Coarse-to-Fine Memory Matching for Joint Retrieval and Classification

Allen Schmaltz

Harvard University

March 2, 2021

Version 2.0

Today's Reading Group

- Goal is to provide a high-level overview of the neural matching line of work, as it is generally applicable to AI/ machine learning in medicine.
- Presented in the context of a new paper: "`Coarse-to-Fine Memory Matching for Joint Retrieval and Classification`"
  - Not directly a medical task, but easy to understand setting regardless of expertise/focus/area
  - Many real-world tasks fit this setup

Plan

1. Highlight challenges of deep learning in medicine with standard approaches

2. High-level overview of alternative: Exemplar auditing

Motivates subsequent modeling decisions

3. Examine an approach for a retrieval-classification task: fact verification

4. As a result, we get an updatable sequence/language model via 2 mechanisms:

    1. The datastore of retrieved information can be updated

    2. The more abstract model behavior can be updated via a dense database

Plan

1. Highlight challenges of deep learning in medicine with standard approaches

2. High-level overview of alternative: Exemplar auditing

3. Examine an approach for a retrieval-classification task: fact verification

4. As a result, we get an updatable sequence/language model via 2 mechanisms:

    1. The datastore of retrieved information can be updated

    2. The more abstract model behavior can be updated via a dense database

Plan

1. Highlight challenges of deep learning in medicine with standard approaches

2. High-level overview of alternative: Exemplar auditing

3. Examine an approach for a retrieval-classification task: fact verification

4. As a result, we get an updatable sequence/language model via 2 mechanisms:

   1. The datastore of retrieved information can be updated

   2. The more abstract model behavior can be updated via a dense database

New model

Plan

New analysis method

1. Highlight challenges of deep learning in medicine with standard approaches

2. High-level overview of alternative: Exemplar auditing

3. Examine an approach for a retrieval-classification task: fact verification

4. As a result, we get an updatable sequence/language model via 2 mechanisms:

New model

1. The datastore of retrieved information can be updated

2. The more abstract model behavior can be updated via a dense database

Challenges of AI/ML in Medicine

- Difficult to understand models
    - Parameters are not identifiable
- Data issues
    - Often reliable subsets not sufficient for training neural models
    - Annotation error costs are significant for high-risk areas
- Opaque models + data issues == Volatile mix in high-risk settings

Argument

- Instead, we should aim to structure and train networks with an inductive bias that is conducive to comparing distances across representations

- Move to abstain+adapt/update paradigm

Interesting connections with very old ideas: instance-based learning/k-NN/ kernel machines/etc., with deep networks learning the features/kernels…

We seek consistent/meaningful distances between representations for analyzing models AND data. Uncertainty is but a distance to what is known…

New Approach:
Exemplar Auditing

Old

~~New~~ Approach:

Exemplar Auditing

We could alternatively call this “k-nearest neighbor” auditing, or perhaps, “prototype” auditing, but “exemplar” is a throwback to 1980’s “exemplar systems”. There are also connections with instance-based learning methods, more generally.

New Approach:

Exemplar Auditing

Deep networks form the substrate that enables revisiting these old ideas to new effect

We could alternatively call this “k-nearest neighbor” auditing, or perhaps, “prototype” auditing, but “exemplar” is a throwback to 1980’s “exemplar systems”. There are also connections with instance-based learning methods, more generally.

Preliminaries: Deep Networks as *hidden* Metric Learners

Preliminaries: Deep Networks as *hidden* Metric Learners

- N training instances: $x_1, \ldots, x_n, \ldots, x_N$; with ground truth training labels: $y_1, \ldots, y_n, \ldots, y_N$

- Seek a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, to predict $\hat{y}_{N+1}$ for a new, unseen instance x_{N+1}, with minimal *distance* between $\hat{y}_{N+1}$ and y_{N+1}

Preliminaries: Deep Networks as *hidden* Metric Learners

- N training instances: $x_1, \ldots, x_n, \ldots, x_N$; with ground truth training labels: $y_1, \ldots, y_n, \ldots, y_N$

- Seek a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, to predict $\hat{y}_{N+1}$ for a new, unseen instance x_{N+1}, with minimal *distance* between $\hat{y}_{N+1}$ and y_{N+1}

- New view: Back-out a metric learner from the parametric deep network: $f = c \circ g$, where $g : \mathbb{X} \rightarrow \mathbb{R}^M$, $c : \mathbb{R}^M \rightarrow \mathbb{Y}$, and $r \in \mathbb{R}^M$ is a dense representation of the input under the parametric model

Preliminaries: Deep Networks as *hidden* Metric Learners

- N training instances: $x_1, \ldots, x_n, \ldots, x_N$; with ground truth training labels: $y_1, \ldots, y_n, \ldots, y_N$

- Seek a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, to predict $\hat{y}_{N+1}$ for a new, unseen instance x_{N+1}, with minimal *distance* between $\hat{y}_{N+1}$ and y_{N+1}

- New view: Back-out a metric learner from the parametric deep network: $f = c \circ g$, where $g : \mathbb{X} \rightarrow \mathbb{R}^M$, $c : \mathbb{R}^M \rightarrow \mathbb{Y}$, and $r \in \mathbb{R}^M$ is a dense representation of the input under the parametric model

- Sense in which: $f\left(x_{N+1}\right) \approx \sum_{n=1}^{N} y_n \cdot \alpha_n \cdot ||r_n - r_{N+1}||_2$

I.e., a test prediction is approx. a distance-weighting (between "***exemplar***" representations) over the training set

Preliminaries: Deep Networks as *hidden* Metric Learners

- N training instances: $x_1, \ldots, x_n, \ldots, x_N$; with ground truth training labels: $y_1, \ldots, y_n, \ldots, y_N$

- Seek a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, to predict $\hat{y}_{N+1}$ for a new, unseen instance x_{N+1}, with minimal *distance* between $\hat{y}_{N+1}$ and y_{N+1}

- New view: Back-out a metric learner from the parametric deep network: $f = c \circ g$, where $g : \mathbb{X} \rightarrow \mathbb{R}^M$, $c : \mathbb{R}^M \rightarrow \mathbb{Y}$, and $r \in \mathbb{R}^M$ is a dense representation of the input under the parametric model

- Sense in which: $f\left(x_{N+1}\right) \approx \sum_{n=1}^{N} y_n \cdot \alpha_n \cdot ||r_n - r_{N+1}||_2$

  I.e., a test prediction is approx. a distance-weighting (between "***exemplar***" representations) over the training set

- Enables interpretable/introspectable decision rules & various analyses (hence, "***auditing***"): E.g., only admit true positive (TP) matches:

$$\hat{y}_{N+1} = f(x_{N+1}) \cdot \left[f(x_{N+1}) = f(x_n) \wedge f(x_n) = y_n\right] + NULL \cdot \left[f(x_{N+1}) \neq f(x_n) \vee f(x_n) \neq y_n\right], \text{where } n = \underset{n \in \{1,\ldots,N\}}{\arg\min} ||r_n - r_{N+1}||_2$$

Preliminaries: Deep Networks as *hidden* Metric Learners

- N training instances: $x_1, \ldots, x_n, \ldots, x_N$; with ground truth training labels: $y_1, \ldots, y_n, \ldots, y_N$

- Seek a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, to predict $\hat{y}_{N+1}$ for a new, unseen instance x_{N+1}, with minimal *distance* between $\hat{y}_{N+1}$ and y_{N+1}

- New view: Back-out a metric learner from the parametric deep network: $f = c \circ g$, where $g : \mathbb{X} \rightarrow \mathbb{R}^M$, $c : \mathbb{R}^M \rightarrow \mathbb{Y}$, and $r \in \mathbb{R}^M$ is a dense representation of the input under the parametric model

- Sense in which: $f\left(x_{N+1}\right) \approx \sum_{n=1}^{N} y_n \cdot \alpha_n \cdot ||r_n - r_{N+1}||_2$

I.e., a test prediction is approx. a distance-weighting (between "***exemplar***" representations) over the training set

- Enables interpretable/introspectable decision rules & various analyses (hence, "***auditing***"): E.g., only admit true positive (TP) matches:

$$\hat{y}_{N+1} = f(x_{N+1}) \cdot \left[f(x_{N+1}) = f(x_n) \wedge f(x_n) = y_n\right] + NULL \cdot \left[f(x_{N+1}) \neq f(x_n) \vee f(x_n) \neq y_n\right], \text{where } n = \underset{n \in \{1,\ldots,N\}}{\arg\min} ||r_n - r_{N+1}||_2$$

We use such training set matching as a post-processing mask (with decision rules) over $f\left(x_{N+i}\right)$, but in principle, with the model presented today, we could directly train against this (via the similarity loss)

Preliminaries: Deep Networks as *hidden* Metric Learners (Cont.)

- Seek a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, to predict $\hat{y}_{N+1}$ for a new, unseen instance x_{N+1}, with minimal *distance* between $\hat{y}_{N+1}$ and y_{N+1}

- New view: Back-out a metric learner from the parametric deep network:

  $f = c \circ g$, where $g : \mathbb{X} \rightarrow \mathbb{R}^M$, $c : \mathbb{R}^M \rightarrow \mathbb{Y}$, and $r \in \mathbb{R}^M$ is a dense representation of the input under the parametric model

- Enables interpretable/introspectable decision rules & various analyses (hence, "***auditing***"): E.g., only admit true positive (TP) matches:

  $\hat{y}_{N+1} = f(x_{N+1}) \cdot \left[f(x_{N+1}) = f(x_n) \wedge f(x_n) = y_n \right] + NULL \cdot \left[f(x_{N+1}) \neq f(x_n) \vee f(x_n) \neq y_n \right]$, where $n = \underset{n \in \{1,\ldots,N\}}{\arg\min} \; || r_n - r_{N+1} ||_2$

Preliminaries: Deep Networks as *hidden* Metric Learners (Cont.)

- Seek a function, $f : \mathbb{X} \rightarrow \mathbb{Y}$, to predict $\hat{y}_{N+1}$ for a new, unseen instance x_{N+1}, with minimal *distance* between $\hat{y}_{N+1}$ and y_{N+1}

- New view: Back-out a metric learner from the parametric deep network:

  $f = c \circ g$, where $g : \mathbb{X} \rightarrow \mathbb{R}^M$, $c : \mathbb{R}^M \rightarrow \mathbb{Y}$, and $r \in \mathbb{R}^M$ is a dense representation of the input under the parametric model

- Enables interpretable/introspectable decision rules & various analyses (hence, "***auditing***"): E.g., only admit true positive (TP) matches:

  $\hat{y}_{N+1} = f(x_{N+1}) \cdot \left[f(x_{N+1}) = f(x_n) \wedge f(x_n) = y_n\right] + NULL \cdot \left[f(x_{N+1}) \neq f(x_n) \vee f(x_n) \neq y_n\right]$, where $n = \underset{n \in \{1,\ldots,N\}}{\arg\min} \; ||r_n - r_{N+1}||_2$

- Also, enables updatability/adaptability:

  - Label changes: $y'_n = y_n + \Delta_n$

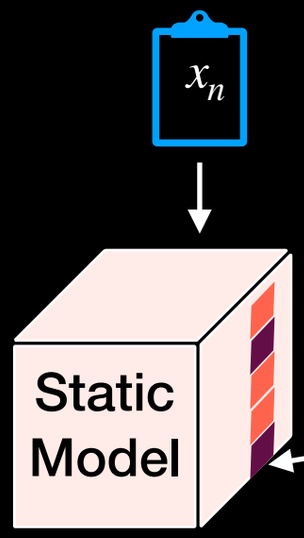
  - Data additions (a.k.a., continual/lifelong learning): $\mathbb{D}^N = \left\{(x_1, y_1), \ldots, (x_N, y_N)\right\}$ becomes $\mathbb{D}^{N'} = \left\{(x_1, y_1), \ldots, (x_N, y_N), \ldots, (x_{N'}, y_{N'})\right\}$

  - New lightweight models over representations (e.g., using data additions): $c' : \mathbb{R}^M \rightarrow \mathbb{Y}'$

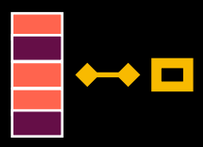
New Approach: Exemplar Auditing

New Approach: Exemplar Auditing

1. Train the model such that ‘exemplar’ vectors summarize the network



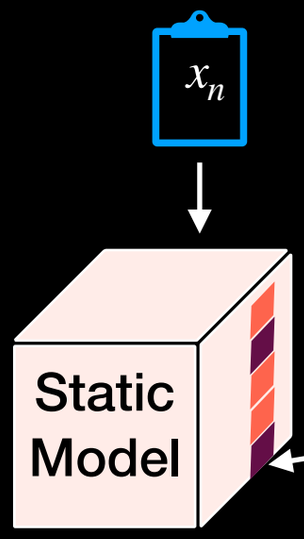
Exemplar vector (dense representation/‘distilled memory’) of the full parametric network



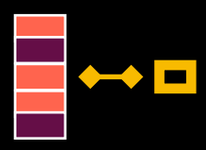
Training exemplar vectors are associated with ground-truth labels

New Approach: Exemplar Auditing

1. Train the model such that ‘exemplar’ vectors summarize the network

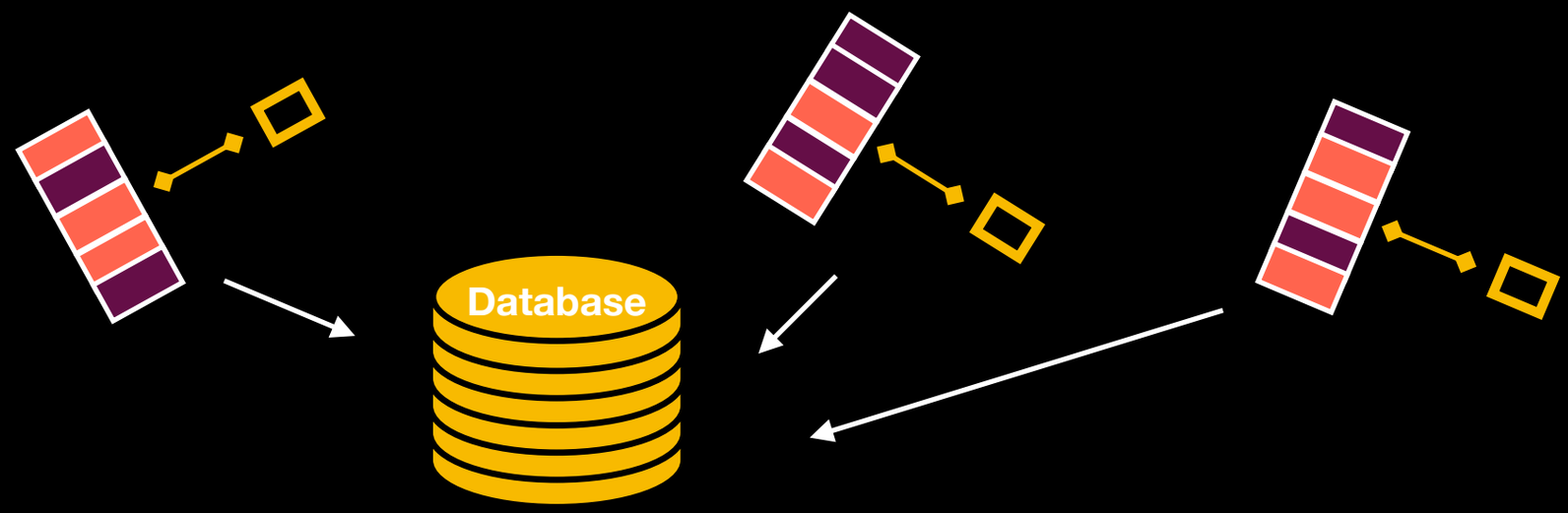


Exemplar vector (dense representation/‘distilled memory’) of the full parametric network



Training exemplar vectors are associated with ground-truth labels

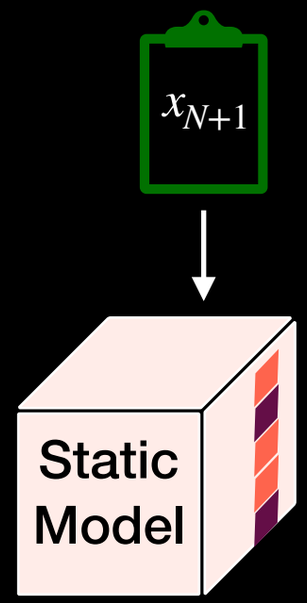
2. For all training instances, add exemplars & labels to a database



New Approach: Exemplar Auditing

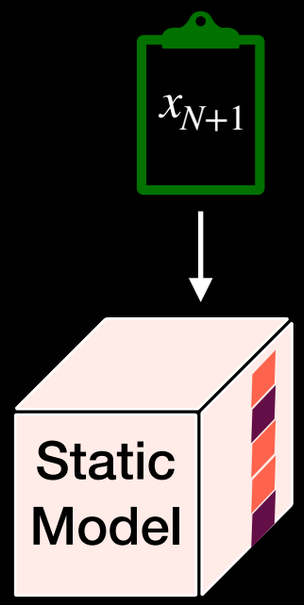
New Approach: Exemplar Auditing

3. At test, create exemplar vector

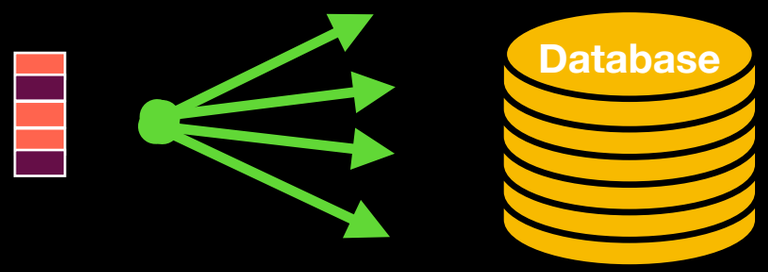


New Approach: Exemplar Auditing

3. At test, create exemplar vector

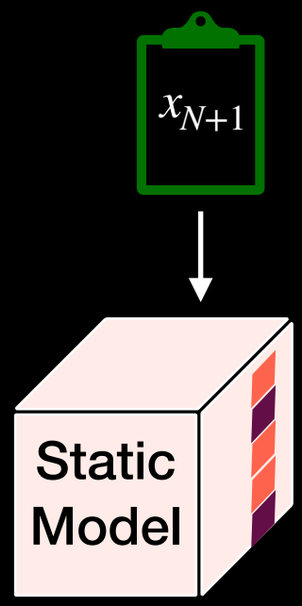


4. Match test exemplar vector to database

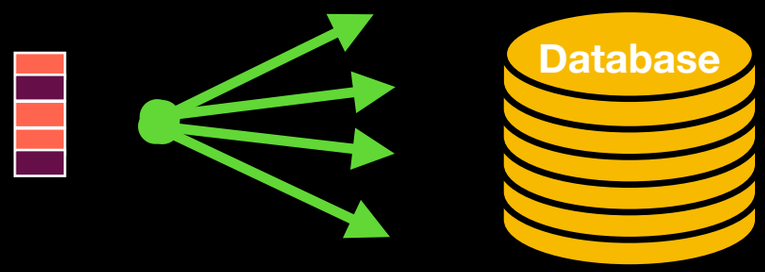


New Approach: Exemplar Auditing

3. At test, create exemplar vector



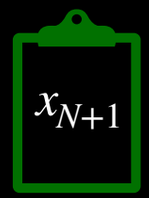
4. Match test exemplar vector to database

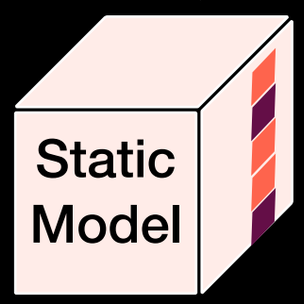


5. Constrain prediction based on nearest label (or distribution of labels) in database
-Can also leverage the distances to the exemplars

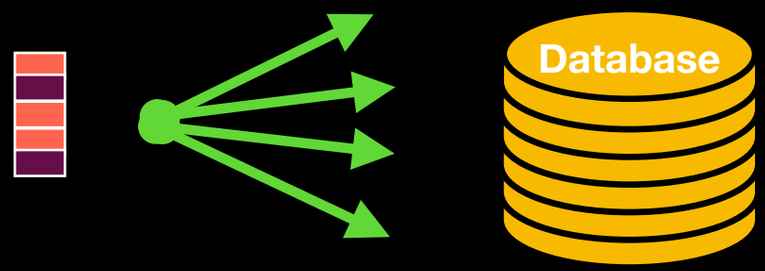
New Approach: Exemplar Auditing

3. At test, create exemplar vector





4. Match test exemplar vector to database



Orthogonal to empirical bounds (conformal, etc.) via held-out set

5. Constrain prediction based on nearest label (or distribution of labels) in database
-Can also leverage the distances to the exemplars

New Approach: Exemplar Auditing

New Approach: Exemplar Auditing

6. We can update the database over time by adding new instances (e.g., out-of-domain), modifying or adding labels, etc.

New Approach: Exemplar Auditing

6. We can update the database over time by adding new instances (e.g., out-of-domain), modifying or adding labels, etc.

7. Can use to analyze the data (annotation errors, etc.)

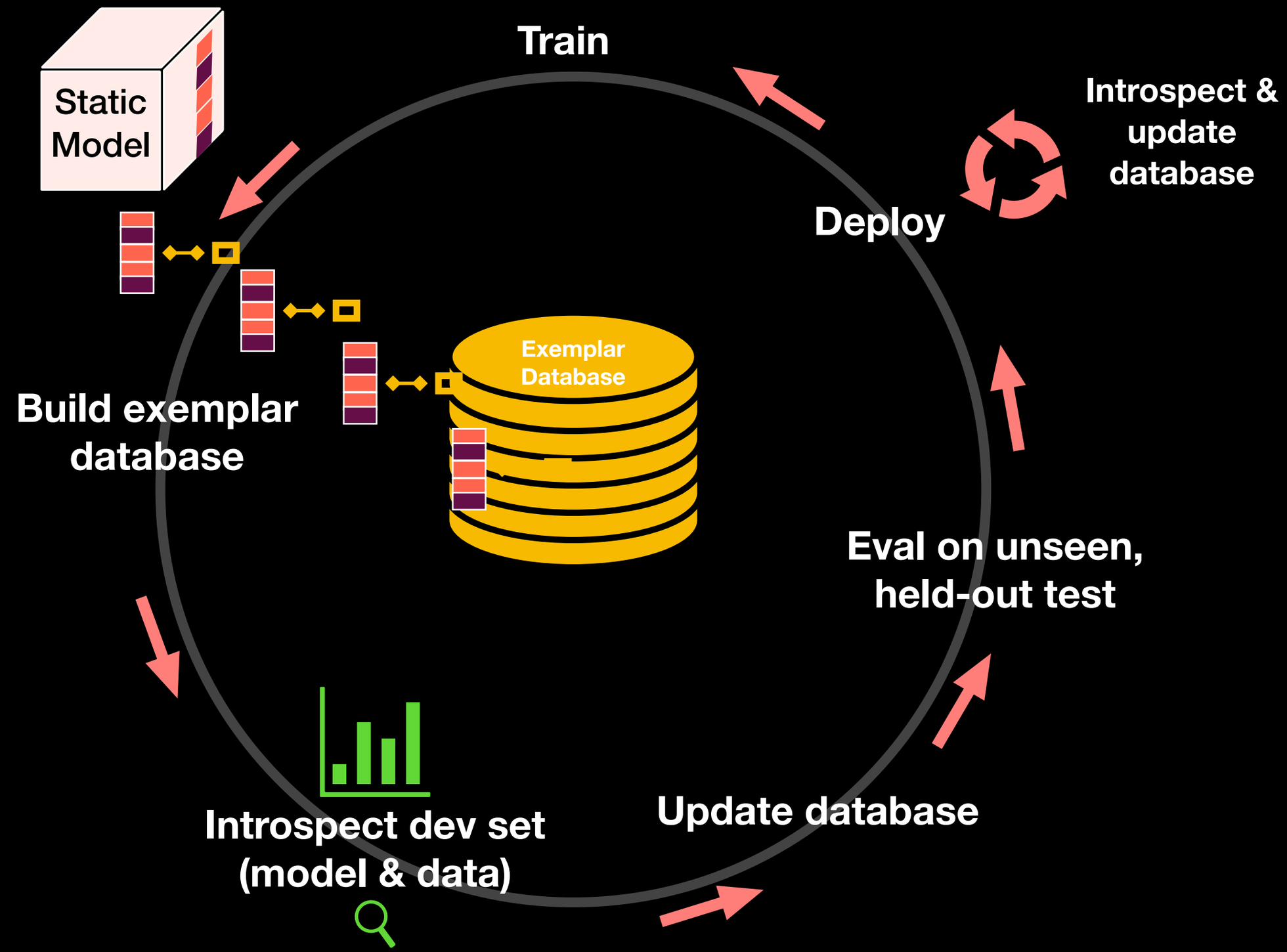
New Approach: Exemplar Auditing

6. We can update the database over time by adding new instances (e.g., out-of-domain), modifying or adding labels, etc.

7. Can use to analyze the data (annotation errors, etc.)

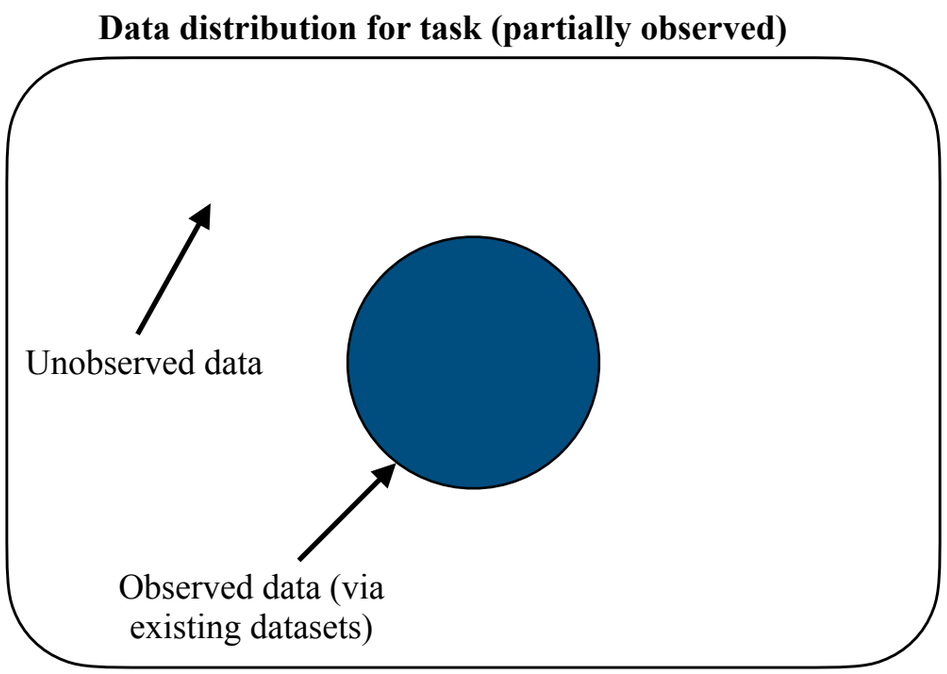
8. As necessary, re-train the model with new/updated labels, instances, etc.

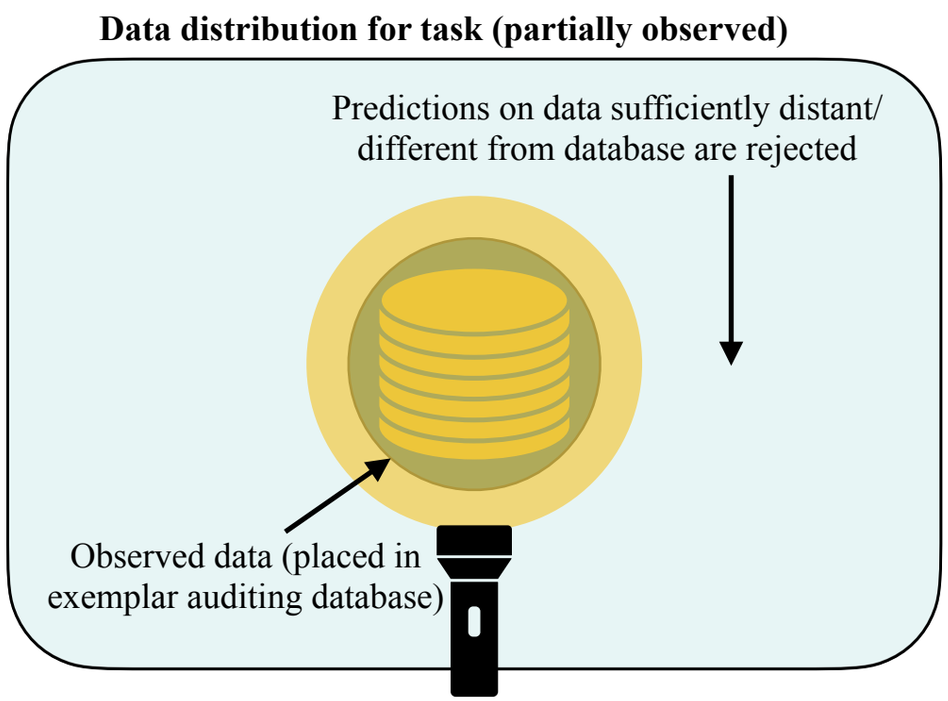
Exemplar Auditing Lifecycle



Motivation: Out-of-domain

- Pre-train with as much data as possible
- Add as much data as possible to the database, including data not seen in training
    - Corral the in-domain space, around the ball of the observed data
- Never predict over out-of-domain in high-risk settings—Instead: Rearrange deployment to handle non-admitted predictions





What makes this work (i.e., why haven’t we always done this)?

What makes this work (i.e., why haven't we always done this)?

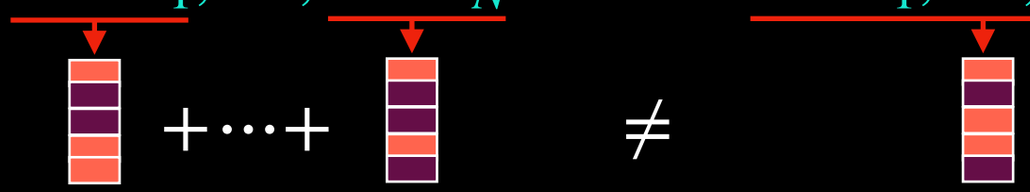
- Need a sufficiently expressive, strong parametric network over the input
  - Sense in which: $\textbf{word}_1, \ldots, \textbf{word}_N$ $+\cdots+$ $\neq$ $\textbf{word}_1, \ldots, \textbf{word}_N$

I.e., ill-posed matching

- Seek coupled, end-to-end models of input dependencies, including for tasks where the input consists of multiple sequences

What makes this work (i.e., why haven't we always done this)?

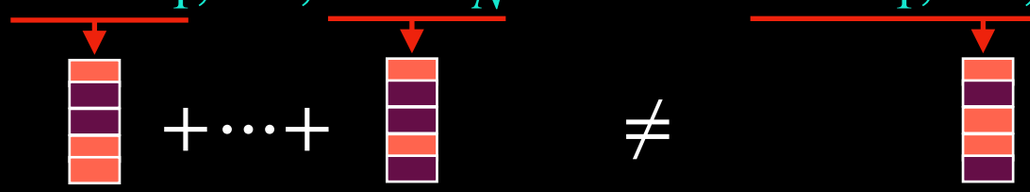
- Need a sufficiently expressive, strong parametric network over the input
  - Sense in which: $\textbf{word}_1, \ldots, \textbf{word}_N$ $+\cdots+$ $\neq$ $\textbf{word}_1, \ldots, \textbf{word}_N$



    - Seek coupled, end-to-end models of input dependencies, including for tasks where the input consists of multiple sequences
- Need effective, compact representations of the parametric network
  - Structure of the network (i.e., the inductive bias) & training approach are critical
  - Representations of the deep network need to be sufficiently compact so that at least approximate search over the database of exemplars is feasible

What makes this work (i.e., why haven't we always done this)?

- Need a sufficiently expressive, strong parametric network over the input
  - Sense in which: $\textbf{word}_1, \ldots, \textbf{word}_N$ $+\cdots+$ $\neq$ $\textbf{word}_1, \ldots, \textbf{word}_N$



  - Seek coupled, end-to-end models of input dependencies, including for tasks where the input consists of multiple sequences
- Need effective, compact representations of the parametric network
  - Structure of the network (i.e., the inductive bias) & training approach are critical
  - Representations of the deep network need to be sufficiently compact so that at least approximate search over the database of exemplars is feasible
- Need additional structures in code and deployments to handle dense search at test/inference and updating the database
  - $\Rightarrow$ More complicated codebases and deployments than the standard approaches

Retrieval-Classification

Retrieval-Classification

- Retrieval-classification tasks: E.g., QA & fact verification
  - Need to retrieve multiple sequences
  - Then make a classification decision over those sequences

Retrieval-Classification

- $f : \mathbb{X} \times \mathscr{D} \to \left\langle \{0,1,2\},\ 2^{|\mathbb{D}|} \right\rangle$

Retrieval-Classification

- $f : \mathbb{X} \times \mathscr{D} \rightarrow \left\langle \{0,1,2\},\ 2^{|\mathbb{D}|} \right\rangle$
- $f : \mathbb{Q} \times \mathscr{D} \rightarrow \left\langle \{\text{Refutes, Supports, Unverifiable}\},\ 2^{|\mathbb{D}|} \right\rangle$

- Given:
  - $q \in \mathbb{Q}$, a ‘query’ (e.g., a sentence)
  - $\mathbb{D} \in \mathscr{D}$, a set of documents (e.g., all of Wikipedia)
- Determine:
  - The query is Supported, Refuted, or Unverifiable AND the subset of $\mathbb{D}$ to support that prediction

Retrieval-Classification

- $f : \mathbb{Q} \times \mathscr{D} \rightarrow \left\langle \{\text{Refutes, Supports, Unverifiable}\},\ 2^{|\mathbb{D}|} \right\rangle$
  - $q \in \mathbb{Q}$, a ‘query’ (e.g., a sentence)
  - $\mathbb{D} \in \mathscr{D}$, a set of documents (e.g., all of wikipedia)
- Viewable as two separate tasks:
  - $f_1 : \mathbb{Q} \times \mathscr{D} \rightarrow 2^{|\mathbb{D}|}$ — Information retrieval
  - $f_2 : \mathbb{Q} \times 2^{|\mathbb{D}|} \rightarrow \{\text{Refutes, Supports, Unverifiable}\}$ — Natural language inference

Retrieval-Classification

- $f : \mathbb{Q} \times \mathscr{D} \rightarrow \left\langle \{\text{Refutes, Supports, Unverifiable}\},\ 2^{|\mathbb{D}|} \right\rangle$
  - $q \in \mathbb{Q}$, a ‘query’ (e.g., a sentence)
  - $\mathbb{D} \in \mathscr{D}$, a set of documents (e.g., all of wikipedia)
- Viewable as two separate tasks:
  - $f_1 : \mathbb{Q} \times \mathscr{D} \rightarrow 2^{|\mathbb{D}|}$ — Information retrieval
  - $f_2 : \mathbb{Q} \times 2^{|\mathbb{D}|} \rightarrow \{\text{Refutes, Supports, Unverifiable}\}$ — Natural language inference
  - Can we learn together with a single model?

Retrieval-Classification Task: FEVER

- Fact Extraction and VERification (FEVER) Shared Task
- Given:
  - A claim (short, declarative sentence)
  - Wikipedia
- Predict:
  - Claim is Supported, Refuted, or Unverifiable
  - ≤ 5 sentences that support that prediction

Example from FEVER

- **INPUT:** Claim: Charles de Gaulle was a leader in the French Resistance.

- **RETRIEVE:** Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.

- **PREDICT:** Supports

Example from FEVER

- **INPUT:** Claim: Charles de Gaulle was a leader in the French Resistance.

- **RETRIEVE:** Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.

- **PREDICT:** Supports

Sentences made unique by article title and sentence index

Example from FEVER

Query sequence

- **INPUT:** Claim: Charles de Gaulle was a leader in the French Resistance.

- **RETRIEVE:** Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.

Support sequence

- **PREDICT:** Supports

Classification label

Example from FEVER

Example from FEVER

- **INPUT:** Claim: Emma Stone was born in Taiwan.

Example from FEVER

- **INPUT:** Claim: Emma Stone was born in Taiwan.
- **RETRIEVE:** Evidence: Emma Stone, sentence 5: Born and raised in Scottsdale, Arizona, Stone began acting as a child, in a theater production of The Wind in the Willows in 2000.

Example from FEVER

- **INPUT:** Claim: Emma Stone was born in Taiwan.
- **RETRIEVE:** Evidence: Emma Stone, sentence 5: Born and raised in Scottsdale, Arizona, Stone began acting as a child, in a theater production of The Wind in the Willows in 2000.

- **PREDICT:** Refutes

Example from FEVER

- **INPUT:** Claim: Emma Stone was born in Taiwan.
- **RETRIEVE:** Evidence: Emma Stone, sentence 5: Born and raised in Scottsdale, Arizona, Stone began acting as a child, in a theater production of The Wind in the Willows in 2000.

- **PREDICT:** Refutes

FEVER: Existing Work

- Most existing works are multi-model pipelines
  - Document retrieval model
  - Sentence selection model
  - Classification model
- Each model is trained and run independently

FEVER: MemMatch

- We instead propose a novel single, end-to-end language model for both retrieval & classification
  - Coarse-to-fine search procedure over dense representations
  - Distances from tightly coupled retrieval and classification can be leveraged to identify low-confidence instances
  - Produces composed dense representations over multiple sequences for exemplar auditing

I.e., a neural model with a long-term, explicit memory unit/cache

FEVER: MemMatch

- Effective:
  - More effective than relying on LM parameters as a knowledge base
  - Approaches multi-pipeline systems despite using significantly fewer parameters

FEVER: MemMatch

- Novel properties: Updatability of language model behavior through two distinct mechanisms:
  - Retrieved information can be updated explicitly
  - Model behavior can be modified via the exemplar database

FEVER: MemMatch Model

FEVER: MemMatch Model

- $f : \mathbb{Q} \times \mathscr{D} \rightarrow \left\langle \{\text{Refutes, Supports, Unverifiable}\}, 2^{|\mathbb{D}|} \right\rangle$
  - $f(q, \mathbb{D})$ $q = \texttt{Query sequence}$
    - $\mathbb{D}$ contains all sentences of Wikipedia, so f, a single deep network, needs a long-term memory unit—& a means of searching through $2^{|\mathbb{D}|}$ possible combinations of `Wikipedia sentences`

FEVER: MemMatch Model

- $f : \mathbb{Q} \times \mathscr{D} \rightarrow \left\langle \{\text{Refutes, Supports, Unverifiable}\},\ 2^{|\mathbb{D}|} \right\rangle$
  - $f(q, \mathbb{D})$ — q = `Query sequence`
    - $\mathbb{D}$ contains all sentences of Wikipedia, so f, a single deep network, needs a long-term memory unit—& a means of searching through $2^{|\mathbb{D}|}$ possible combinations of `Wikipedia sentences`

- Approach (*high-level*): Run the same shared network over all of Wikipedia, cacheing the representations, & then perform search by matching the query representation with progressively built-up support sequences

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

Search Level 1

Search Level 2

Search Level 3

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

s = Support sequence

I.e., 1 Wikipedia sentence

Search Level 1

$$g\left(s_1\right) = r_{s_1} \in \mathbb{R}^M \qquad s_i \in \mathbb{D}$$

$$\vdots$$

$$g\left(s_{|\mathbb{D}|}\right) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$$

Search Level 2

Search Level 3

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

Search Level 1

s = Support sequence

$$g\left(s_1\right) = r_{s_1} \in \mathbb{R}^M$$
$$\vdots$$
$$g\left(s_{|\mathbb{D}|}\right) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$$

I.e., 1 Wikipedia sentence

$s_i \in \mathbb{D}$

Search Level 2

$r_{s_1}, \ldots, r_{s_{|\mathbb{D}|}}$ can be cached, & then reused for every query

Search Level 3

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

s = Support sequence

I.e., 1 Wikipedia sentence

Search Level 1

$$g\left(s_1\right) = r_{s_1} \in \mathbb{R}^M \qquad s_i \in \mathbb{D}$$

$$\vdots$$

$$g\left(s_{|\mathbb{D}|}\right) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$$

Search Level 2

Search Level 3

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

q = Query sequence

s = Support sequence

I.e., 1 Wikipedia sentence

Search Level 1 $g(q) = r_q \in \mathbb{R}^M$ $g(s_1) = r_{s_1} \in \mathbb{R}^M$ $s_i \in \mathbb{D}$

$\vdots$

$g(s_{|\mathbb{D}|}) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$

Search Level 2

Search Level 3

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

q = Query sequence

s = Support sequence

I.e., 1 Wikipedia sentence

Search Level 1 $g(q) = r_q \in \mathbb{R}^M \longleftrightarrow g(s_1) = r_{s_1} \in \mathbb{R}^M$

$\vdots$

$g(s_{|\mathbb{D}|}) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$

$s_i \in \mathbb{D}$

I.e., set of K nearest Wikipedia sentences

$$s'_k \in \underset{s_i}{\arg K \min} \, || \, r_q - r_{s_i} \, ||_2$$

Search Level 2

Search Level 3

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

q = Query sequence

s = Support sequence

I.e., 1 Wikipedia sentence

Search Level 1

$$g(q) = r_q \in \mathbb{R}^M \longleftrightarrow g(s_1) = r_{s_1} \in \mathbb{R}^M$$

$$\vdots$$

$$g(s_{|\mathbb{D}|}) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$$

$s_i \in \mathbb{D}$

I.e., set of K nearest Wikipedia sentences

$$s'_k \in \arg K \min_{s_i} ||r_q - r_{s_i}||_2$$

Search Level 2

$$g((q, s'_1)) = r_{(q,s'_1)} \in \mathbb{R}^M$$

$$\vdots$$

$$g((q, s'_K)) = r_{(q,s'_K)} \in \mathbb{R}^M$$

Search Level 3

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

q = Query sequence

s = Support sequence

I.e., 1 Wikipedia sentence

Search Level 1

$$g(q) = r_q \in \mathbb{R}^M \longleftrightarrow g(s_1) = r_{s_1} \in \mathbb{R}^M$$

$$\vdots$$

$$g(s_{|\mathbb{D}|}) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$$

$$s_i \in \mathbb{D}$$

I.e., set of K nearest Wikipedia sentences

$$s'_k \in \arg K \min_{s_i} ||r_q - r_{s_i}||_2$$

Search Level 2

$$r_q \longleftrightarrow g((q, s'_1)) = r_{(q,s'_1)} \in \mathbb{R}^M$$

$$\vdots$$

$$g((q, s'_K)) = r_{(q,s'_K)} \in \mathbb{R}^M$$

I.e., set of Z nearest Wikipedia sentences from Search Level 2

Search Level 3

$$s''_z \in \arg Z \min_{s'_k} ||r_q - r_{(q,s'_k)}||_2$$

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

q = Query sequence

s = Support sequence

I.e., 1 Wikipedia sentence

Search Level 1

$$g(q) = r_q \in \mathbb{R}^M \longleftrightarrow g(s_1) = r_{s_1} \in \mathbb{R}^M$$

$$\vdots$$

$$g(s_{|\mathbb{D}|}) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$$

$$s_i \in \mathbb{D}$$

I.e., set of K nearest Wikipedia sentences

Search Level 2

$$r_q \longleftrightarrow g((q, s'_1)) = r_{(q,s'_1)} \in \mathbb{R}^M$$

$$\vdots$$

$$g((q, s'_K)) = r_{(q,s'_K)} \in \mathbb{R}^M$$

$$s'_k \in \arg\underset{s_i}{K\min} ||r_q - r_{s_i}||_2$$

I.e., set of Z nearest Wikipedia sentences from Search Level 2

Search Level 3

$$s''_z \in \arg\underset{s'_k}{Z\min} ||r_q - r_{(q,s'_k)}||_2$$

$$\hat{y} = \underset{y \in \{\text{Supports, Refutes, Unverifiable}\}}{\arg\min} ||r_q - r_{(y,q,s''_1,\ldots,s''_Z)}||_2$$

FEVER: MemMatch Model

- Approach (*high-level*): Run the same shared network, g, over all of Wikipedia, caching the representations, & then perform search by matching the query representation with progressively built-up support sequences

q = Query sequence

s = Support sequence

I.e., 1 Wikipedia sentence

Search Level 1

$$g(q) = r_q \in \mathbb{R}^M \longleftrightarrow g(s_1) = r_{s_1} \in \mathbb{R}^M$$

$$\vdots$$

$$g(s_{|\mathbb{D}|}) = r_{s_{|\mathbb{D}|}} \in \mathbb{R}^M$$

$$s_i \in \mathbb{D}$$

I.e., set of K nearest Wikipedia sentences

Search Level 2

$$r_q \longleftrightarrow g((q, s'_1)) = r_{(q,s'_1)} \in \mathbb{R}^M$$

$$\vdots$$

$$g((q, s'_K)) = r_{(q,s'_K)} \in \mathbb{R}^M$$

$$s'_k \in \arg K \min_{s_i} ||r_q - r_{s_i}||_2$$

I.e., set of Z nearest Wikipedia sentences from Search Level 2

Search Level 3

$$\hat{y} = \arg\min_{y \in \{\text{Supports, Refutes, Unverifiable}\}} ||r_q - r_{(y,q,s''_1,\ldots,s''_Z)}||_2$$

$$s''_z \in \arg Z \min_{s'_k} ||r_q - r_{(q,s'_k)}||_2$$

$\hat{y}$ is the label prediction

$\{s''_1, \ldots, s''_Z\}$ is the set of Wikipedia support sentences

Challenges Building End-to-end Neural Model

- Seek to produce dense representations from a deep neural network (Transformer) for similarity matching

Transformer LM

Millions of params & ~quadratic run time

- Problem: Computationally infeasible to run multiple passes of a deep Transformer over a large datastore (Wikipedia) for every new query

Would result in many 10's of millions of unique sequences

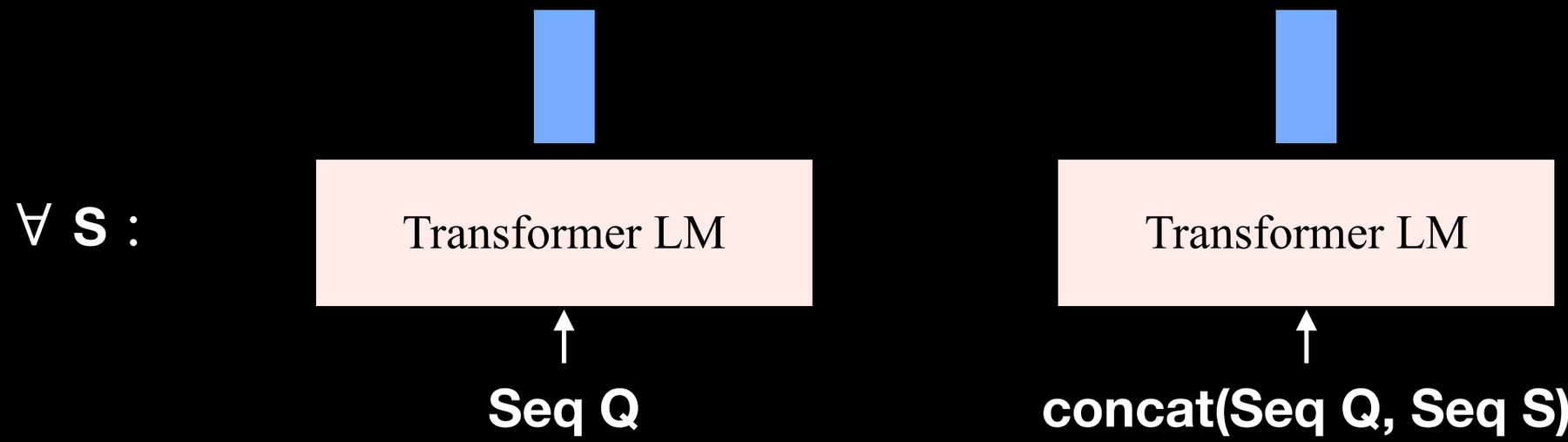
bi-encoder vs. cross-encoder dilemma

bi-encoder vs. cross-encoder dilemma

- Most effective to compose all relevant sequences in the input to the network **(cross-encoder)**

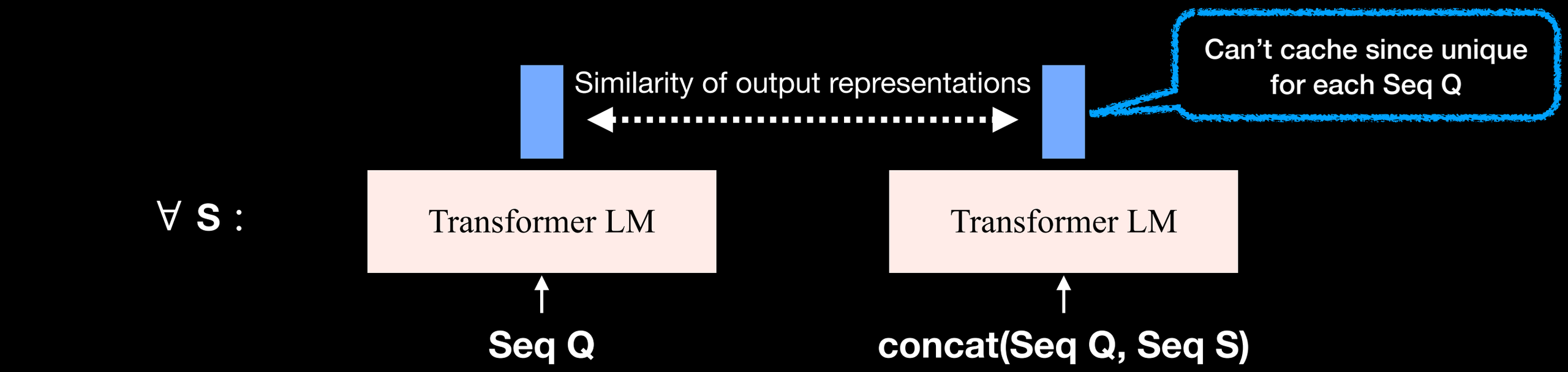
bi-encoder vs. cross-encoder dilemma

- Most effective to compose all relevant sequences in the input to the network **(cross-encoder)**



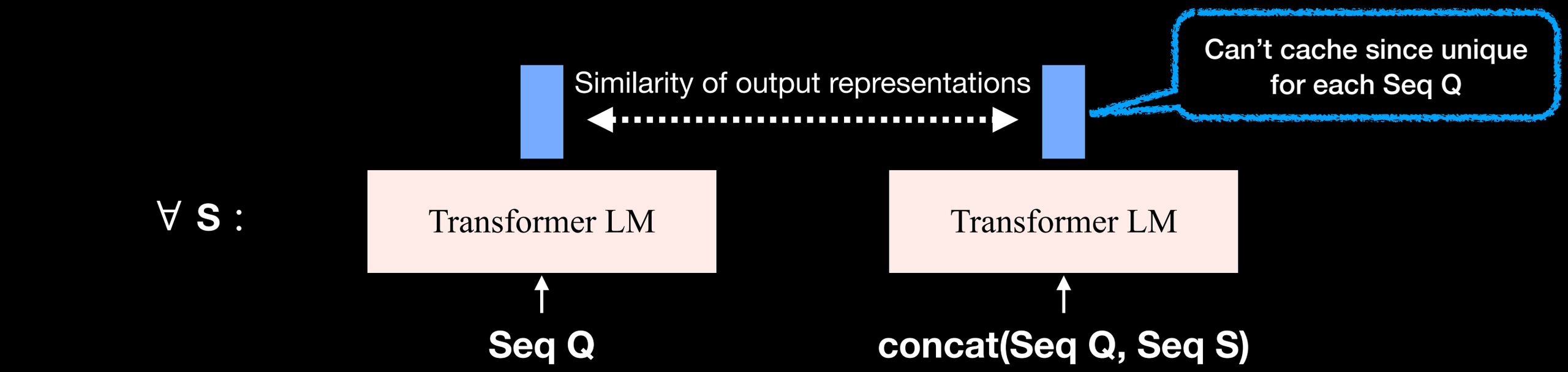
bi-encoder vs. cross-encoder dilemma

- Most effective to compose all relevant sequences in the input to the network **(cross-encoder)**



bi-encoder vs. cross-encoder dilemma

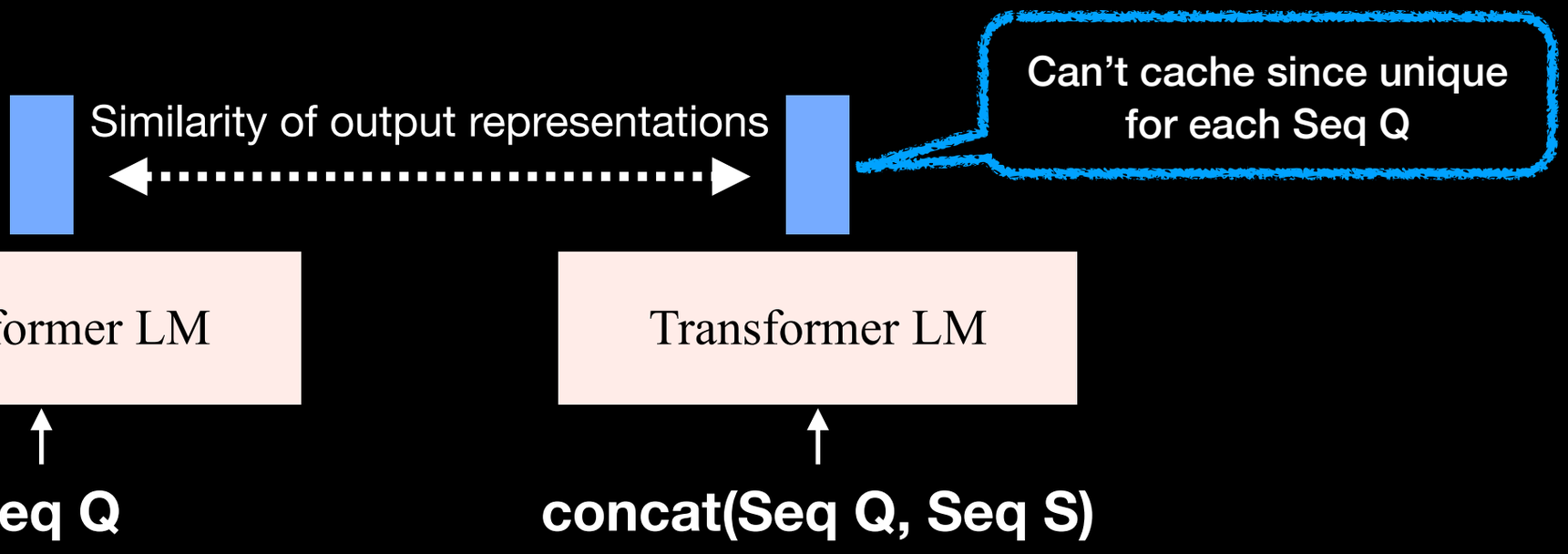
- Most effective to compose all relevant sequences in the input to the network **(cross-encoder)**



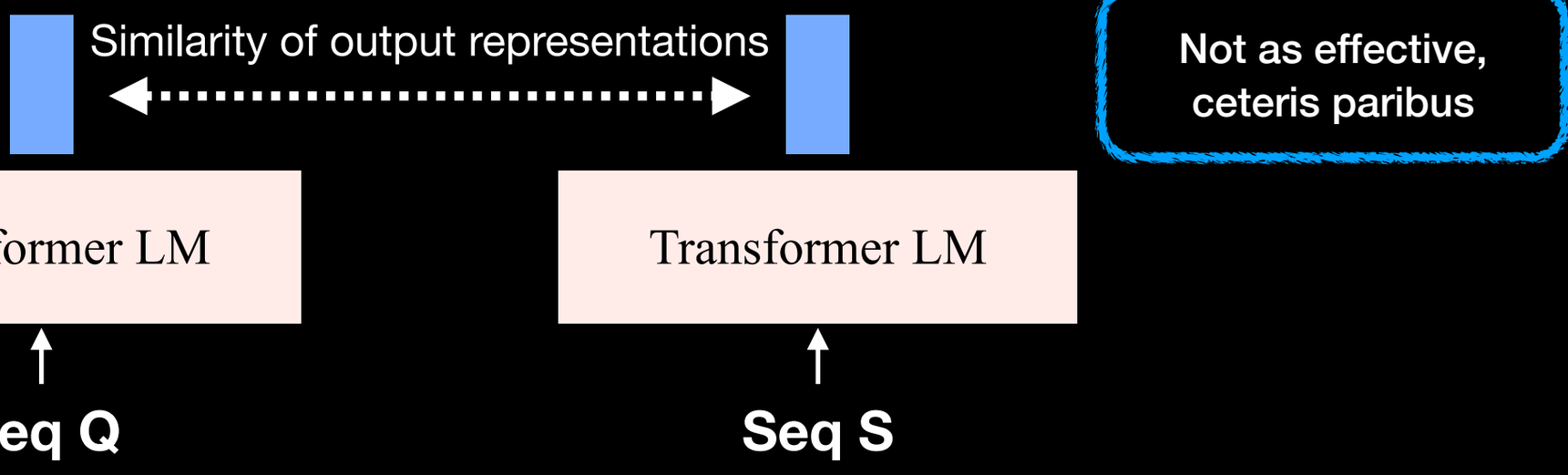
- Instead, resort to matching against representations from independent passes through the network **(bi-encoder)**

bi-encoder vs. cross-encoder dilemma

- Most effective to compose all relevant sequences in the input to the network **(cross-encoder)**

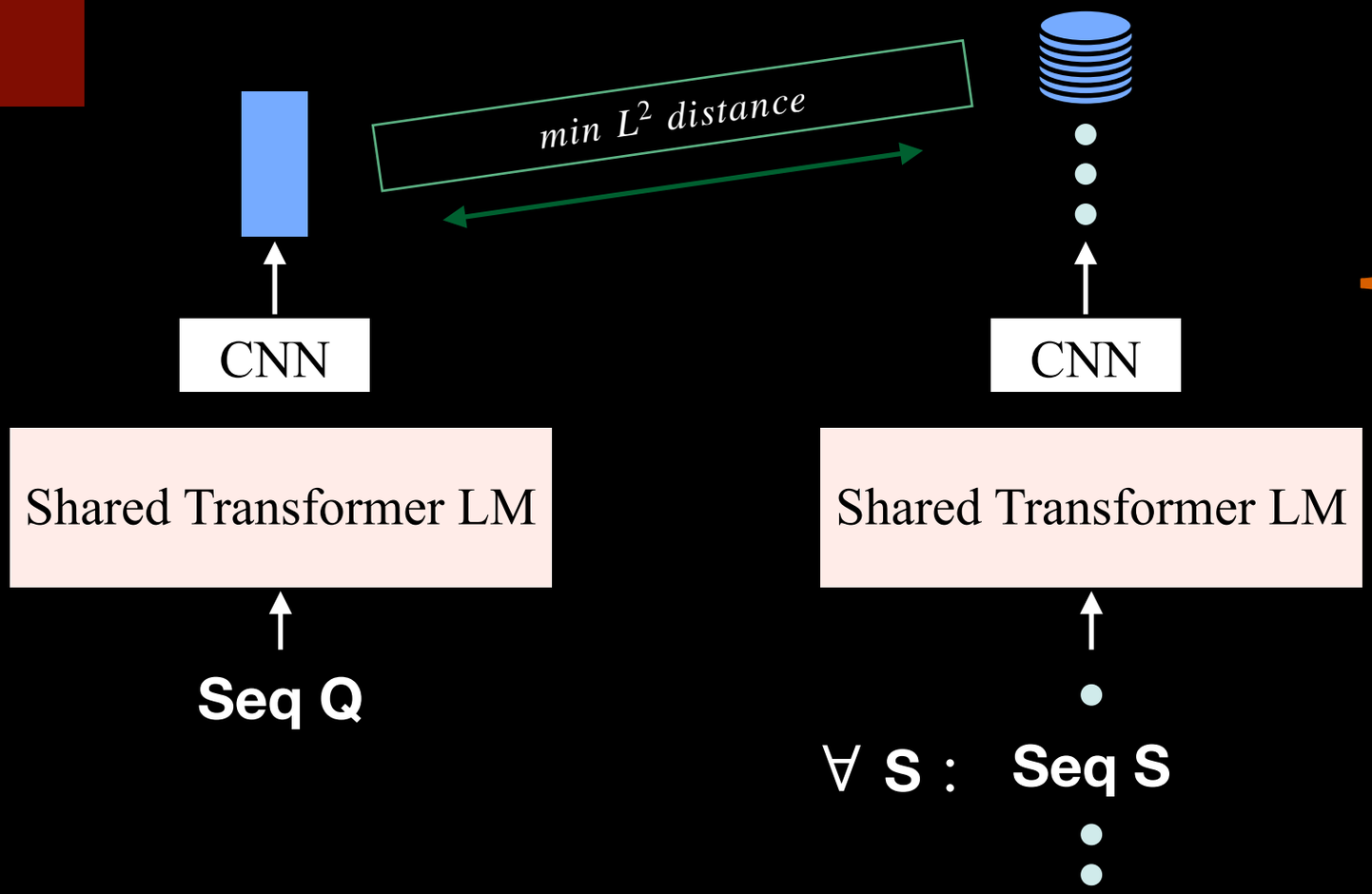


- Instead, resort to matching against representations from independent passes through the network **(bi-encoder)**



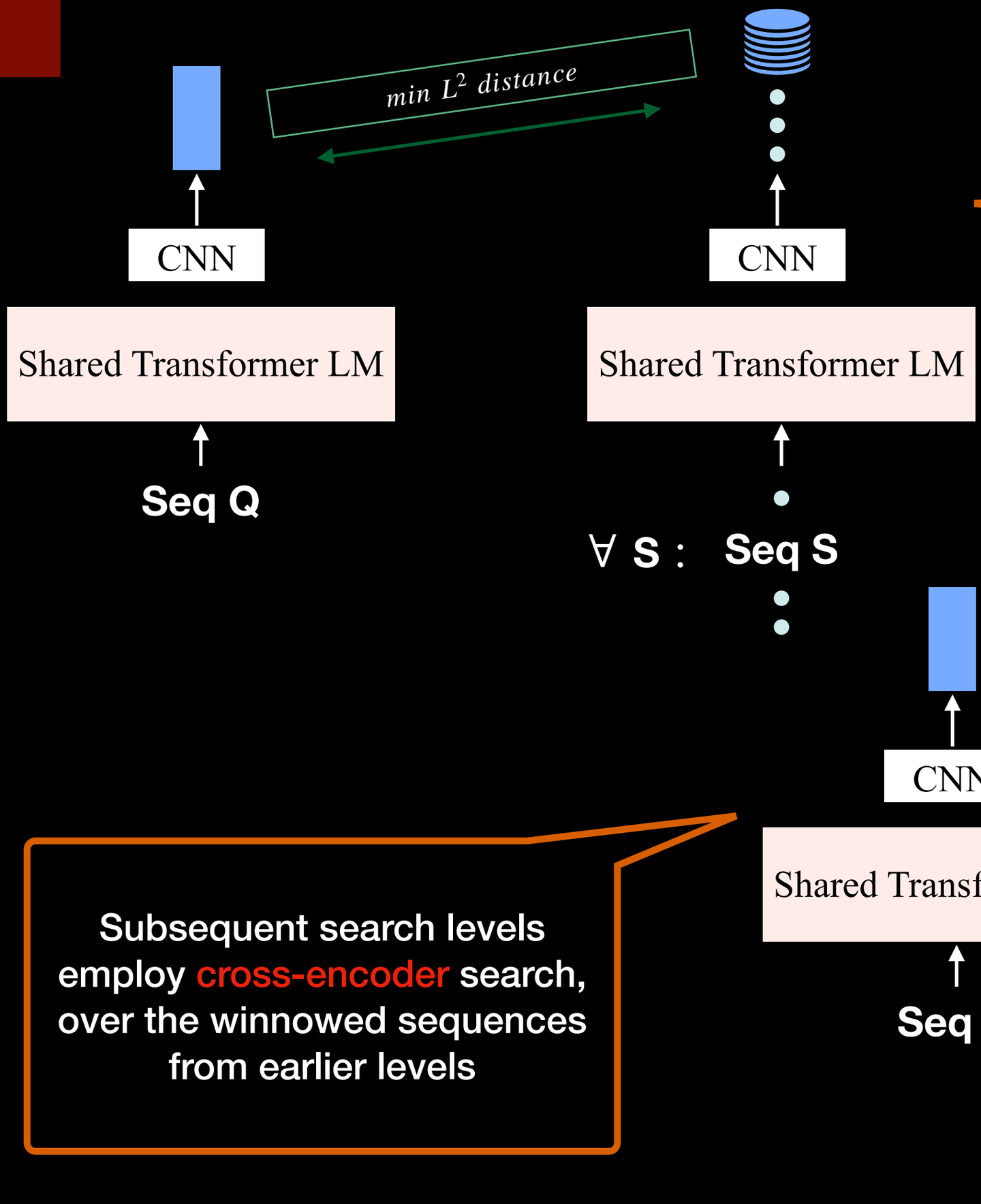
Solution: Add memory layers to Transformer & search over dense representations

Solution: Add memory layers to Transformer & search over dense representations



Initial, coarse bi-encoder search

Solution: Add memory layers to Transformer & search over dense representations



Initial, coarse bi-encoder search

Subsequent search levels employ cross-encoder search, over the winnowed sequences from earlier levels

Input Terminology

- Claim: Charles de Gaulle was a leader in the French Resistance.
  - **`Query sequence`**
- Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.
  - **`Support sequence`**
- During search, dynamically create **`Support sequences`**

Coarse-to-Fine Search

Coarse-to-Fine Search

- For each of 3 levels, we aim to minimize the distance between the **QUERY SEQUENCE** and the correct **SUPPORT SEQUENCE**

Coarse-to-Fine Search

- For each of 3 levels, we aim to minimize the distance between the **QUERY SEQUENCE** and the correct **SUPPORT SEQUENCE**

- Level 1 (bi-encoder retrieval)
  - Match [**claim**] to a [**Wikipedia sentence**]

Coarse-to-Fine Search

- For each of 3 levels, we aim to minimize the distance between the **QUERY SEQUENCE** and the correct **SUPPORT SEQUENCE**

- Level 1 (bi-encoder retrieval)
  - Match [**claim**] to a [**Wikipedia sentence**]

Coarse-to-Fine Search

- For each of 3 levels, we aim to minimize the distance between the **QUERY SEQUENCE** and the correct **SUPPORT SEQUENCE**

- Level 1 (bi-encoder retrieval)
  - Match [**claim**] to a [**Wikipedia sentence**]

- Level 2 (cross-encoder retrieval)
  - Match [**claim**] to a [**claim + Wikipedia sentence**]

Coarse-to-Fine Search

- For each of 3 levels, we aim to minimize the distance between the **QUERY SEQUENCE** and the correct **SUPPORT SEQUENCE**

- Level 1 (bi-encoder retrieval)
    - Match [**claim**] to a [**Wikipedia sentence**]
- Level 2 (cross-encoder retrieval)
    - Match [**claim**] to a [**claim + Wikipedia sentence**]

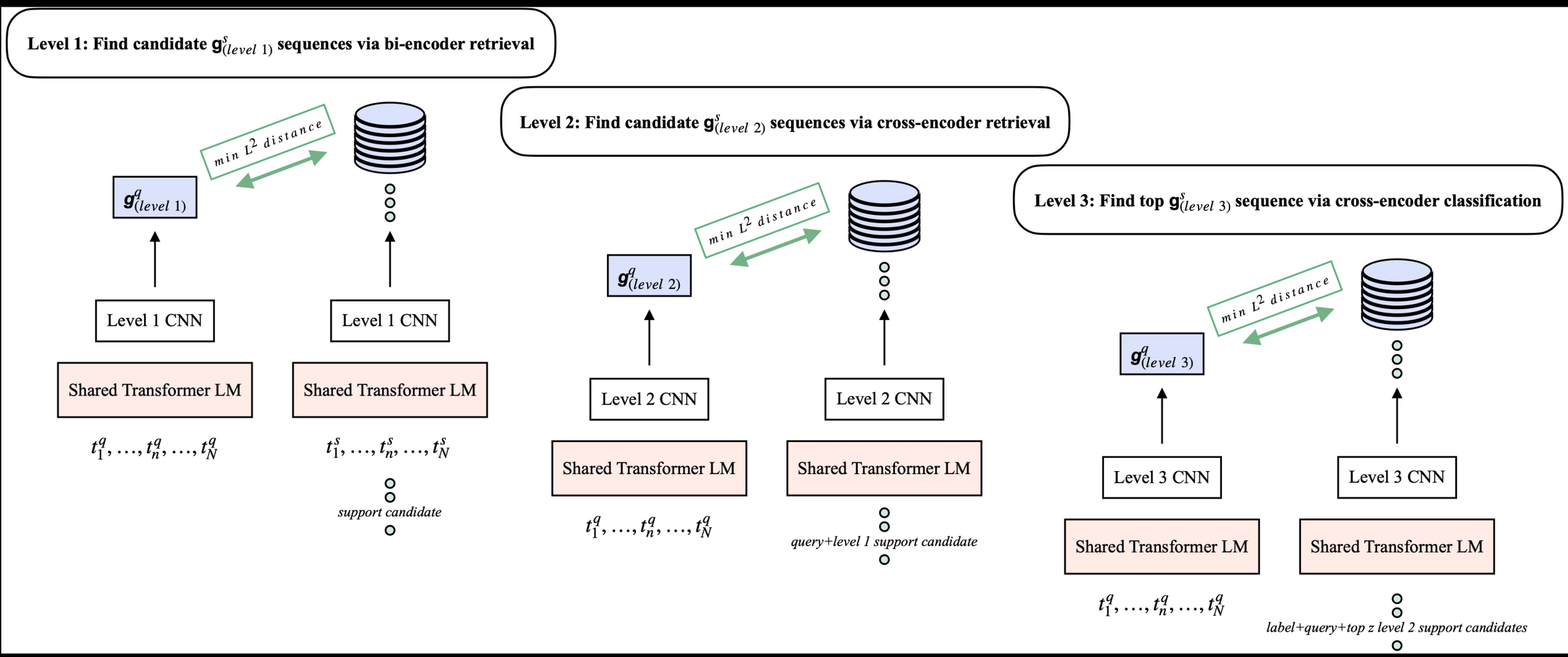
Coarse-to-Fine Search

- For each of 3 levels, we aim to minimize the distance between the **QUERY SEQUENCE** and the correct **SUPPORT SEQUENCE**

- Level 1 (bi-encoder retrieval)
  - Match [**claim**] to a [**Wikipedia sentence**]
- Level 2 (cross-encoder retrieval)
  - Match [**claim**] to a [**claim + Wikipedia sentence**]
- Level 3 (cross-encoder classification)
  - Match [**claim**] to a [**LABEL + claim + Wikipedia sentences**]

QUERY SEQUENCE **SUPPORT SEQUENCE**

Full Model


Level 1: Find candidate $\mathbf{g}^s_{(level\ 1)}$ sequences via bi-encoder retrieval
$min\ L^2\ distance$
$\mathbf{g}^q_{(level\ 1)}$
Level 1 CNN
Level 1 CNN
Shared Transformer LM
Shared Transformer LM
$t^q_1, \ldots, t^q_n, \ldots, t^q_N$
$t^s_1, \ldots, t^s_n, \ldots, t^s_N$
support candidate
Level 2: Find candidate $\mathbf{g}^s_{(level\ 2)}$ sequences via cross-encoder retrieval
$min\ L^2\ distance$
$\mathbf{g}^q_{(level\ 2)}$
Level 2 CNN
Level 2 CNN
Shared Transformer LM
Shared Transformer LM
$t^q_1, \ldots, t^q_n, \ldots, t^q_N$
query+level 1 support candidate
Level 3: Find top $\mathbf{g}^s_{(level\ 3)}$ sequence via cross-encoder classification
$min\ L^2\ distance$
$\mathbf{g}^q_{(level\ 3)}$
Level 3 CNN
Level 3 CNN
Shared Transformer LM
Shared Transformer LM
$t^q_1, \ldots, t^q_n, \ldots, t^q_N$
label+query+top z level 2 support candidates

Training

- Run coarse-to-fine search to find hard negatives and prediction sequences
  - Unlike typical supervised learning settings, training set is not static
- Loss MINIMIZES distance to CORRECT MATCHES
- Loss MAXIMIZES distance to WRONG MATCHES

Loss

Dense representation of query sequence:

$$\boldsymbol{g}^q \in \mathbb{R}^{1000} = \begin{bmatrix} g_1^q \\ g_2^q \\ \vdots \\ g_{1000}^q \end{bmatrix}$$

$\boldsymbol{g}^q_{(level\ L)}$

Minimize difference to correct matches

$$\boldsymbol{\delta}_L = \left| \boldsymbol{g}^q - \boldsymbol{g}^s \right| \in \mathbb{R}^M$$

$\boldsymbol{g}^s_{(level\ L)}$

Dense representation of support sequence:

$$\boldsymbol{g}^s \in \mathbb{R}^{1000} = \begin{bmatrix} g_1^s \\ g_2^s \\ \vdots \\ g_{1000}^s \end{bmatrix}$$

Level L CNN

Level L CNN

Shared Transformer LM

Shared Transformer LM

$t_1^q, \ldots, t_n^q, \ldots, t_N^q$

$t_1^s, \ldots, t_n^s, \ldots, t_N^s$

Loss

Maximize difference
to
incorrect matches

$\boldsymbol{g}^q_{(level\ L)}$

$\boldsymbol{g}^s_{(level\ L)}$

$\boldsymbol{\delta}_L = \left|\boldsymbol{g}^q - \boldsymbol{g}^s\right| \in \mathbb{R}^M$

Level L CNN

Level L CNN

Shared Transformer LM

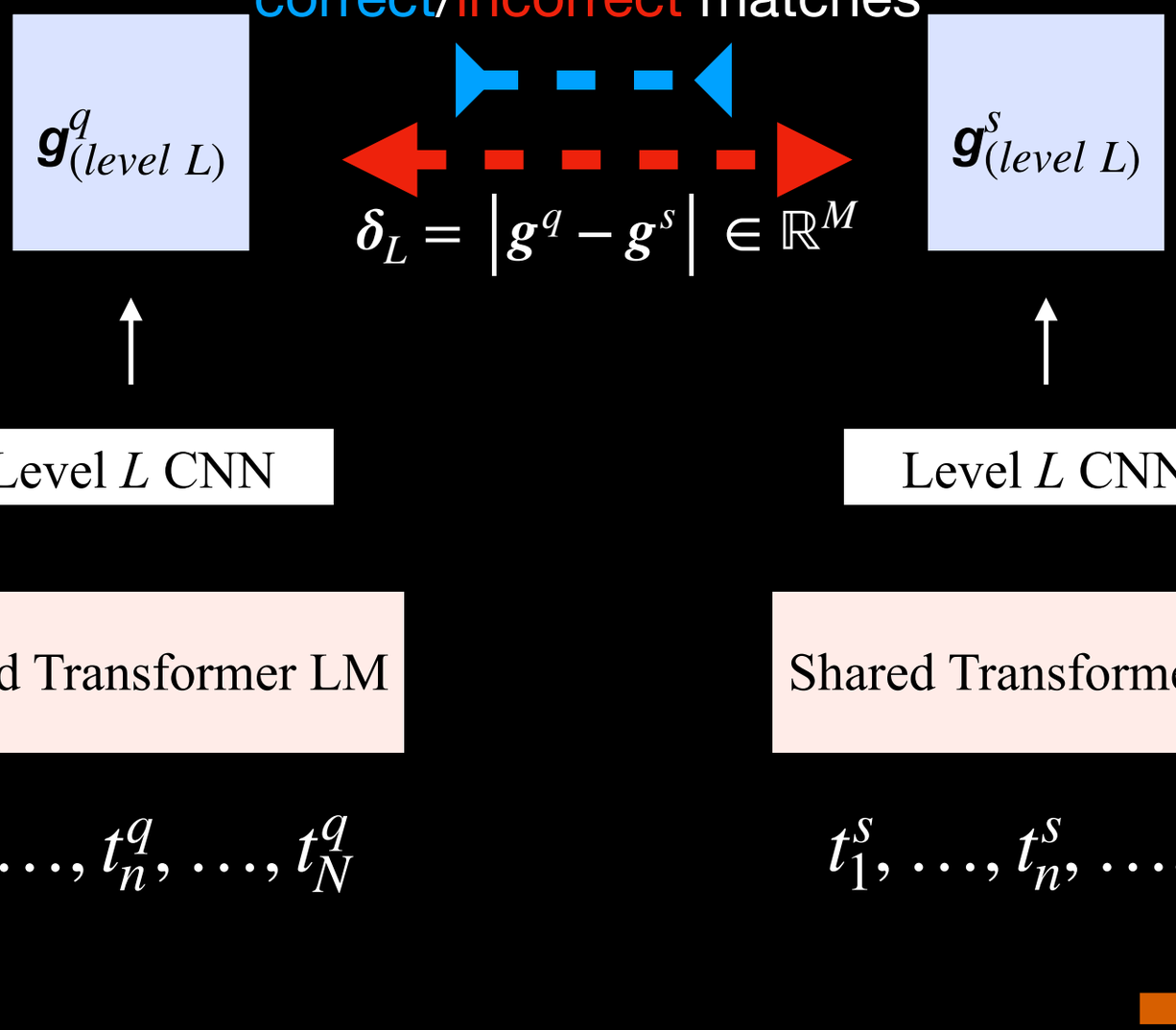
Shared Transformer LM

$t^q_1, \ldots, t^q_n, \ldots, t^q_N$

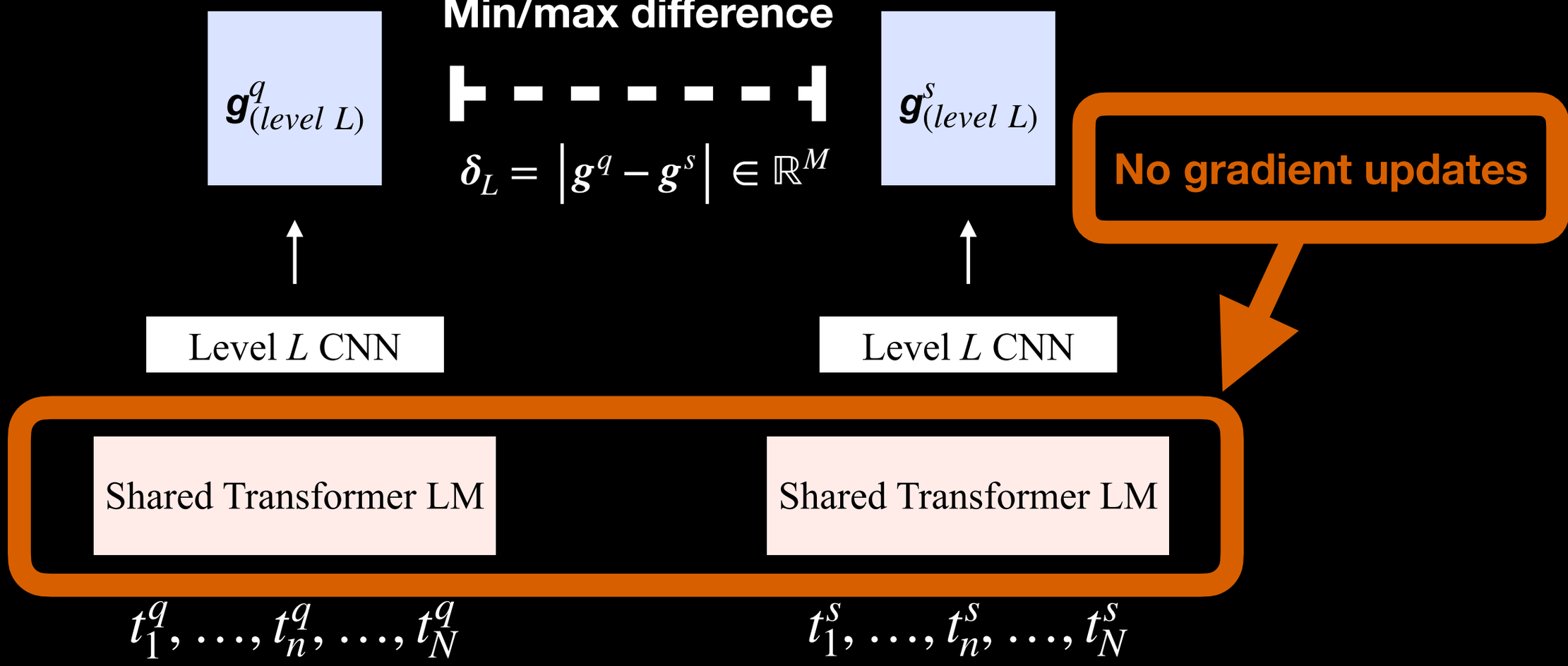
$t^s_1, \ldots, t^s_n, \ldots, t^s_N$

Loss

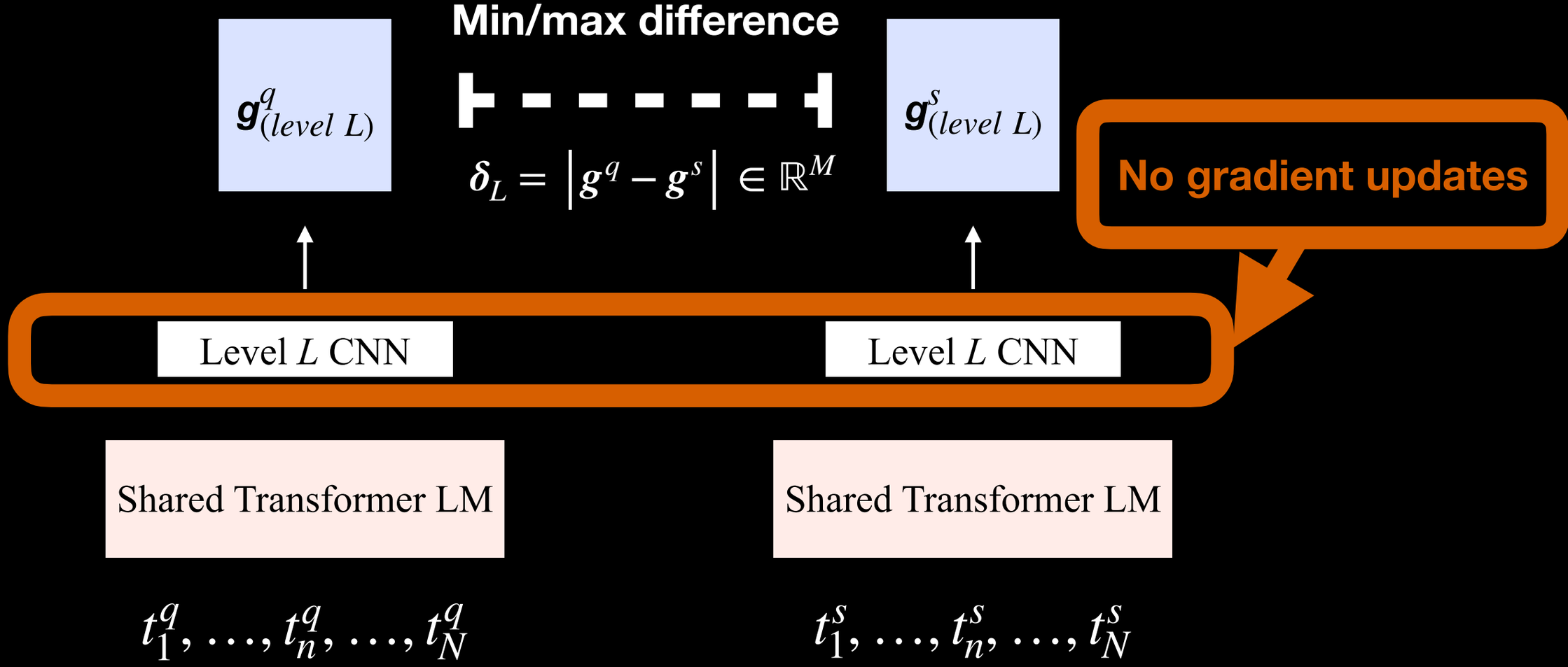
Minimize/maximize difference
to
correct/incorrect matches



Iterative Freezing: Epoch mod 2==0



Iterative Freezing: Epoch mod 2==1



Inference

- Run coarse-to-fine search

- Top level 3 support sequence contains the predicted classification label and evidence sentences

Example - Level 1

Level 1	
QUERY sequence	Claim: Charles de Gaulle was a leader in the French Resistance.
SUPPORT sequence, beam index 0	Evidence: French Resistance, sentence 0: The French Resistance (La Résistance) was the collection of French resistance movements that fought against the Nazi German occupation of France and against the collaborationist Vichy régime during the Second World War.
SUPPORT sequence, beam index 1	Evidence: Charles de Gaulle, sentence 1: He was the leader of Free France (1940 – 44) and the head of the Provisional Government of the French Republic (1944 – 46).
⋮	⋮
SUPPORT sequence, beam index 14	Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.
⋮	⋮
SUPPORT sequence, beam index 99	Evidence: Resistance (EP), sentence 7: This EP or mini-album sold nearly all of its 200,000 copies.

The ground-truth evidence sentence is in the 15th beam position in level 1.

Example - Level 2

Level 2	
QUERY sequence	Consider: Claim: Charles de Gaulle was a leader in the French Resistance.
SUPPORT sequence, beam index 0	Consider: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 1: He was the leader of Free France (1940 – 44) and the head of the Provisional Government of the French Republic (1944 – 46).
SUPPORT sequence, beam index 1	Consider: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.
SUPPORT sequence, beam index 2	Consider: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 0: Charles André Joseph Marie de Gaulle ([ʃaʁl də gol]; 22 November 1890 – 9 November 1970) was a French general and statesman.

The ground-truth evidence sentence rises to the 2nd beam position in level 2 (i.e., cross-encoding is important).

Example - Level 3

Level 3	
QUERY sequence	Predict: Claim: Charles de Gaulle was a leader in the French Resistance.
SUPPORT sequence, beam index 0	Supports: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 1: He was the leader of Free France (1940 – 44) and the head of the Provisional Government of the French Republic (1944 – 46). Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance. Evidence: Charles de Gaulle, sentence 0: Charles André Joseph Marie de Gaulle ([ʃaʁl də gol]; 22 November 1890 – 9 November 1970) was a French general and statesman.
SUPPORT sequence, beam index 1	Unverifiable: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 1: He was the leader of Free France (1940 – 44) and the head of the Provisional Government of the French Republic (1944 – 46). Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance. Evidence: Charles de Gaulle, sentence 0: Charles André Joseph Marie de Gaulle ([ʃaʁl də gol]; 22 November 1890 – 9 November 1970) was a French general and statesman.
SUPPORT sequence, beam index 2	Refutes: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 1: He was the leader of Free France (1940 – 44) and the head of the Provisional Government of the French Republic (1944 – 46). Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance. Evidence: Charles de Gaulle, sentence 0: Charles André Joseph Marie de Gaulle ([ʃaʁl də gol]; 22 November 1890 – 9 November 1970) was a French general and statesman.

In level 3, we include the classification label and the top 3 predicted evidence sentences from level 2. The final prediction is the top of the level 3 beam.

Example - Level 3 - training

Level 3 (training only)	
QUERY sequence	Reference: Claim: Charles de Gaulle was a leader in the French Resistance.
SUPPORT sequence, positive training instance	Supports: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.
SUPPORT sequence, negative training instance	Refutes: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.
SUPPORT sequence, negative training instance	Unverifiable: Claim: Charles de Gaulle was a leader in the French Resistance. Evidence: Charles de Gaulle, sentence 12: Despite frosty relations with Britain and especially the United States, he emerged as the undisputed leader of the French resistance.

For training, in level 3 we also* create positive and negative instances by flipping the label on the ground-truth evidence sentences.

(*in addition to hard negatives and predictions from search)

Empirical Comparisons

Model	Acc.	FEV.	Pt.ct.	IR	Ling.
GEAR	71.60	67.10	> 204	•	•
DREAM	76.85	70.60	$\geq (373, 833]$	•	•
CompoundLabel	66.21	61.65	18	•	
NSMN	68.16	64.23	28	•	•
$\text{BERT}_{\text{LARGE}}$	38.	N/A	340		
$\text{BERT}_{\text{LARGE}}$+FT	57.	N/A	340		
$\text{BERT}_{\text{LARGE}}$+KBFeat	49.	N/A	> 340		•
RAG	72.5	N/A	626		
$\text{BERT}_{\text{BASE}}$+MemMatch	70.42	63.95	120		

FEVER hidden test results, with light-gray rows indicating end-to-end models
Accuracy (Acc.); FEVER score (FEV.); Parameter estimates, in millions (Pt.ct.); Non-neural IR features (IR); Linguistic tools (Ling.); Our model is **$\text{BERT}_{\text{BASE}}$ + MemMatch**

Empirical Comparisons

Model	Acc.	FEV.	Pt.ct.	IR	Ling.
GEAR	71.60	67.10	> 204	•	•
DREAM	76.85	70.60	$\geq (373, 833]$	•	•
CompoundLabel	66.21	61.65	18	•	
NSMN	68.16	64.23	28	•	•
$\text{BERT}_{\text{LARGE}}$	38.	N/A	340		
$\text{BERT}_{\text{LARGE}}$+FT	57.	N/A	340		
$\text{BERT}_{\text{LARGE}}$+KBFeat	49.	N/A	> 340		•
RAG	72.5	N/A	626		
$\text{BERT}_{\text{BASE}}$+MemMatch	70.42	63.95	120		

FEVER hidden test results, with light-gray rows indicating end-to-end models
Accuracy (Acc.); FEVER score (FEV.); Parameter estimates, in millions (Pt.ct.); Non-neural IR features (IR); Linguistic tools (Ling.); Our model is **$\text{BERT}_{\text{BASE}}$ + MemMatch**

Comparisons with higher parameter variants to appear in v2

Analysis

Analysis Properties

- Level distances: Can use distances at each search level to analyze and constrain the model

- Exemplar auditing: Can create exemplar vectors via the differences between the query and support sequences

Level Distances

Euclidean distance for nearest predicted matches at each level (retrieval & classification)

$\boldsymbol{g}^{q}_{(level\ L)}$

$$distance(\boldsymbol{g}^{\mathbf{q}}, \boldsymbol{g}^{\mathbf{s}}) = \|\boldsymbol{g}^{\mathbf{q}} - \boldsymbol{g}^{\mathbf{s}}\|_2 \in \mathbb{R}^1$$

$\boldsymbol{g}^{s}_{(level\ L)}$

Level L CNN

Level L CNN

Shared Transformer LM

Shared Transformer LM

$t^q_1, \ldots, t^q_n, \ldots, t^q_N$

$t^s_1, \ldots, t^s_n, \ldots, t^s_N$

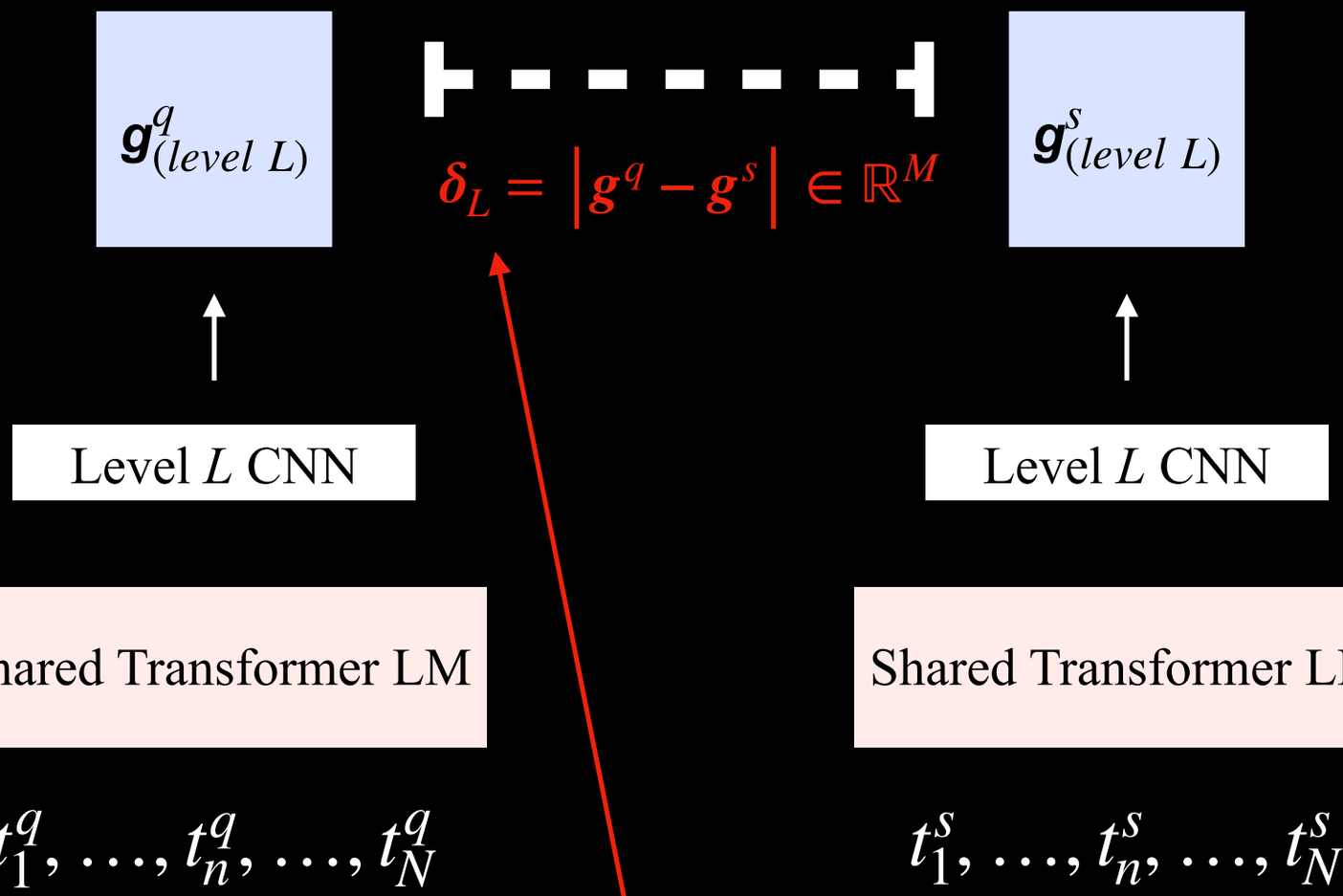
Analysis: Level Distances

- Smaller level distances associated with more reliable predictions on challenge set that modifies Wikipedia and claims

	Claim id 2024540000004 (reference label: FALSE)
$\text{SUPPORT}_{\text{LEVEL3}}$ sequence Level 2 distance: 200.89 Level 3 distance: 2.39	Supports: Claim: Tinker Tailor Soldier Spy is an espionage film. Evidence: Tinker Tailor Soldier Spy (film), sentence 0: Tinker Tailor Soldier Spy is a 2011 music video by Tomas Alfredson.
	Claim id 1390370000004 (reference label: TRUE)
$\text{SUPPORT}_{\text{LEVEL3}}$ sequence Level 2 distance: 0.05 Level 3 distance: 0.54	Supports: Claim: Star Trek: Discovery is an album. Evidence: Star Trek: Discovery, sentence 0: Star Trek: Discovery is an upcoming music album of Lady Gaga.

The mean level 2 distance from the training set at the top of the beam, given a correct retrieval, is 0.49 (+/- 4.75), and the mean level 3 distance, given a correct classification, is 0.92 (+/- 1.80). Incorrect classification labels and distances > mean are in red.

Exemplar Auditing



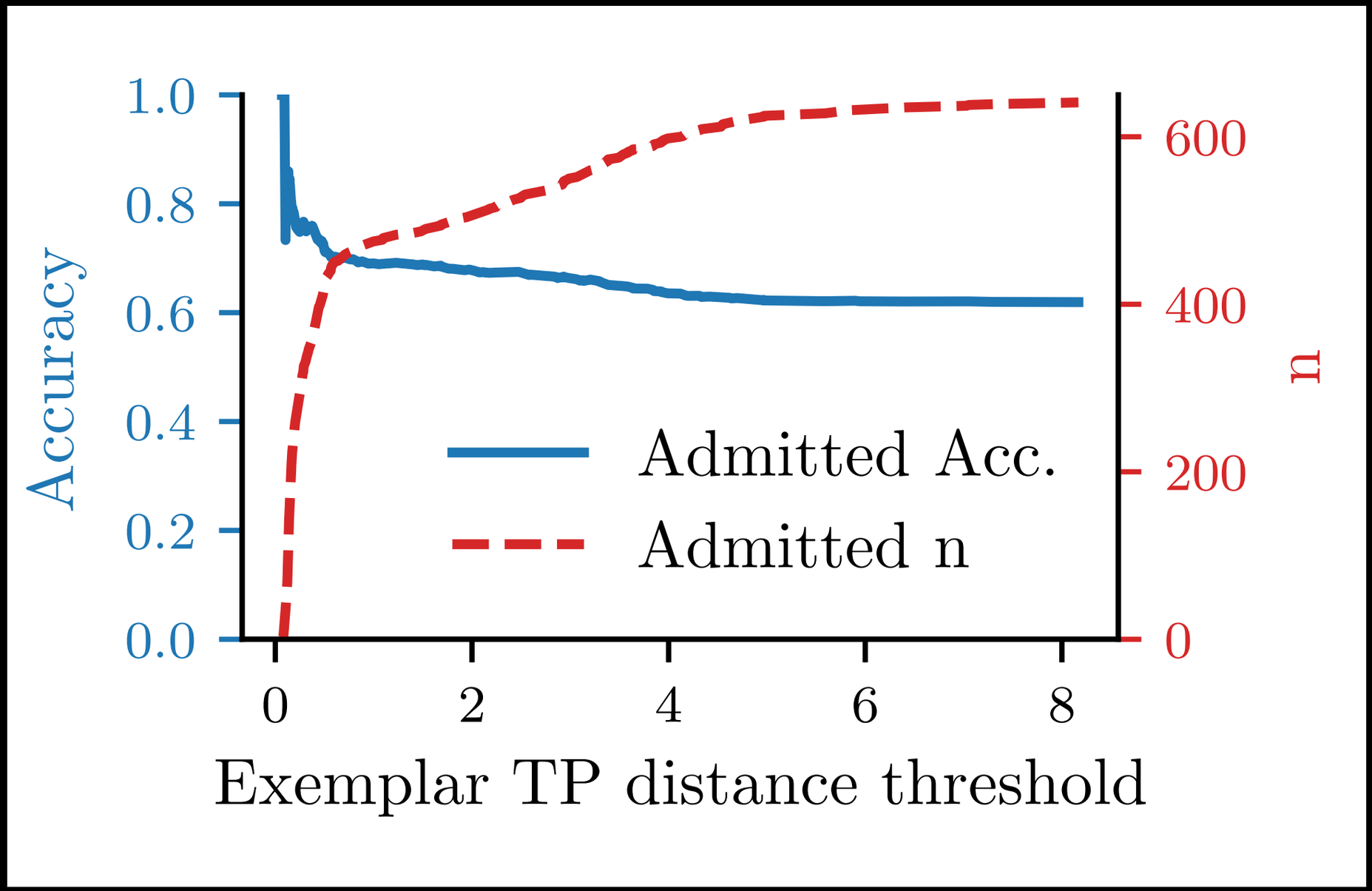
- Create exemplar vectors from difference vectors from final 2 levels:
  - concat(δ_2, δ_3)

Note: Exemplar database is distinct from datastores created during the coarse-to-fine search

Exemplar Auditing

- Associate test with nearest exemplar vectors in the database, only admitting predictions for True Positive (TP) exemplars
- Instances matched to TP exemplars with closer distances are associated with more accurate model predictions on challenge set

Exemplar Auditing



Exemplar Auditing

	Claim id 147493 (reference label: TRUE)
Test $\text{SUPPORT}_{\text{LEVEL3}}$ sequence	`Supports: Claim: T2 Trainspotting is set in and around a city. Evidence: T2 Trainspotting, sentence 0: T2 Trainspotting is a 2017 British comedy drama film, set in and around Edinburgh, Scotland.`
Exemplar $\text{SUPPORT}_{\text{LEVEL3}}$ sequence Exemplar distance: 0.14	`Supports: Claim: All My Children is set in a fictional suburb of a city. Evidence: All My Children, sentence 1: Created by Agnes Nixon, All My Children is set in Pine Valley, Pennsylvania, a fictional suburb of Philadelphia, which is modeled on the actual Philadelphia suburb of Rosemont.`
	Claim id 166634 (reference label: FALSE)
Test $\text{SUPPORT}_{\text{LEVEL3}}$ sequence	`Refutes: Claim: Anne Rice was born in Japan. Evidence: Anne Rice, sentence 5: Born in New Orleans, Rice spent much of her early life there before moving to Texas, and later to San Francisco.`
Exemplar $\text{SUPPORT}_{\text{LEVEL3}}$ sequence Exemplar distance: 0.21	`Refutes: Claim: Emma Stone was born in Taiwan. Evidence: Emma Stone, sentence 5: Born and raised in Scottsdale, Arizona, Stone began acting as a child, in a theater production of The Wind in the Willows in 2000.`

The close distance mappings between the test and exemplar instances tend to exhibit similar abstract, relational patterns

Exemplar Auditing

- Can update the model behavior by modifying the labels and/or instances in the exemplar database
- We illustrate this behavior on the challenge set (see paper for details)

Discussion

Broader Implications

- In principle, model can be frozen and substituted in for other settings using pre-trained Transformers, but with the new retrieval and analysis functionalities

- Many real-world tasks can be re-cast to the retrieval-classification setting

  - EHR data, Proteins, Medical QA, …

- Prospectively, multi-hop reasoning and even generation can be re-cast to this setting via a deeper search graph

  - Analysis advantages + offload model capacity to datastores

Future Work

Future Work: Application to Standard Classification Tasks

- Support sequences can be constructed from applicable data (Wikipedia, reference texts, knowledge bases, etc.) making the setting analogous to explicit retrieval-classification tasks.

- Alternatively, we can construct support sequences from nearest neighbors from the existing training set dynamically during training.

Future Work: Application to Standard Classification Tasks

- Support sequences can be constructed from applicable data (Wikipedia, reference texts, knowledge bases, etc.) making the setting analogous to explicit retrieval-classification tasks.

- Alternatively, we can construct support sequences from nearest neighbors from the existing training set dynamically during training.

In a sense, we can recurse the “auditing” process 1 level, with the model learning a mapping to the top-k nearest via the cross-encoder. In principle, up to computational limitations, we could recurse multiple times (matching of (matching (of matching….))).

Future Work

- Additional understanding of learning dynamics
    - Identifying structures/architectures with the inductive bias most conducive to comparing distances across representations
- Connections to kernel machines

cf., arc-cosine kernel (Cho and Saul 2011; Alber et al. 2017; inter alia.)

- Connections/comparisons to influence functions

E.g., we can cache representations/ distances dynamically during training, within & between epochs

https://arxiv.org/abs/2012.02287

Aside: Update coming in the spring, with efficiency improvements and demonstrations of how to apply this to standard classification tasks.

Additional Considerations

- Why are we using a CNN over the network to create representations rather than using/unfolding the full network, or a sample, thereof?

- When would we want to update/adapt a model just by matching to new instances rather than training a lightweight model over the representations (diff vectors), or re-training the full network?