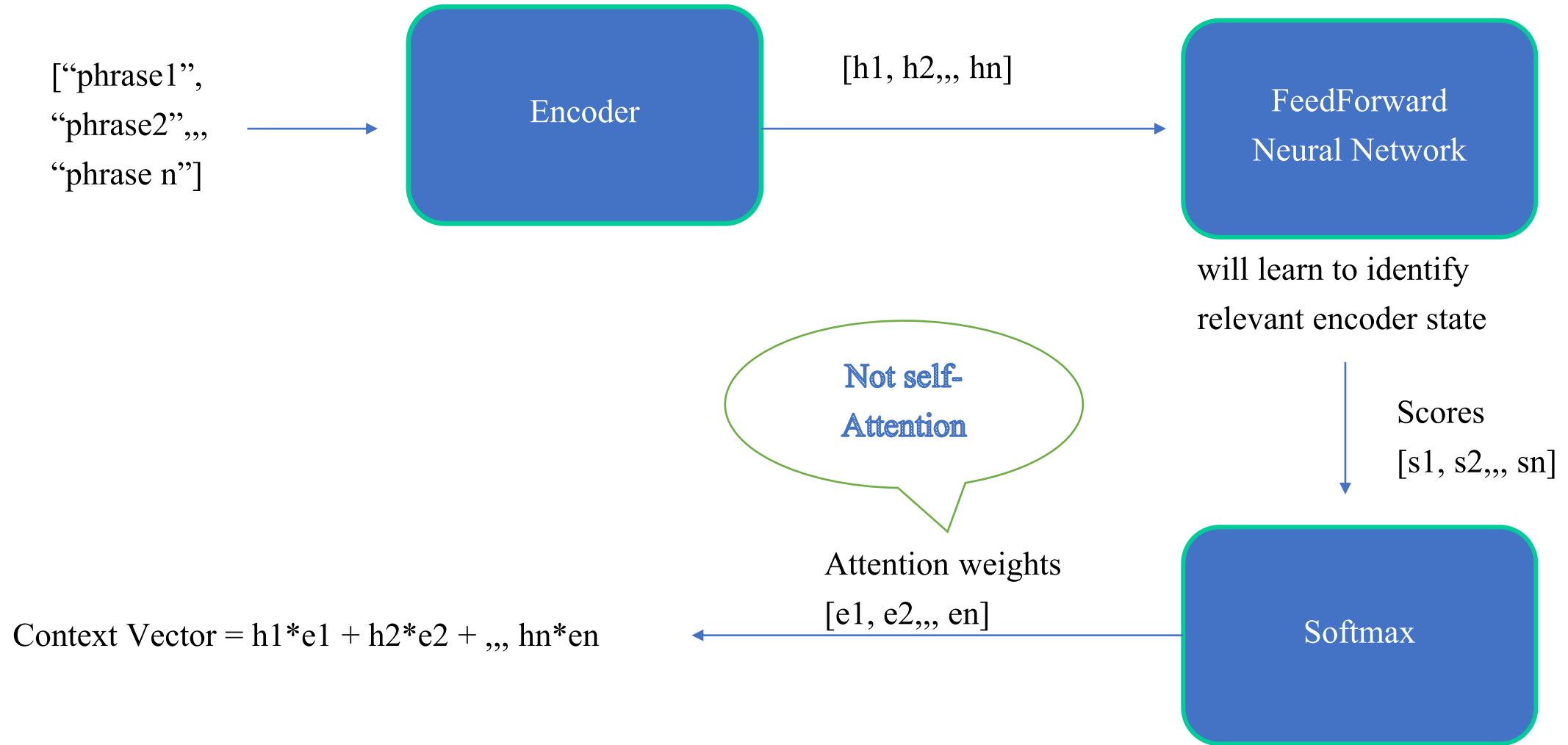
Output sequence

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 downtown	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 waterfront	seafood restaurants	seattle waterfront	along the seattle waterfront	
20	GET_PLACE	CATEGORY	RECOGNIZED_ENTITY	NEAR	

Seq2Seq model architecture

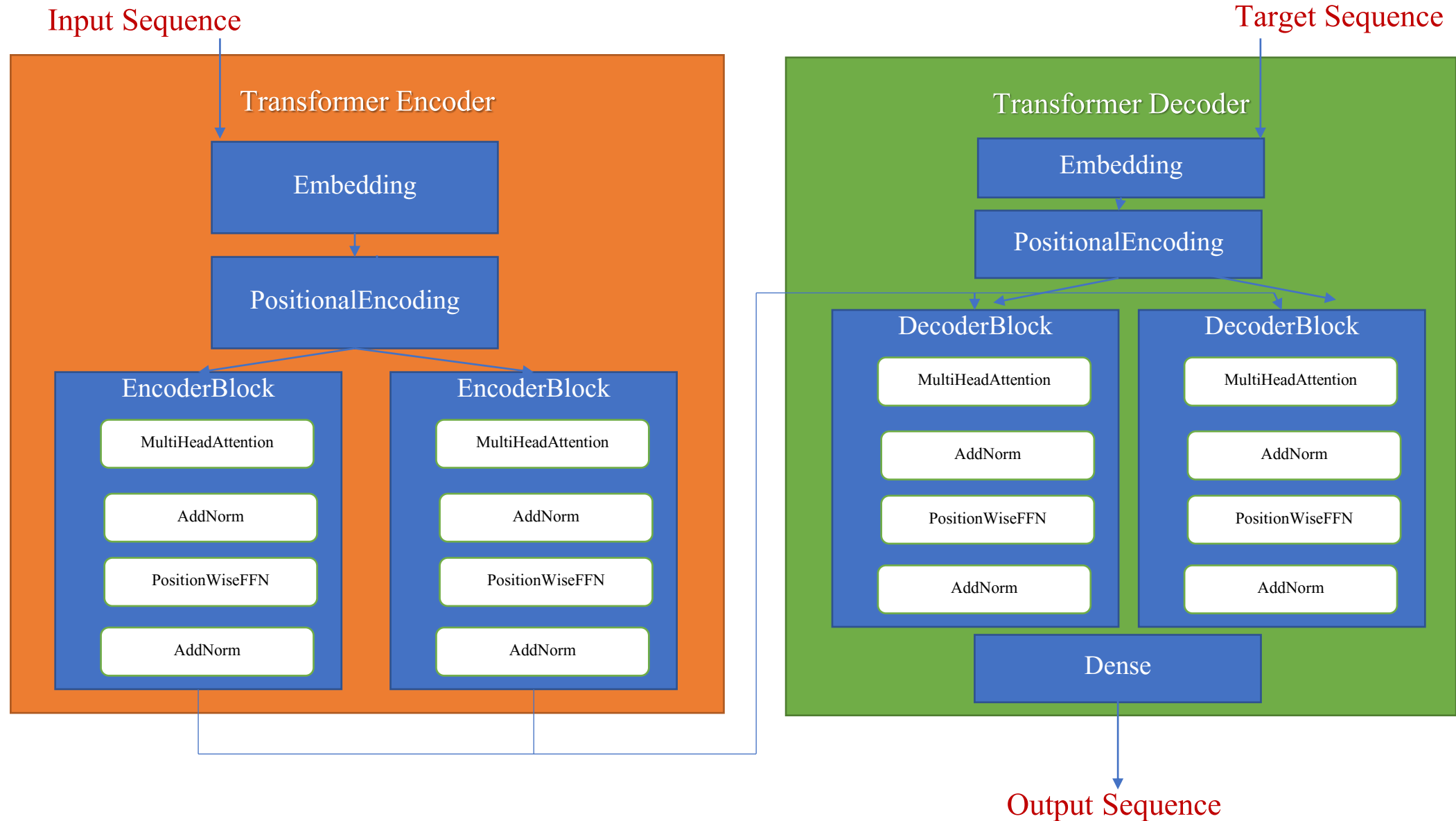


Attention weights vector

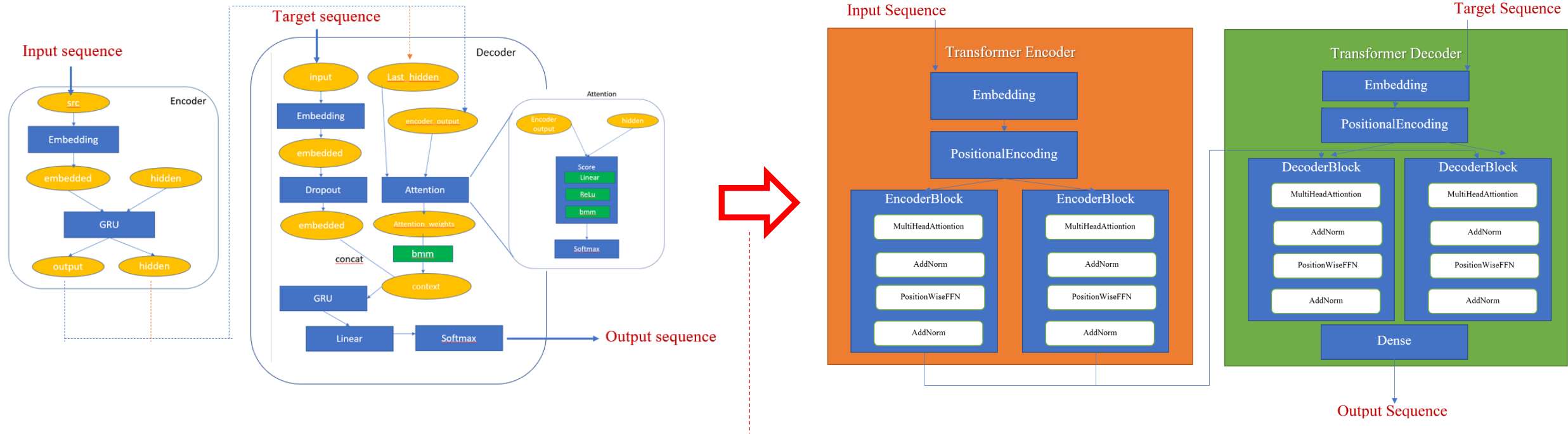


Query to Intent-Slot Sequence Model

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



Query to Intent-Slot Sequence Model

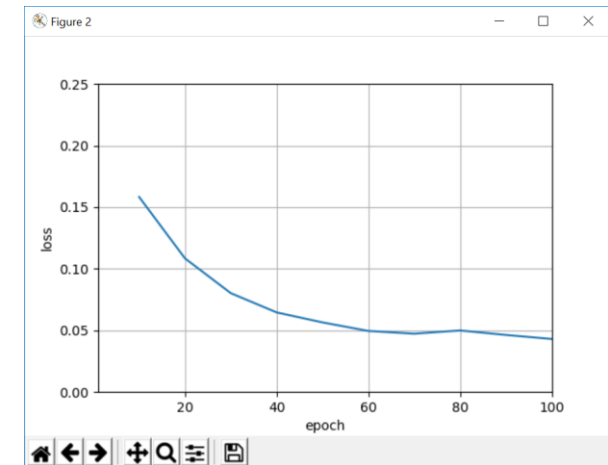


Attention Weights vector → Self-Attention
Single Head → Multi-Head (9 heads)
Non-Positional → Positional Encoding

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) :

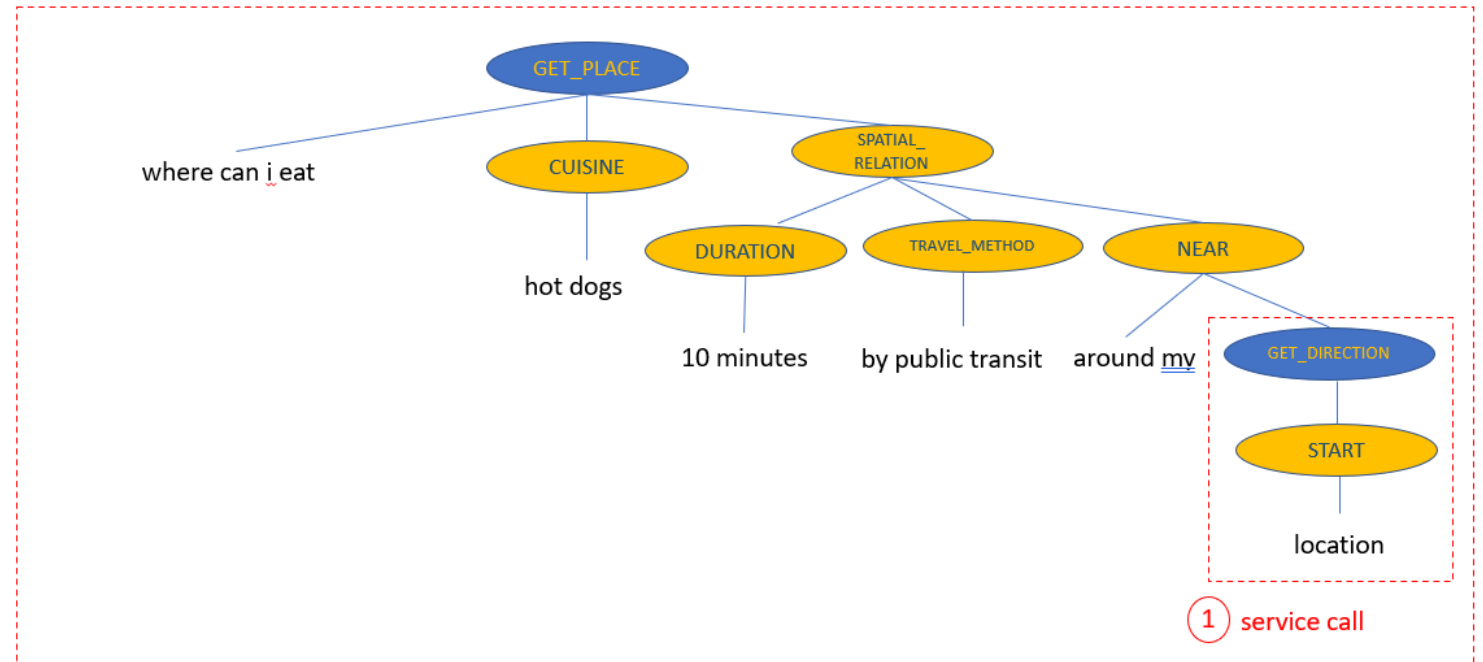
where can i eat hot dogs 10 minutes by public transit around my location, hot dogs, 10 minutes, 10 minutes by public transit around my location, public transit, around my location, location

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



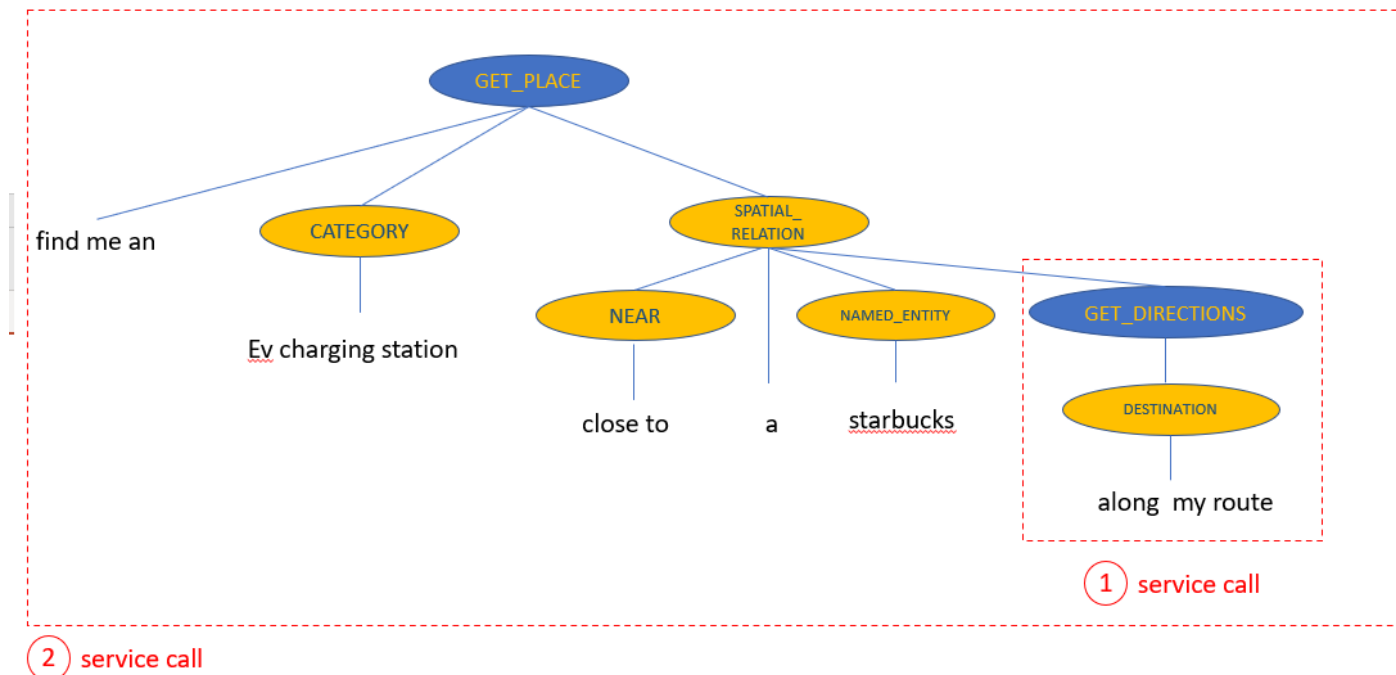
Labels (Annotations) with Input sequence :

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



$K \times K$ matrices W_q , W_k and W_v

Compute three linear transformations of each x_i

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

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

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

$$y_i = \sum_j w_{ij} v_j$$

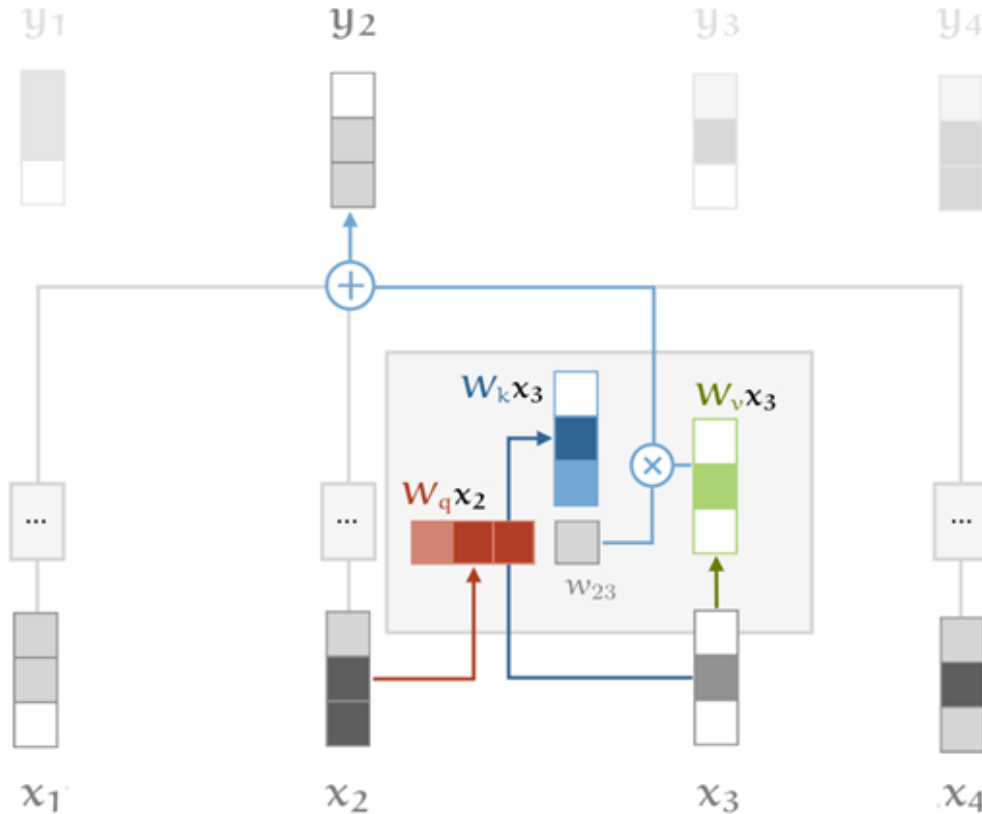


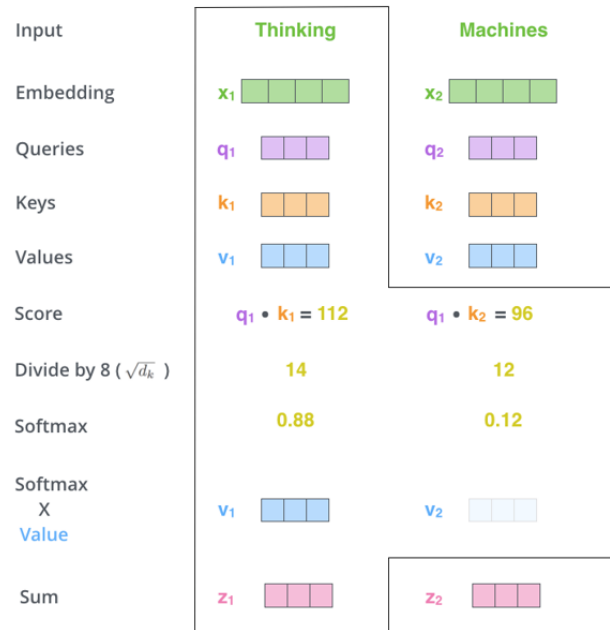
Illustration of the self-attention with **key**, **query** and **value**

- **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

MultiHead Self Attention & Positional Encoding

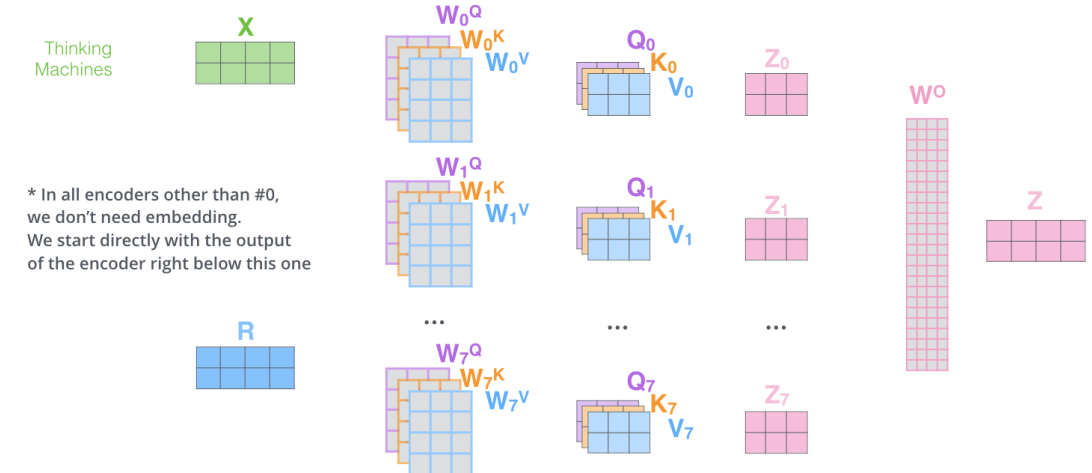


- Permutation equivalent : Self-Attention see the input sequence as a set not a sequence.
- the self-attention by itself ignores the sequential nature of the input
- MultiHead Attention works better than Single Attention



$$\text{softmax} \left(\frac{Q \times K^T}{\sqrt{d_k}} \right) V = Z$$

- 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 W^O to produce the output of the layer



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