Neuro-Symbolic Representations for IR
2.2 – Neural Text and Graph Representations

Laura Dietz

SIGIR

2023

Opportunities for Query-Specific Entity Representations

Context-aware representations of entities in text based on a query

- Query Understanding
  - Captures both inherent properties of entities
  - Models entity relevance to the query
- Retrieval and Ranking
  - Overcomes limitations of simple keyword matching
  - Uncovers deeper connections between entities, documents, and the query
- Answer Generation
  - Generates more coherent, contextually appropriate responses
  - Tailors output to the specific information need

Slide genrated by chatGPT 3.5

Opportunities for Query-Specific Entity Representations

Context-aware representations of entities in text based on a query

- Query Understanding
  - Captures both inherent properties of entities
  - Models entity relevance to the query
- Retrieval and Ranking
  - Overcomes limitations of simple keyword matching
  - Uncovers deeper connections between entities, documents, and the query
- Answer Generation
  - Generates more coherent, contextually appropriate responses
  - Tailors output to the specific information need

Lots of magical hand waving. Let's look at that!

Deep Learning Techniques

Demystify some magic :

- Metric Learning as Latent Space Projection
- Clustering, Retrieval = Metric Learning
- Attention = Metric Learning
- Transformers = Graph Neural Networks

Tutorial Timetable

Part 1: Knowledge Graphs and Entities

- 1.1 Welcome & Motivation
- 1.2 Knowledge Graphs and GPT
- 1.3 Entity Linking

Part 2: Neuro-Symbolic Foundations

- 2.1 Ranking Wikipedia Entities / Aspects
- 2.2 Neural Text Representations and Semantic Annotations ←
- 2.3 Infusion of Symbolic Knowledge into Text Representation

Part 3: Reasoning, Robustness, and Relevance

- 3.1 Denoising Dense Representations with Symbols
- 3.2 Reasoning about Relevance
- 3.3 From PRF to Retrieval Enhanced Generation

Part 4: Emerging Topics

- 4.1 Conclusion and Outlook
- 4.2 Panel Discussion

https://github.com/laura-dietz/neurosymbolic-representations-for-IR/SIGIR23/

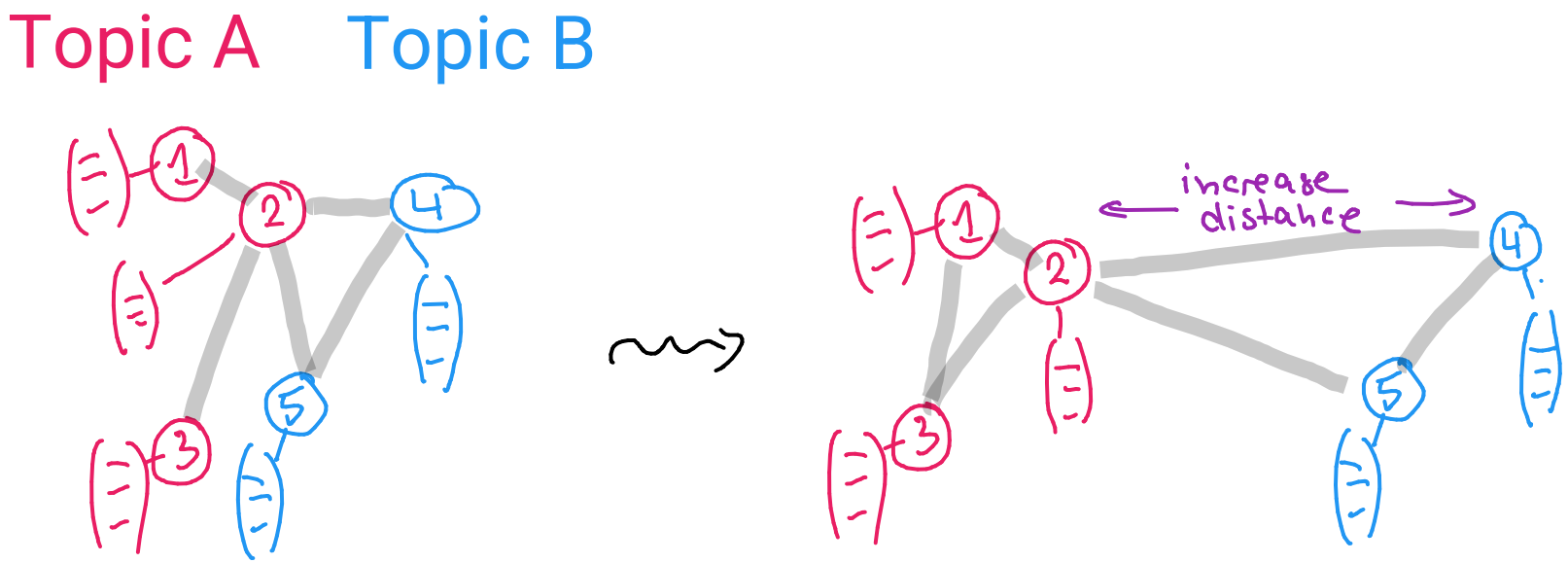
Outline

1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

Metric Learning for Topics

Task:

- Given data points (vectors), with true topic labels
- Project into vector space where points with
  - same labels are close
  - different labels are far apart

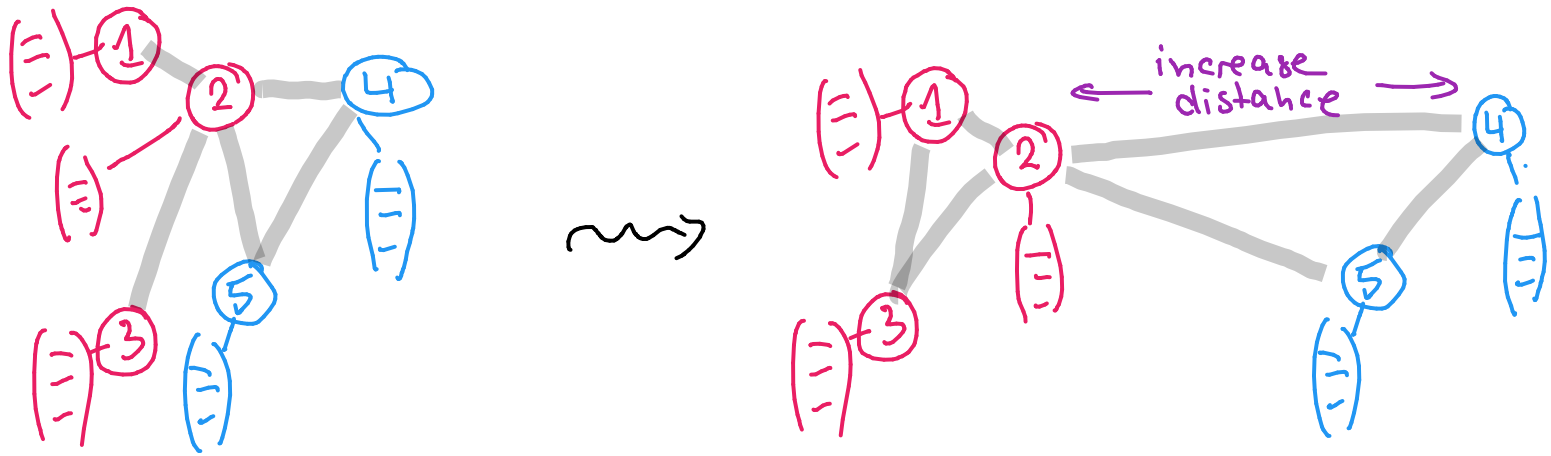


Train: latent space has "useful" representation of proximity.

Metric Learning as Neural Network

- Project each point $\mathbf{x}$ with a function $\texttt{proj}$.
  - $\mathbf{y} = \texttt{proj}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$
  - where $\mathbf{W}$ and $\mathbf{b}$ are trainable parameters.

Topic A Topic B



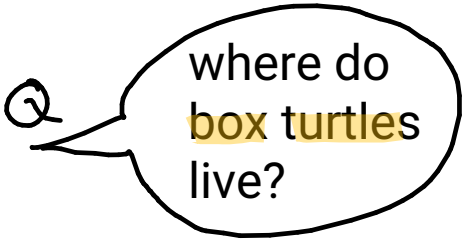
$\mathbf{x}$ $\texttt{proj}(\mathbf{x})$

Learn Projection function so that proximity $\approx$ same/different label

Outline

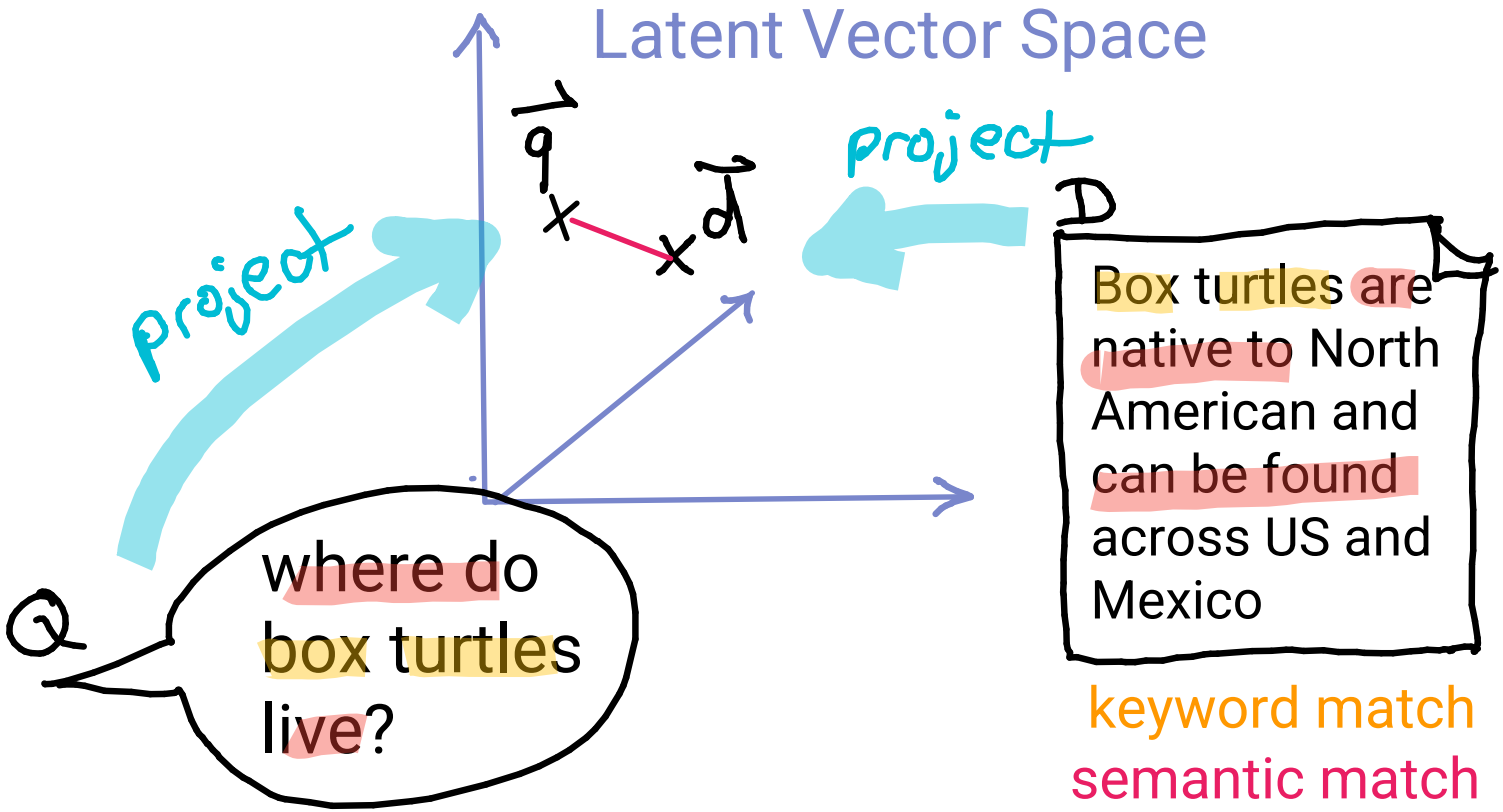
1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

Ad hoc Text Ranking



SOTA: ad hoc Text Ranking

Dense Retrievel Approach:

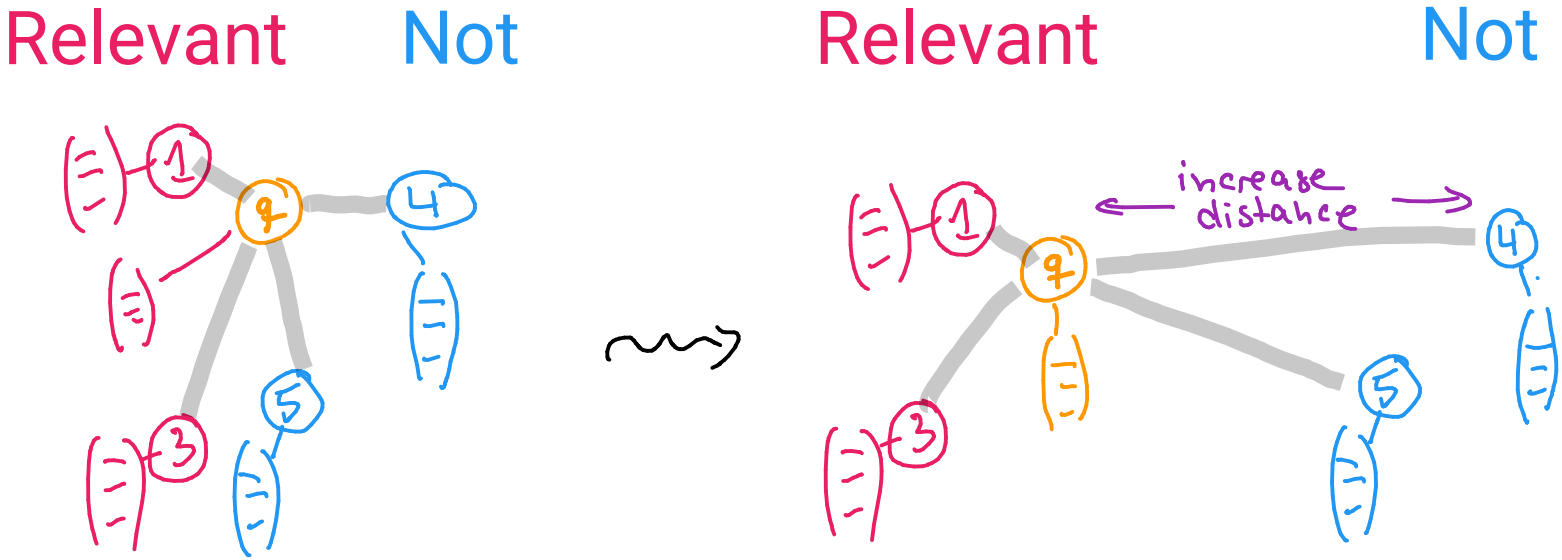


Latent space and projections are trained, so that $\mathbf{q}$ and $\mathbf{d}$ are close whenever documents are relevant for the query.

Ranking as Metric Learning

Task:

- Given query and document points, with relevance labels
- Project into vector space where
  - relevant points close to query
  - not relevant points are far away

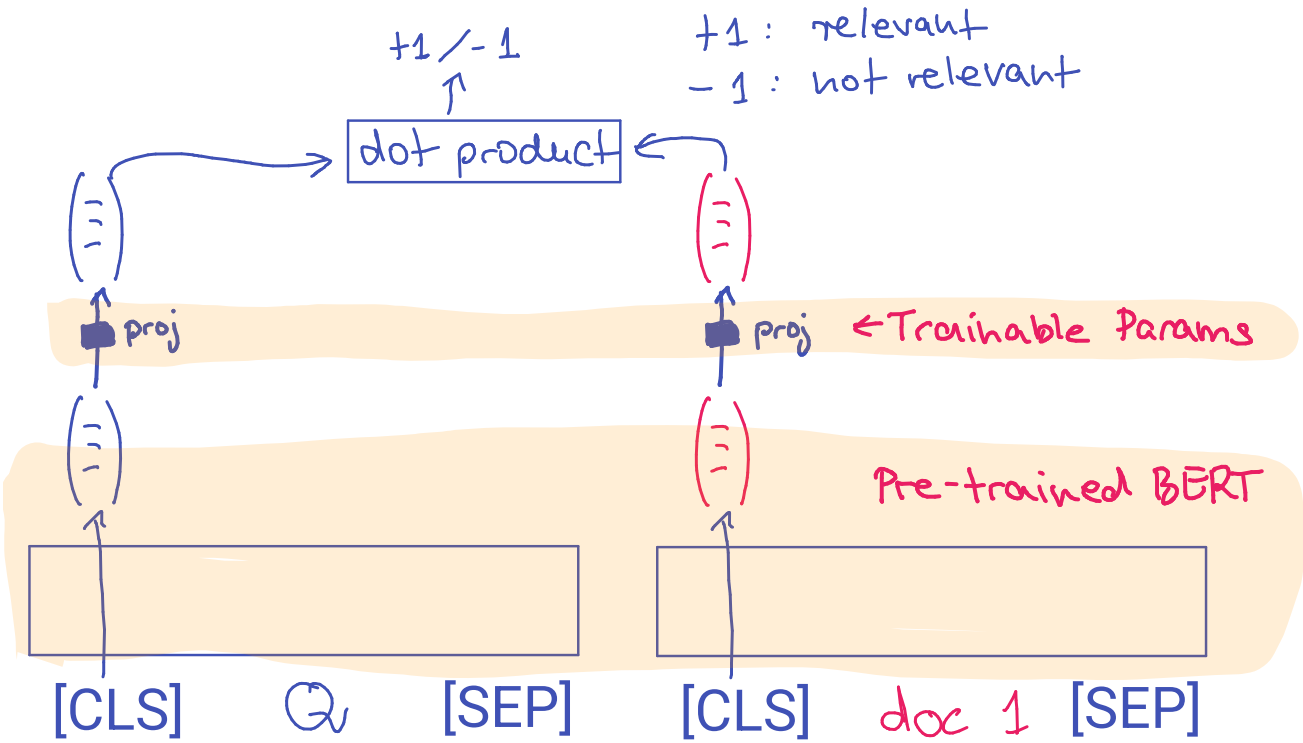


Learn Projection function so that proximity $\approx$ relevance.

$\rightarrow$ Can use Standard K-NN for ranking

Ranking as Neural Network (Bi-encoders)

- Projection layer for training
  - $\texttt{proj}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$
  - where $\mathbf{W}$ and $\mathbf{b}$ are trainable parameters.
- (Non-) Linear projection with trainable parameters



- Alternative: Single trainable Transformer layer

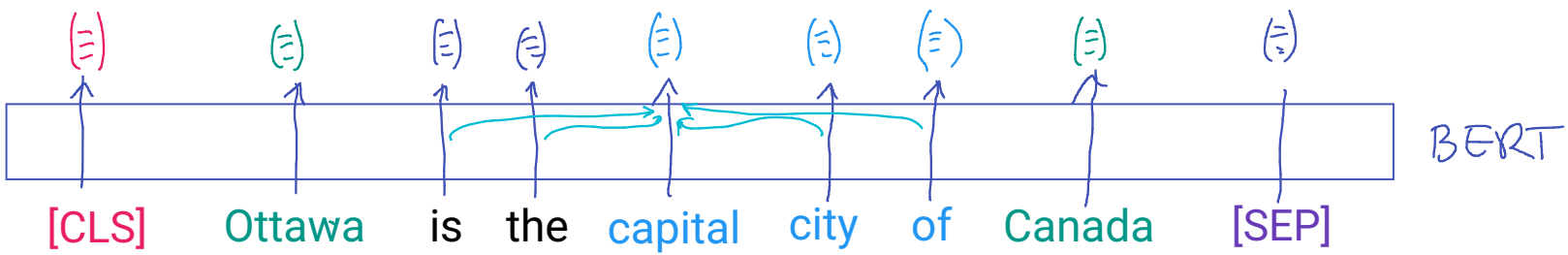
Outline

1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

Text Representation with Large Language Models (LLM)

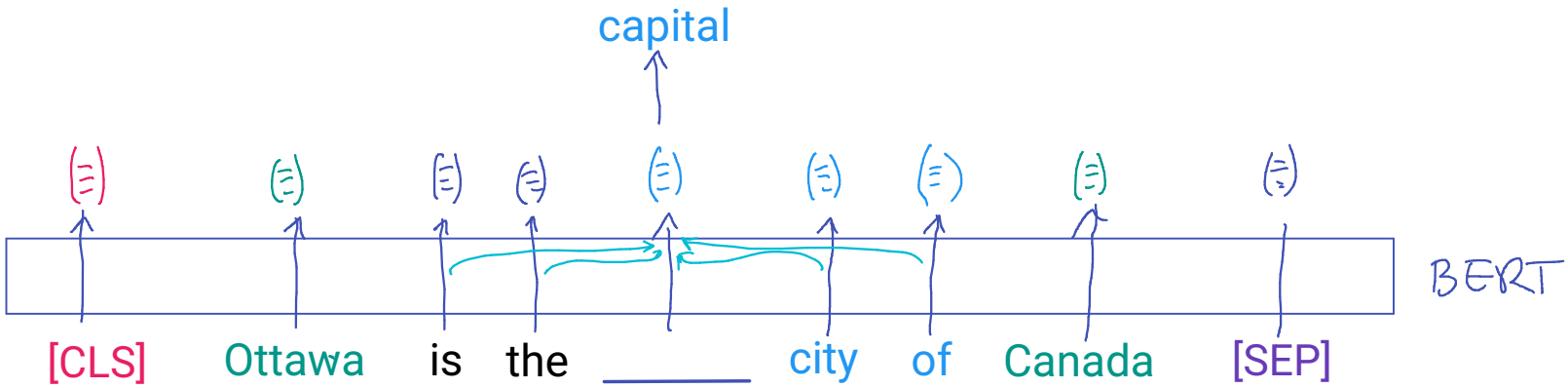
Large Language Models (BERT, T5, GPT, Word2Vec)

- Capture rich contextual information
- Can be fine-tuned for tasks such as entity linking, recognition, or relation extraction
- Represent each word as a vector



Each word vector is aware of neighboring words!
Also: special `[CLS]` and `[SEP]` tokens have vectors

BERT Training with CLOZE "fill the blanks"



Each word vector is aware of neighboring words!

→Contextualized representation

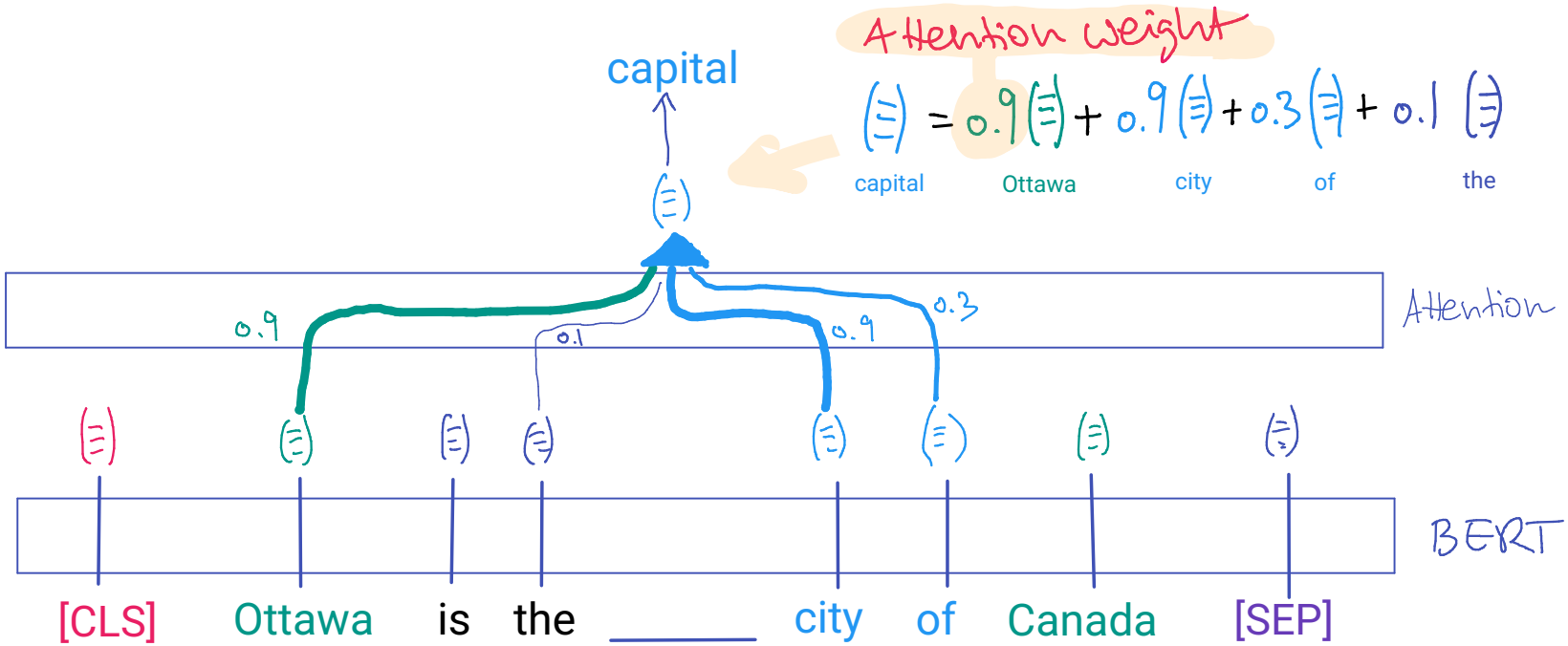
How does this "awareness" work?

Outline

1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

Attention Mechanisms = Adding vectors

- Selectively focuses on relevant parts for each prediction
- Improves the performance especially for long input sequences

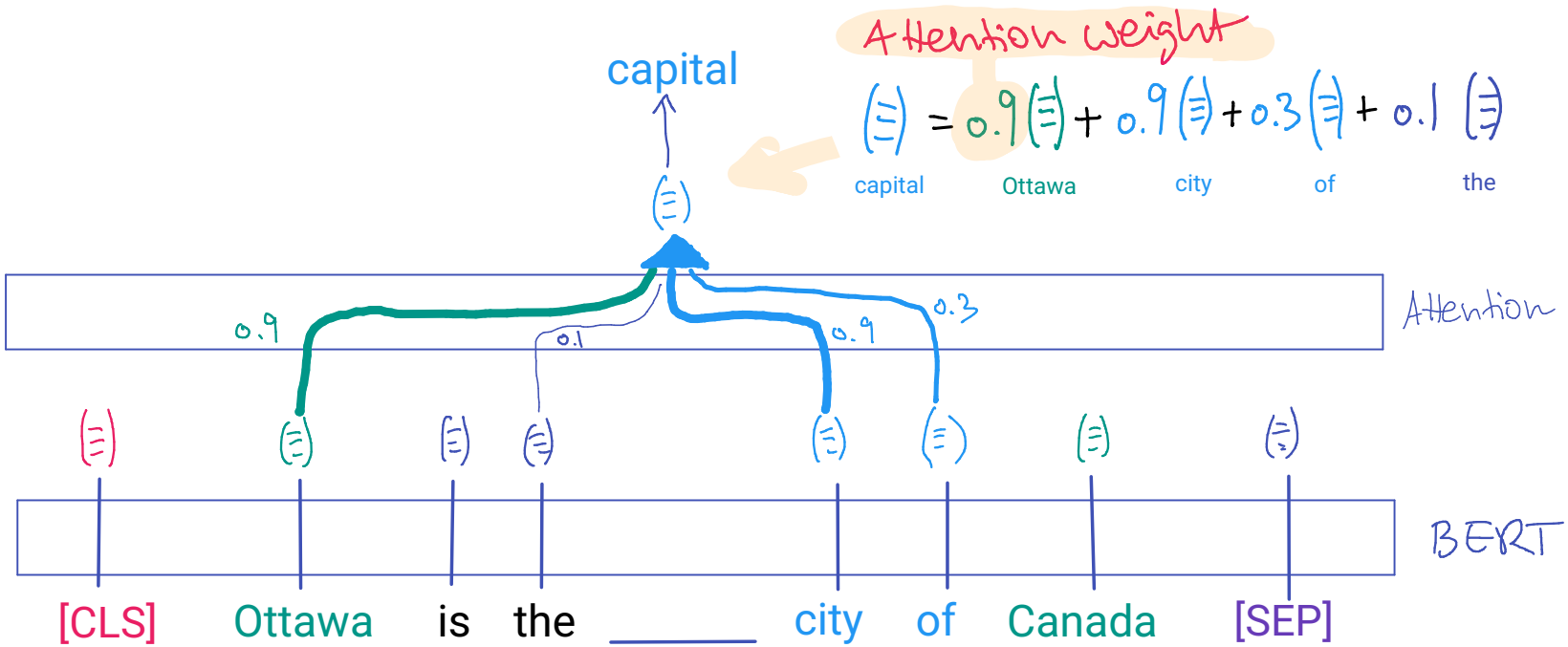


What does "selectively focus" mean?

How Attention is Trained?

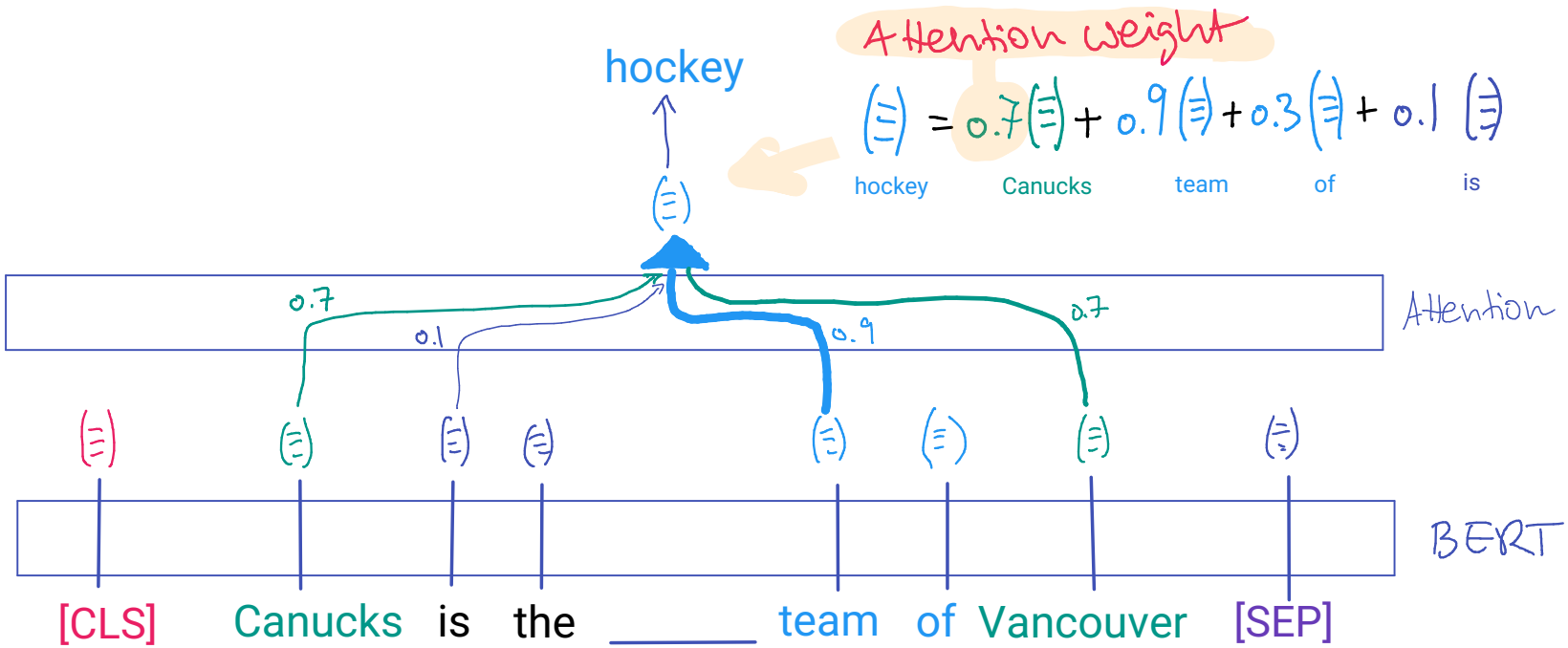
Thought experiment:

- Which tokens should we sum, so we get the best representation?
- Train to assign those tokens a high attention weight!



Generalizing Attention to Unseen Text

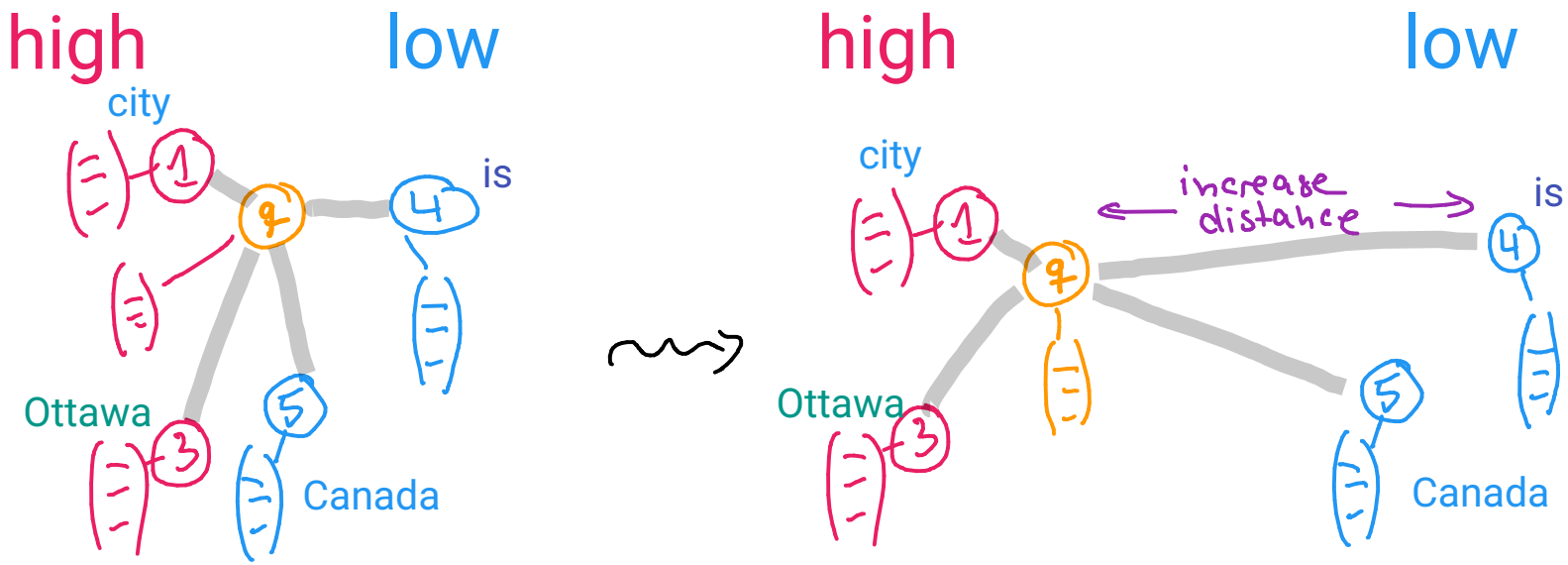
- Need a model to assign these "ideal" attention weights
- .... needs to generalize to unseen text ($\rightarrow$ Delexicalized Space)



Which words should be weighted high in sum?

Attention as Metric Learning on Words

- Project each point $\mathbf{x}$ with a two functions $\texttt{proj}$.
  - Query: $\texttt{proj}(\mathbf{q}) = \mathbf{W}^q\mathbf{q} + \mathbf{b}^q$
  - Key: $\texttt{proj}(\mathbf{k}) = \mathbf{W}^k\mathbf{k} + \mathbf{b}^k$



$\mathbf{q}, \mathbf{k}$ $\qquad$ $\texttt{proj}(\mathbf{q}), \texttt{proj}(\mathbf{k})$

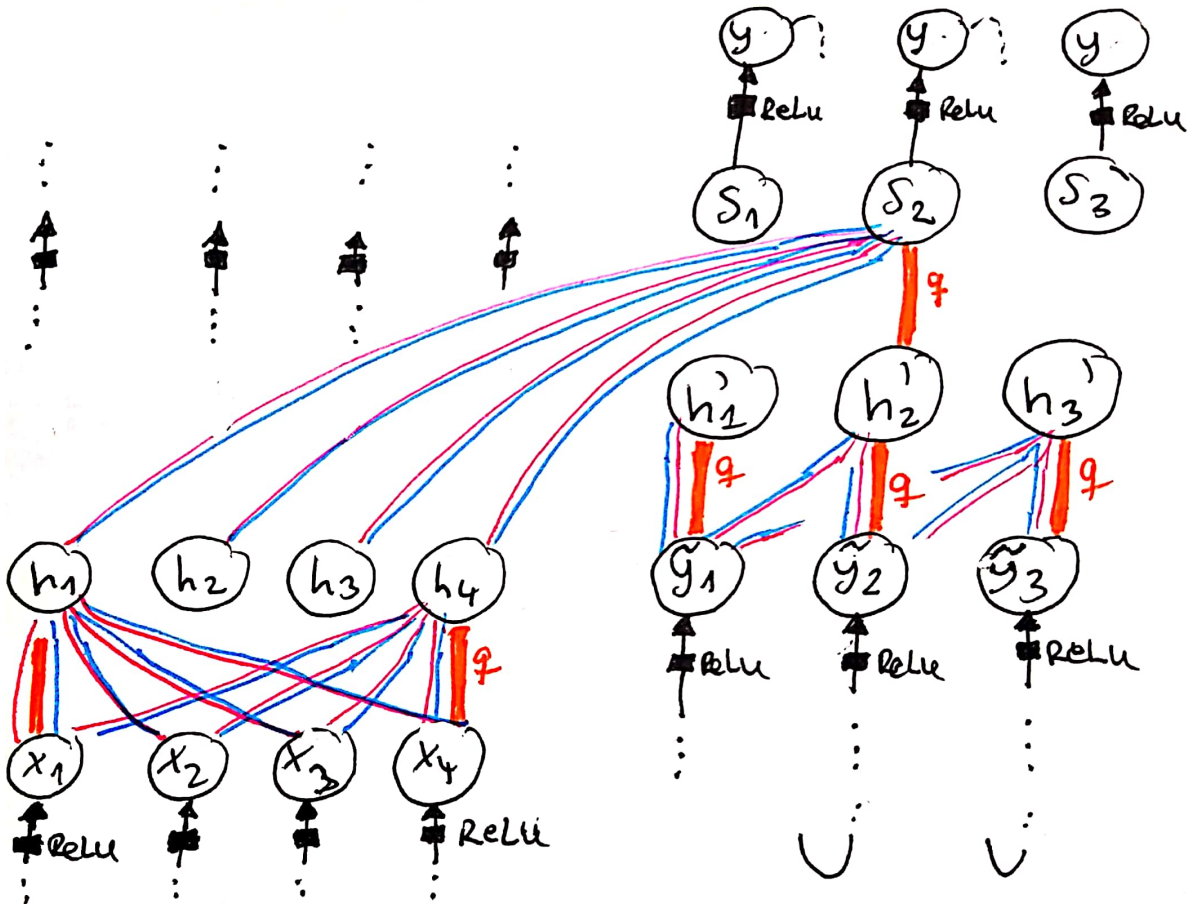
The blank word plays the role of "the query"
High impact words $\approx$ proximity $= \texttt{proj}(\mathbf{q}) \cdot \texttt{proj}(\mathbf{k})$

Outline

1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

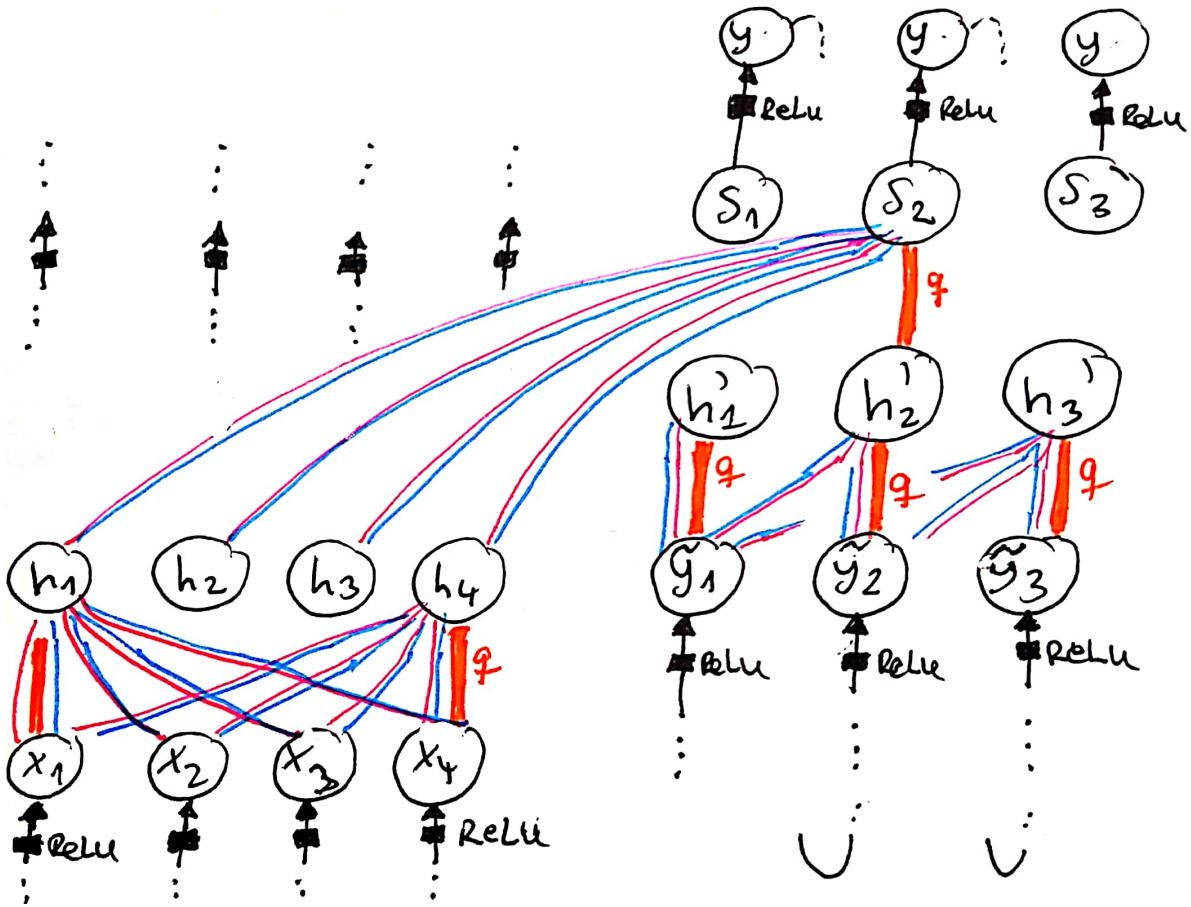
Transformers

- Several layers with attention mechanism = Capture context
- Fine-tuned for specific tasks, such as generation or classification
- Encoder → State → Decoder



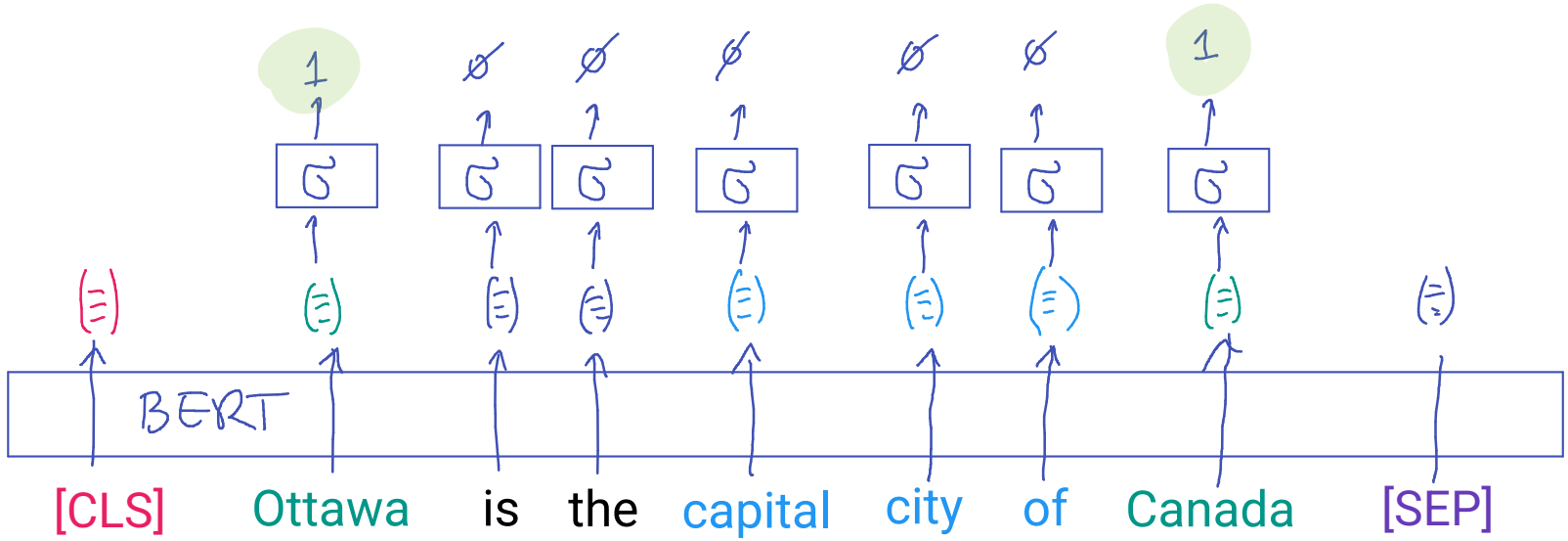
Transformers

- Several layers with attention mechanism = Capture context
- Fine-tuned for specific tasks, such as generation or classification
- Encoder → State → Decoder



Tagging

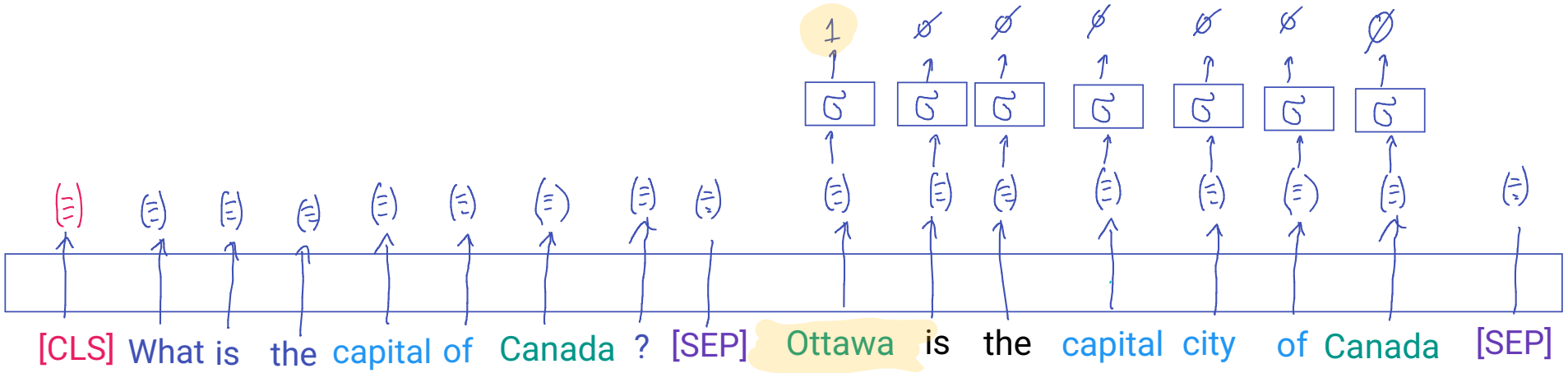
Example: Tag all LOCATION entities in the sentence (green)



- Connect ground truth to vector of words
- Connect to a logistic layer to predict yes/no

Question Answering: Answer Extraction

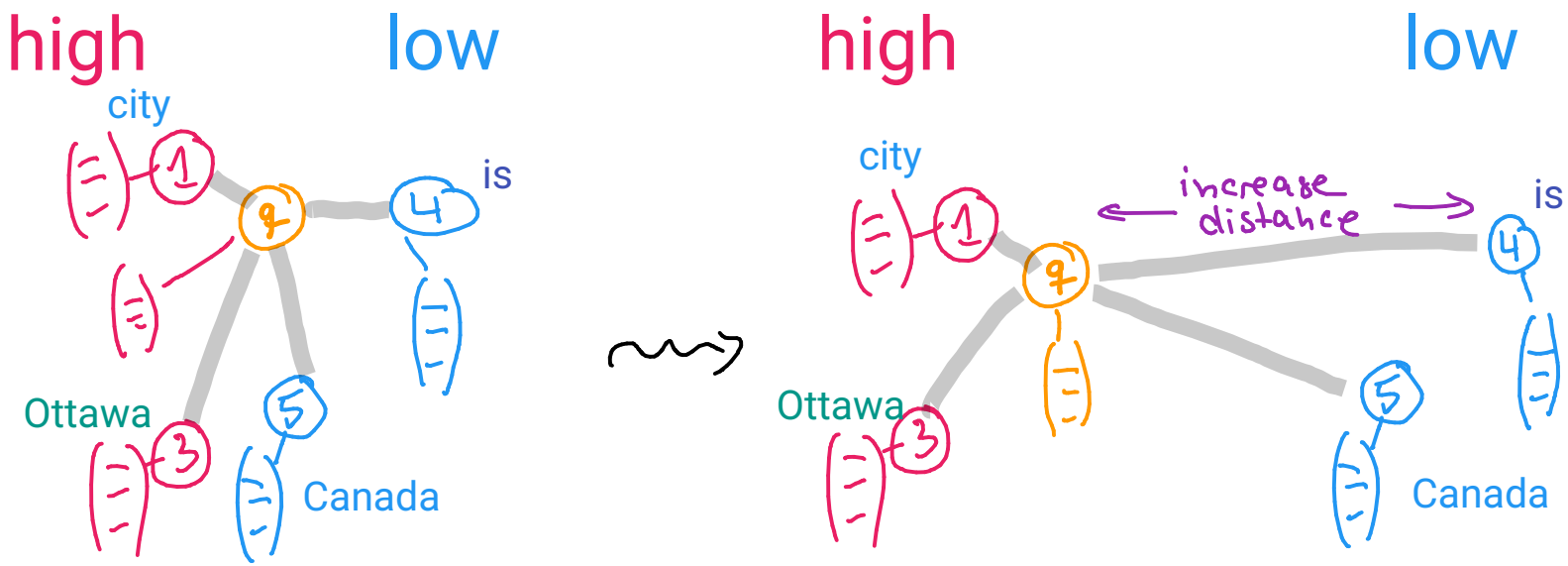
Example: What is the capital of Canada?
Find the answer in the given passage.



- Connect ground truth to vector of words
- Connect to a logistic layer to predict yes/no

Attention and "long-range dependencies"

- "Attention can model long range dependencies" – How?
- Because: Attention does not consider the sequence
- Add **position encoding** to teach about "neighboring words"



Outline

1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

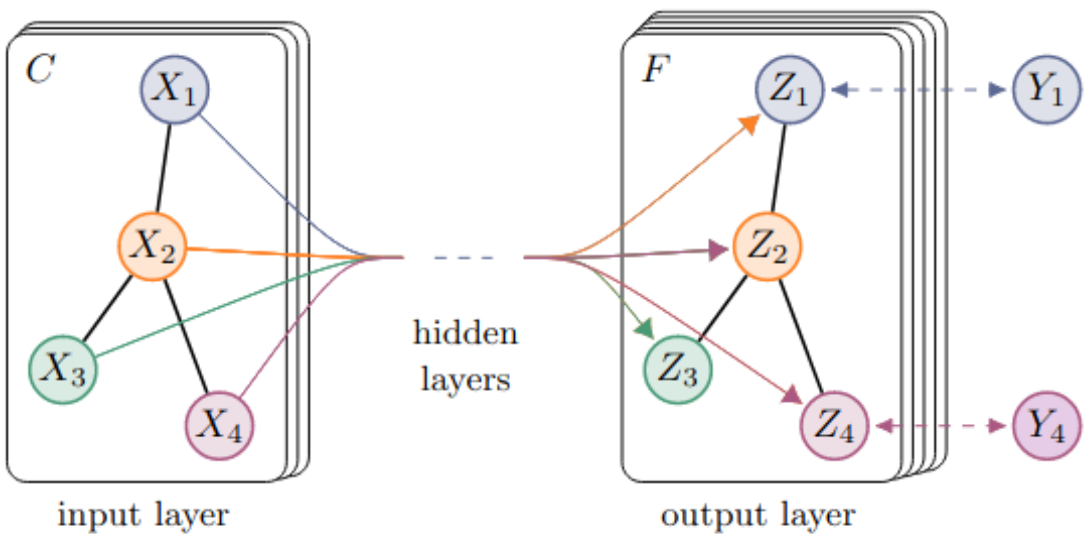
Graph Attention Networks

Given a graph, where

- nodes: entities
- edges: relations

Task:
Which nodes are researchers?

Represent nodes by adding vectors of neighbors

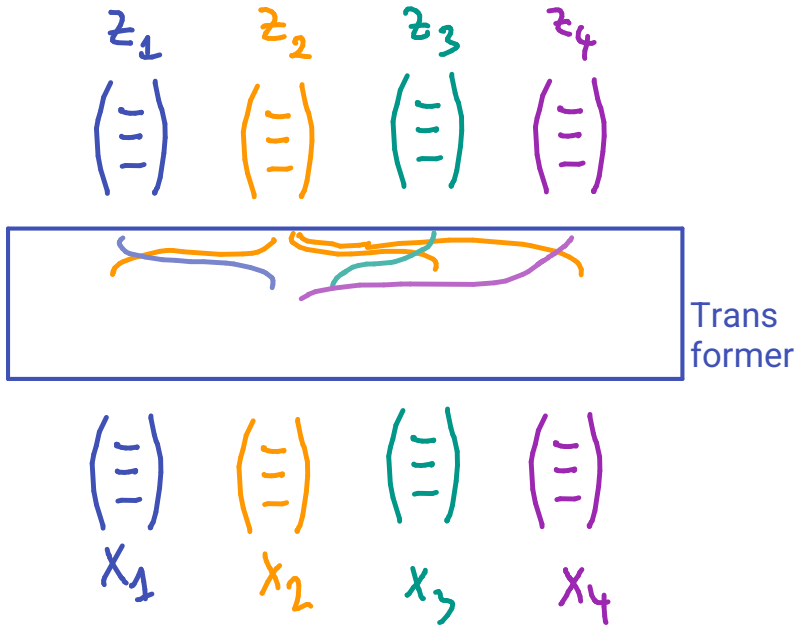


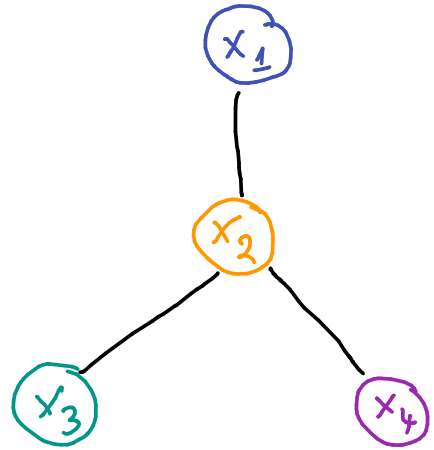
(a) Graph Convolutional Network

(b) Hidden layer activations

Kipf 2017 "Semi-supervised Classification with Graph Concolutional Neural Networks"

Graph Attention = Addition of Neighbors





Attention $\boldsymbol{a}_{qk}$ is restricted to edges (k, q)
Weight in sum is zero when there is no edge

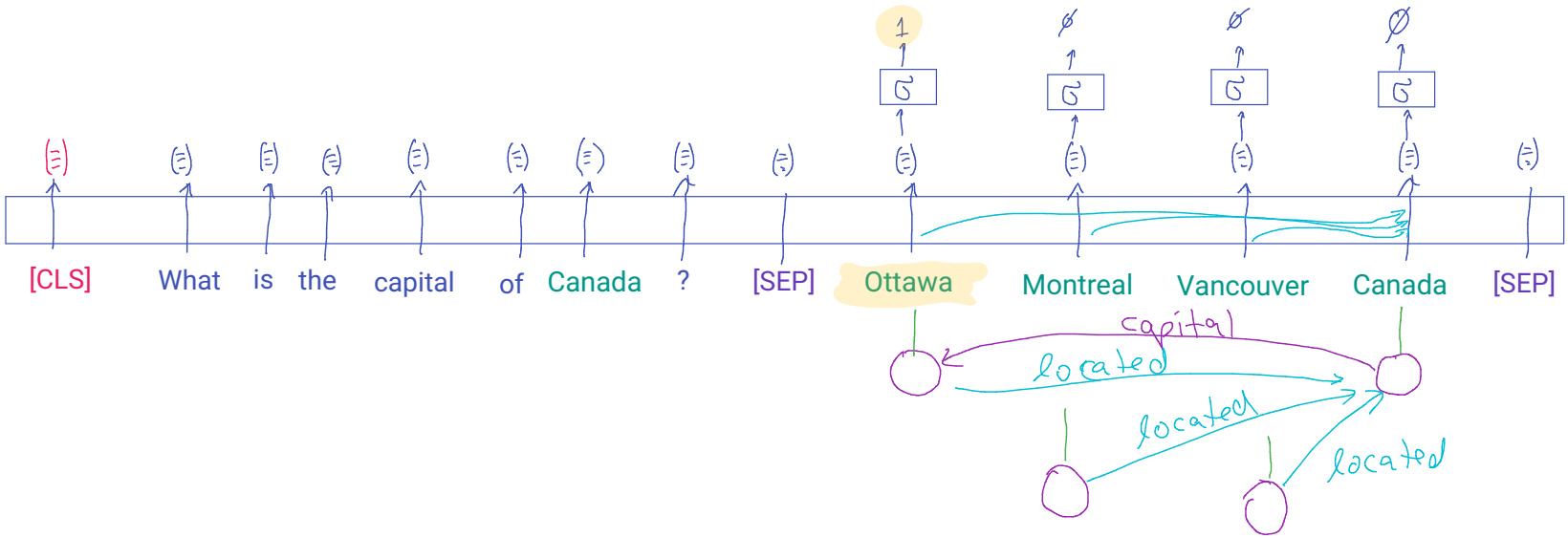
Metric learning on neighbors (stretch edges with low attention)

Outline

1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

Question Answering with Knowledge Graphs

Example: What is the capital of Canada?
Find the node in the knowledge graph that is the answer.



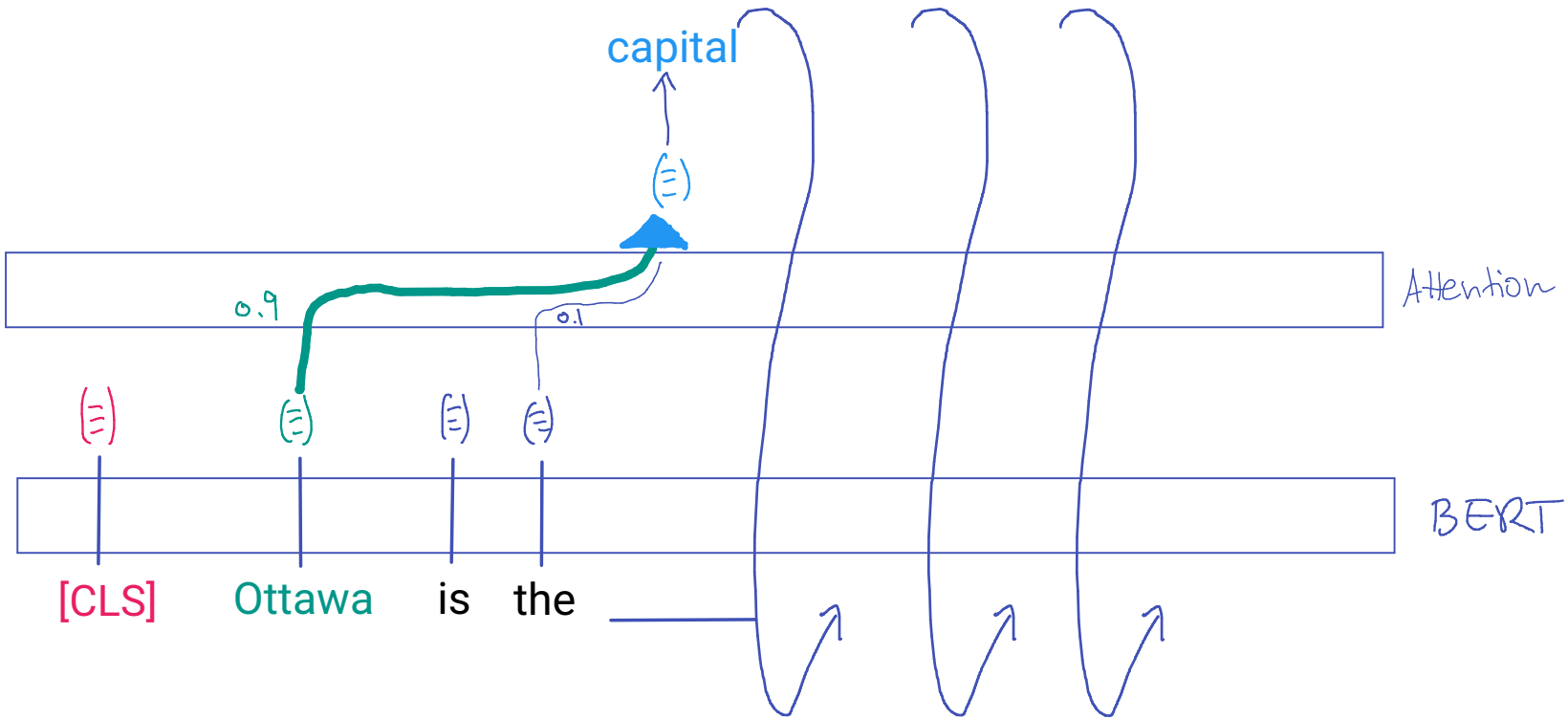
- Combine sequence and graph data via attention
- Graph Attention Networks to select a node = answer.
- Just works because: Attention does not consider the sequence

Outline

1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

GPT

Input: Sequence of Text
Output: Next Word



Main differences to BERT Transformers:

- Can't access the "future"
- Needs to commit to a word before proceeding

Outline

1. Metric Learning
2. Text Ranking
3. Document Representation with Transformers
4. Attention Learning in Transformers
5. Transformers
6. Graph Neural Networks
7. Transformers + Graph Neural Nets
8. Auto-regressive LLMs
9. Neuro-Symbolic

Representing "Hard" Symbols

In Neuro-symbolic methods we think of

- words as soft
- symbols as hard

Transformers can directly handle both words and symbols.
(because attention has no inate concept of position)

Difference:
Symbols come with additional information (KG, DB, semantics)
...we only need to leverage it!