



Question Duplicates

How old are you?



Question Duplicates

How old are you?

What is your age?



Question Duplicates

How old are you? = What is your age?



Question Duplicates

How old are you? = What is your age?

Where are you from?

Where are you going?



Question Duplicates

How old are you? = What is your age?

Where are you from? ≠ Where are you going?



What do Siamese Networks learn?

I am happy because I am learning



Classification: categorize things

Siamese Networks: Identify similarity between things

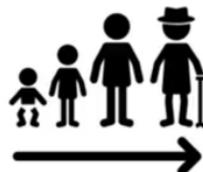
What do Siamese Networks learn?

I am happy because I am learning



Classification: categorize things

Siamese Networks: Identify similarity between things



What is your age?
How old are you?

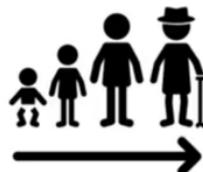
What do Siamese Networks learn?

I am happy because I am learning



Classification: categorize things

Siamese Networks: Identify similarity between things



What is your age?
How old are you?

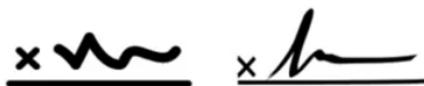


Difference or
Similarity

Siamese Networks in NLP



Handwritten checks



What is your age?
How old are you?

Question duplicates



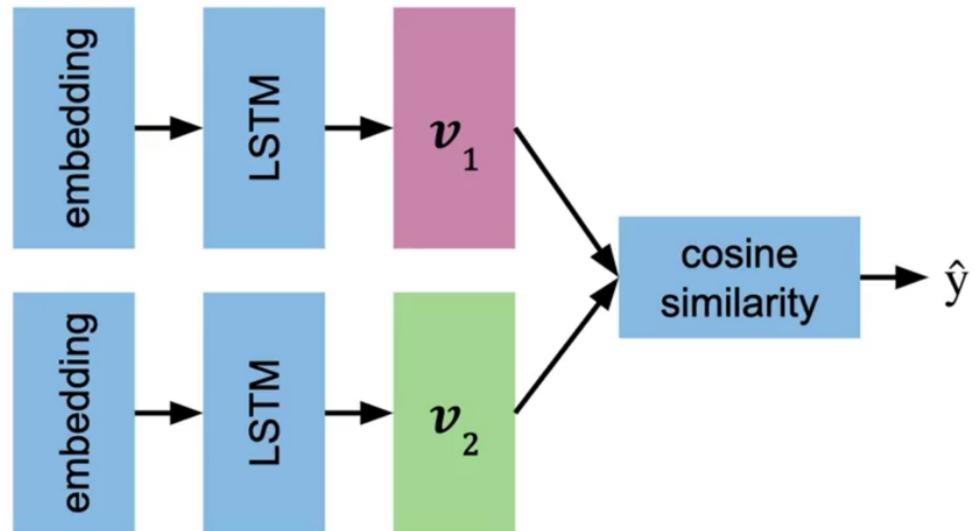
Queries

X

Architecture

Model Architecture

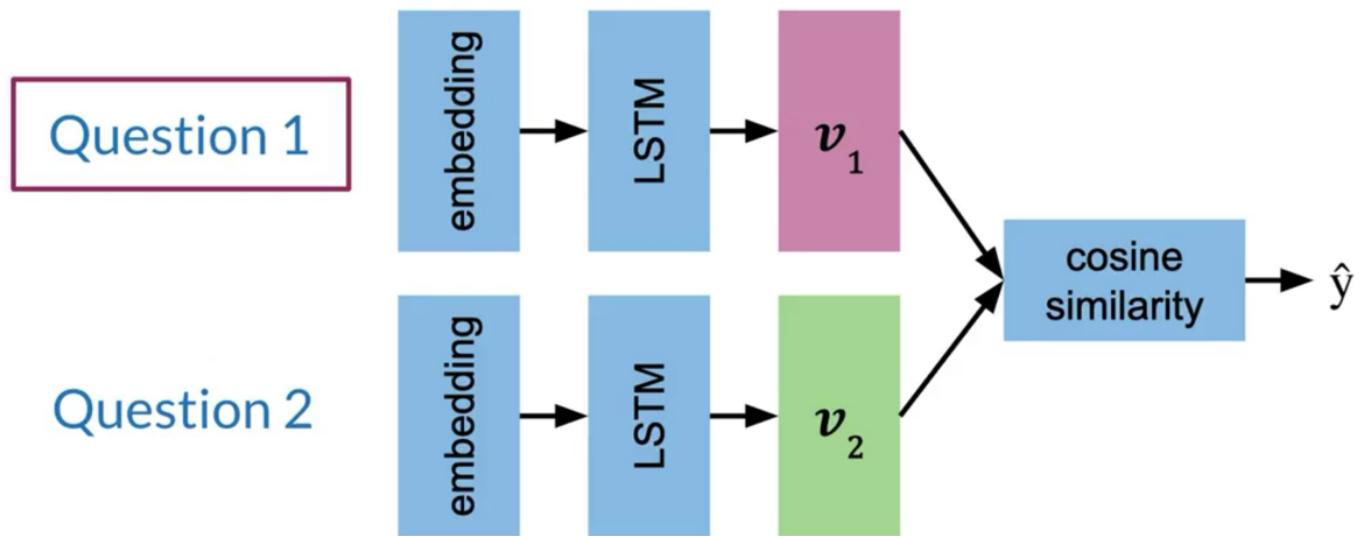
Question 1



Question 2

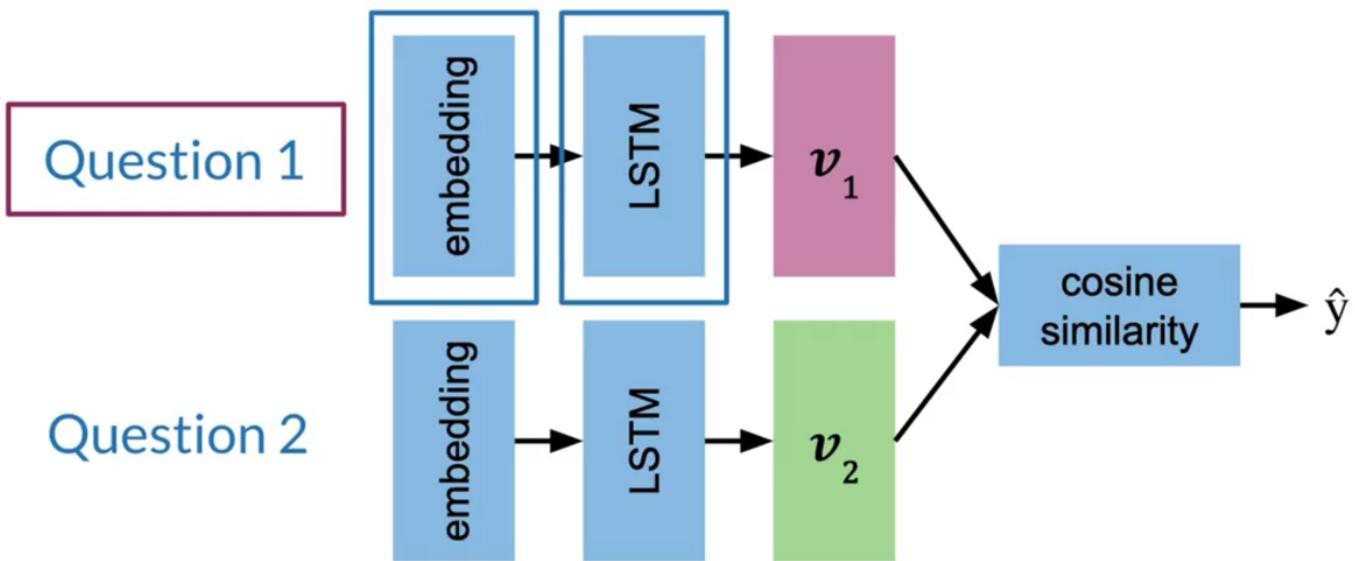
Architecture

Model Architecture



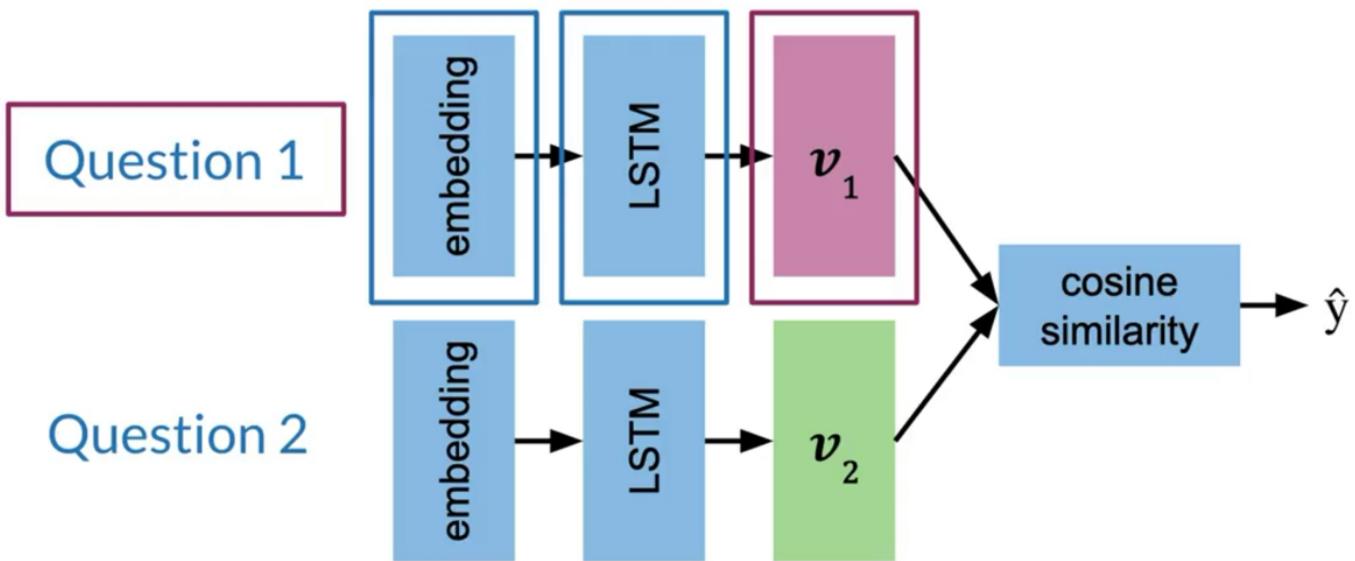
Architecture

Model Architecture



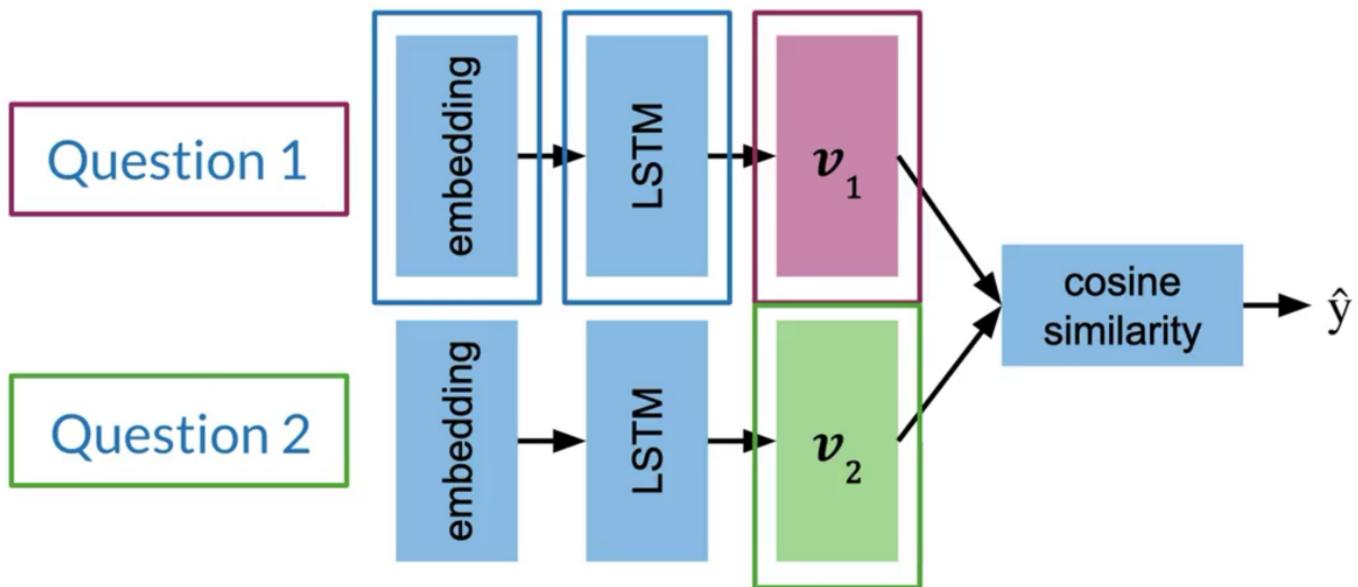
Architecture

Model Architecture



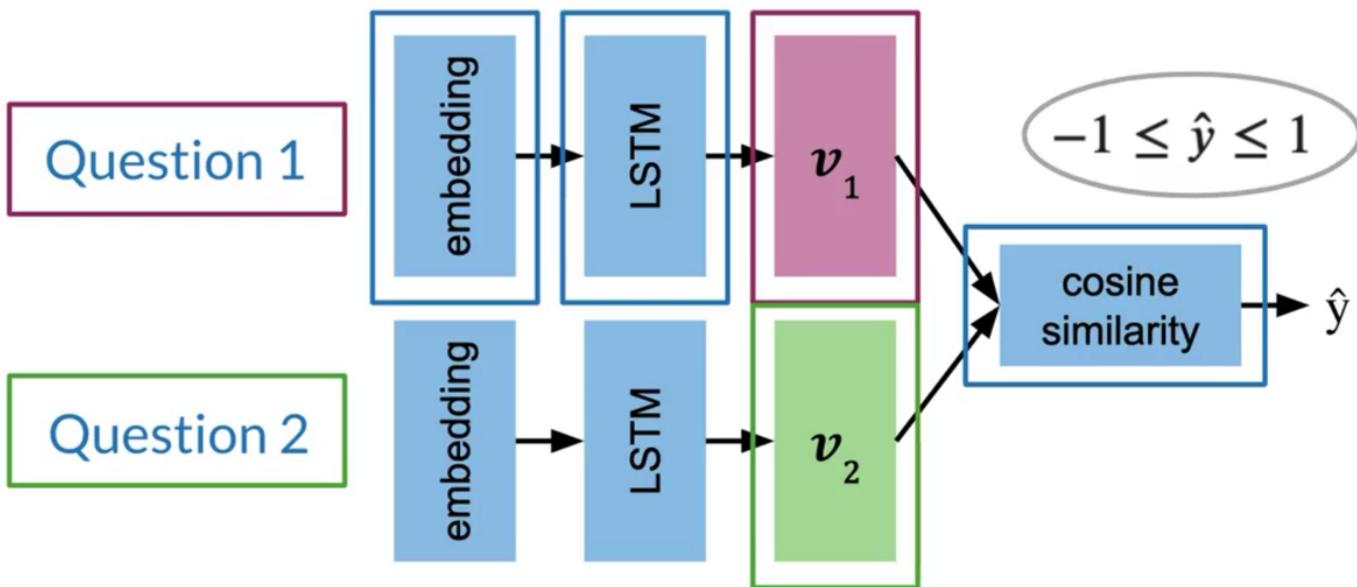
Architecture

Model Architecture



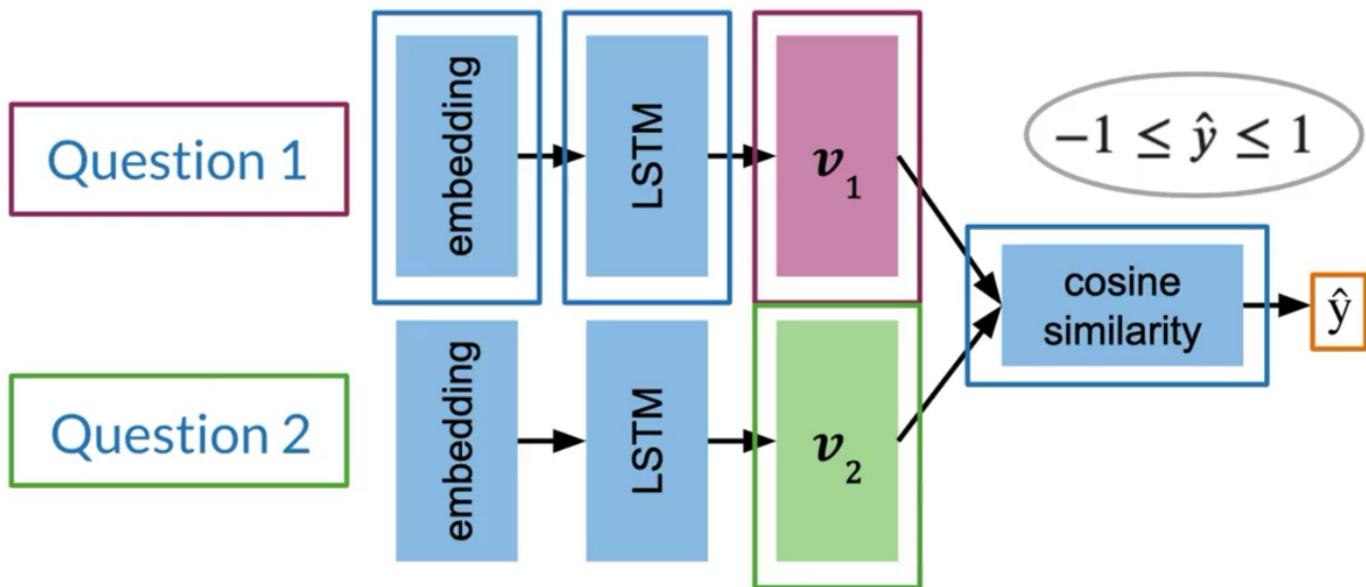
Architecture

Model Architecture



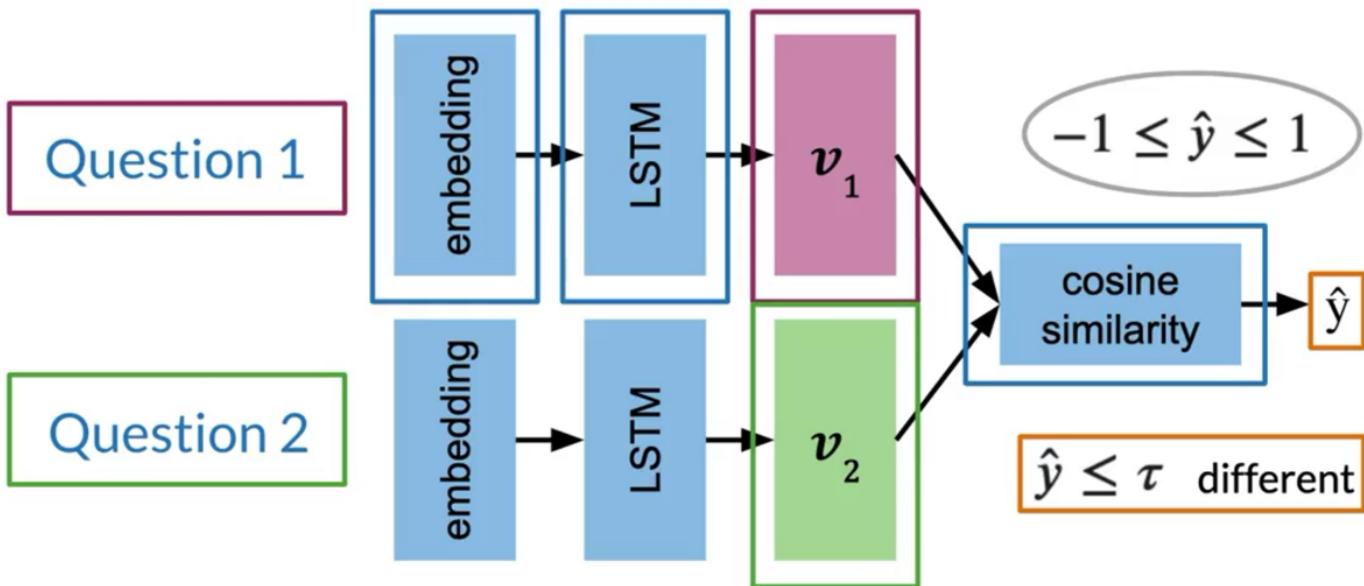
Architecture

Model Architecture



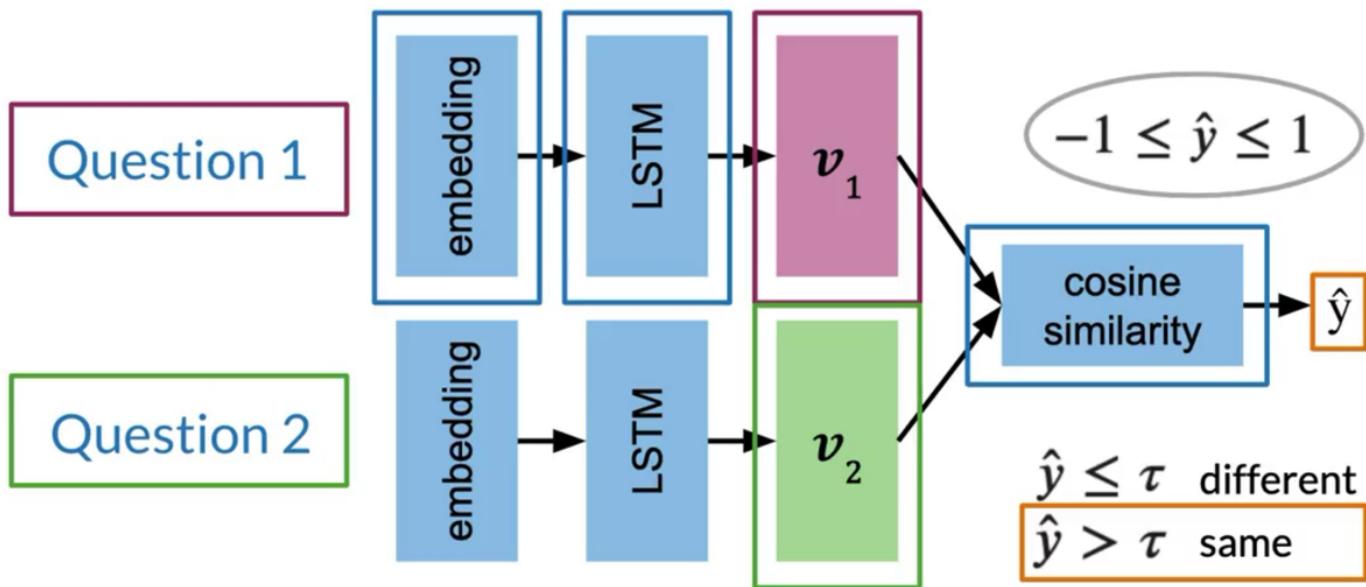
Architecture

Model Architecture



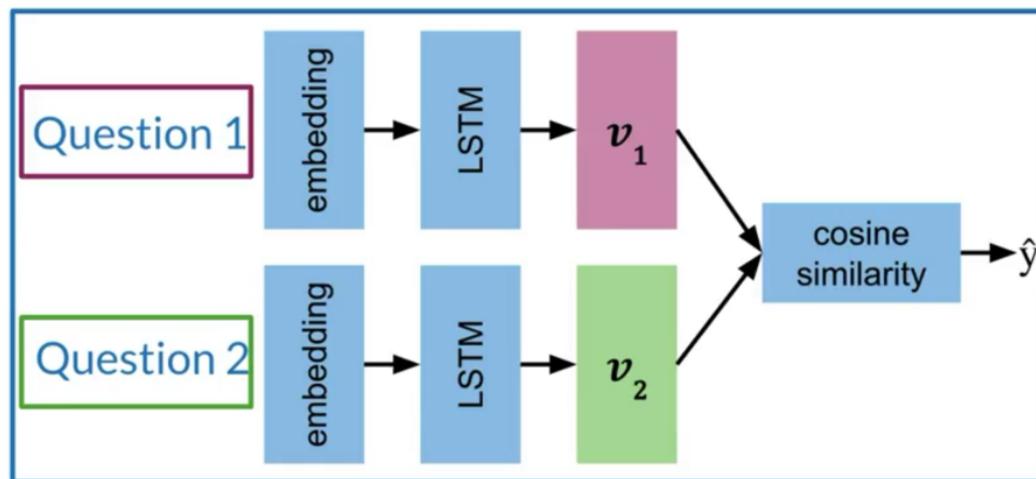
Architecture

Model Architecture



Architecture

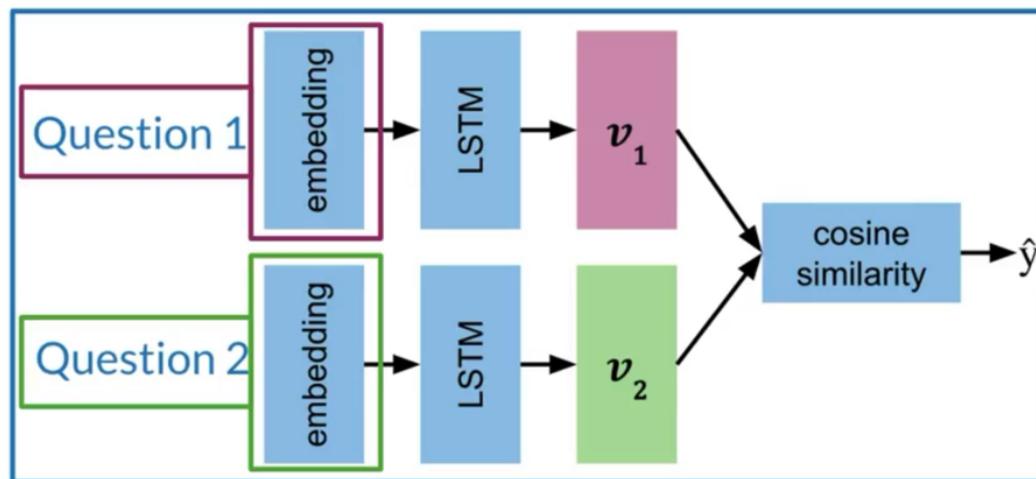
Model Architecture



1) Inputs

Architecture

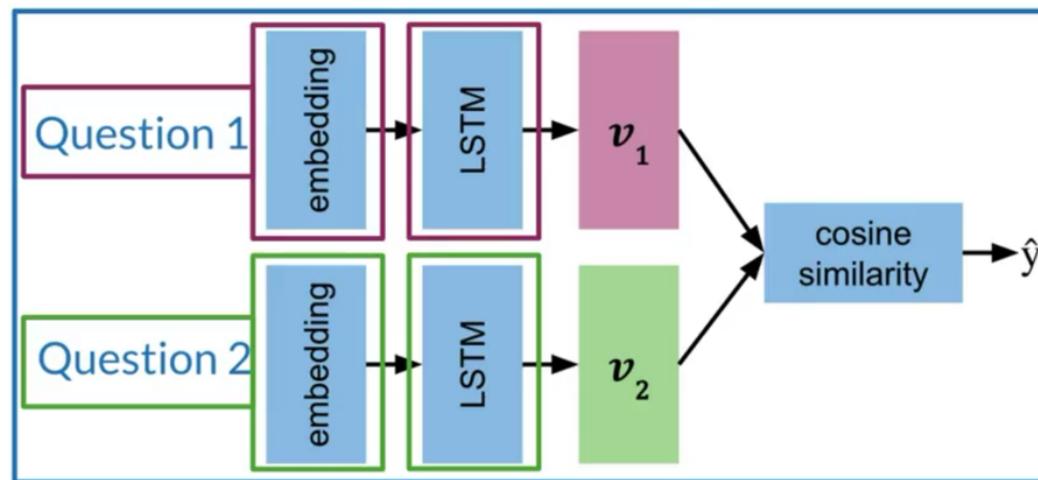
Model Architecture



- 1) Inputs
- 2) Embedding

Architecture

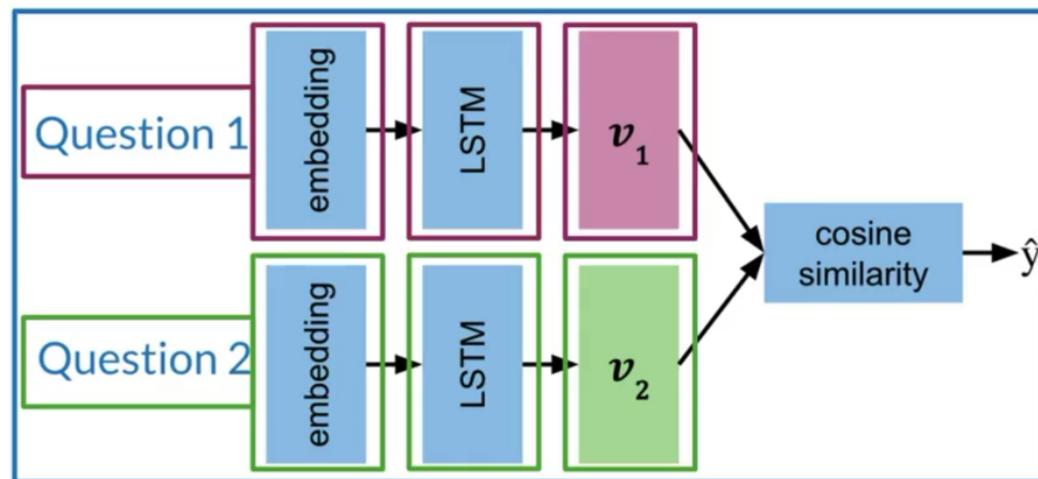
Model Architecture



- 1) Inputs
- 2) Embedding
- 3) LSTM

Architecture

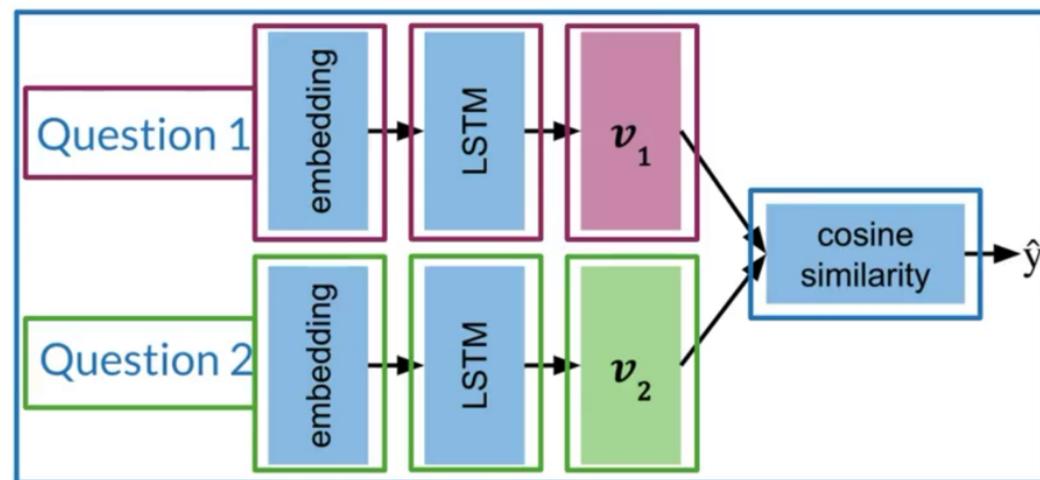
Model Architecture



- 1) Inputs
- 2) Embedding
- 3) LSTM
- 4) Vectors

Architecture

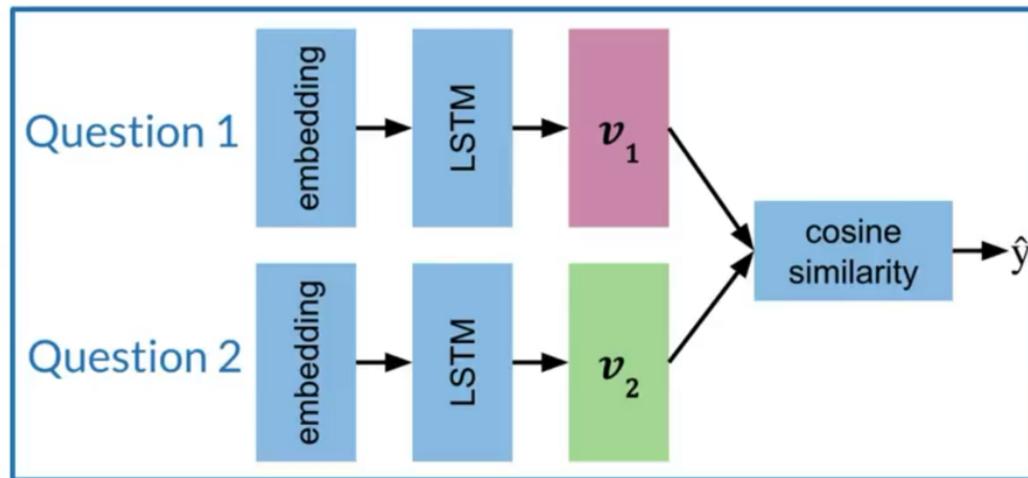
Model Architecture



- 1) Inputs
- 2) Embedding
- 3) LSTM
- 4) Vectors
- 5) Cosine Similarity

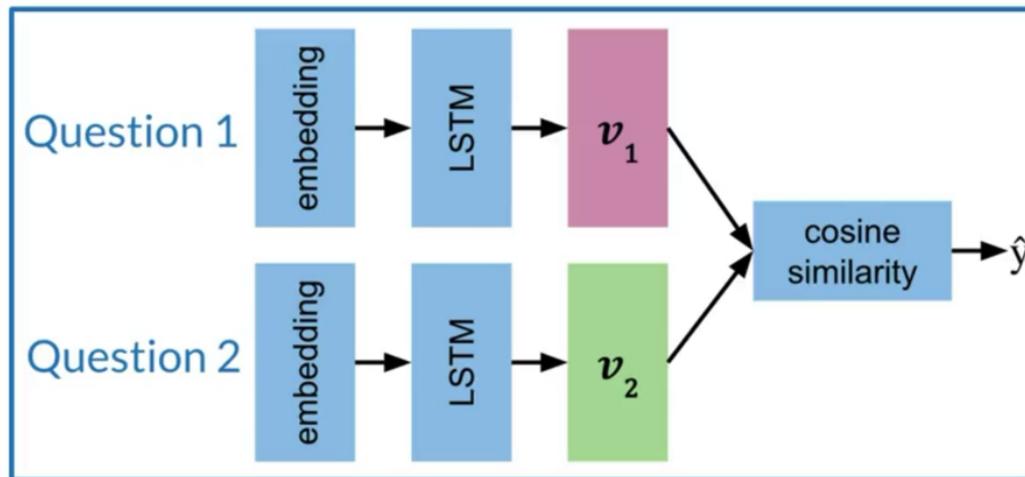
Cost Function

Loss Function



Cost Function

Loss Function



$$\hat{y} = s(v_1, v_2)$$

Cost Function

Loss Function

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$

How old are you?

Anchor

$s(v_1, v_2)$

What is your age?

Positive

$s(A, P)$

Where are you from?

Negative

Cost Function

Loss Function

How old are you?

Anchor

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$
$$s(v_1, v_2)$$

What is your age?

Positive

$$s(A, P)$$

Where are you from? Negative

Cost Function

Loss Function

How old are you?

Anchor

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$
$$s(v_1, v_2)$$

What is your age?

Positive

$$s(A, P) \approx 1$$

Where are you from? Negative

Cost Function

Loss Function

How old are you?

Anchor

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$
$$s(v_1, v_2)$$

What is your age?

Positive

$$s(A, P) \approx 1$$

Where are you from?

Negative

$$s(A, N) \approx -1$$

$$s(A, N) - s(A, P)$$

Cost Function

Loss Function

How old are you?

Anchor

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$
$$s(v_1, v_2)$$

What is your age?

Positive

$$s(A, P) \approx 1$$

Where are you from?

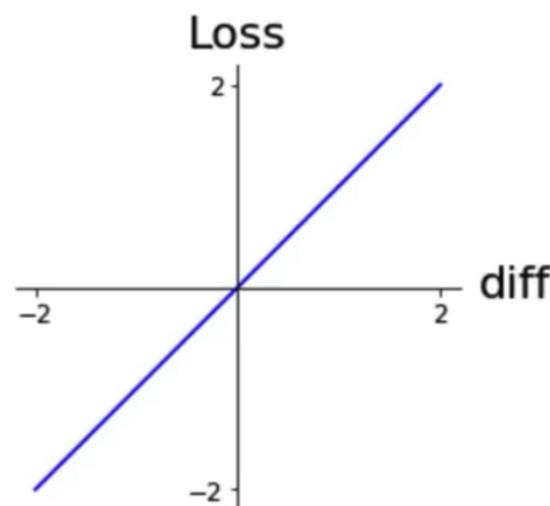
Negative

$$s(A, N) \approx -1$$

$$s(A, N) - s(A, P)$$

Cost Function

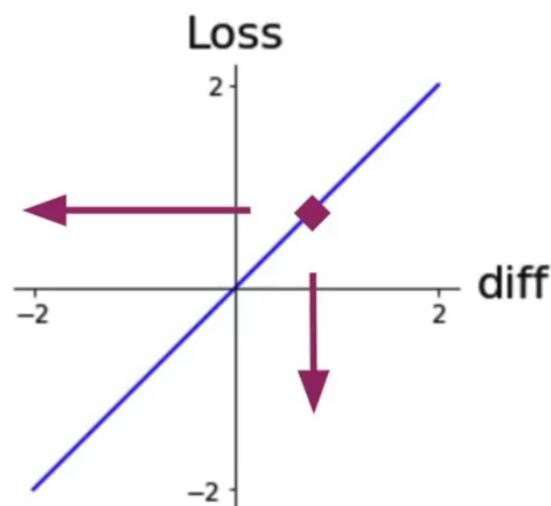
Loss Function



$$\text{diff} = s(A, N) - s(A, P)$$

Cost Function

Loss Function



$$\text{diff} = s(A, N) - s(A, P)$$

X

Triplets

Triplets

How old are you?

What is your age?

Where are you from?

A

P

N



Triplets !!!



X

Triplets

Triplet Loss

Simple loss:

$$\text{diff} = s(A, N) - s(A, P)$$

How old are you? A

What is your age? P

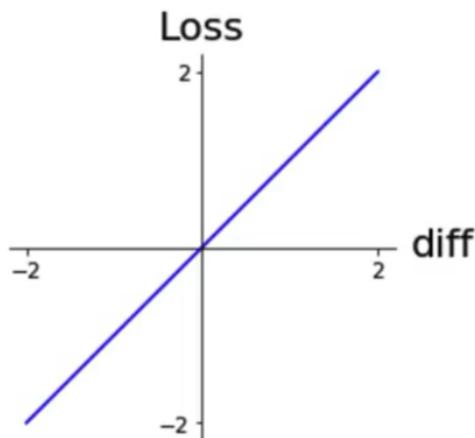
Where are you from? N

X Triplets

Triplet Loss

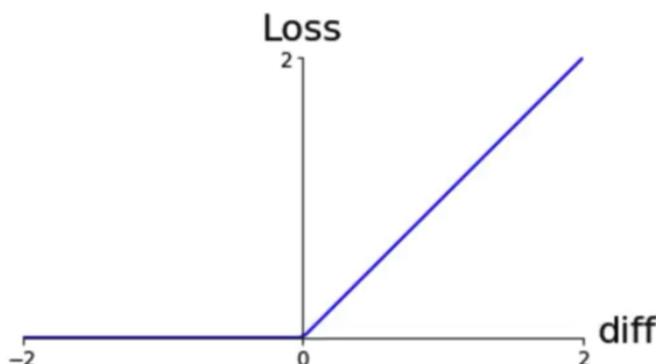
Simple loss:

$$\text{diff} = s(A, N) - s(A, P)$$



X Triplets

Triplet Loss



Simple loss:

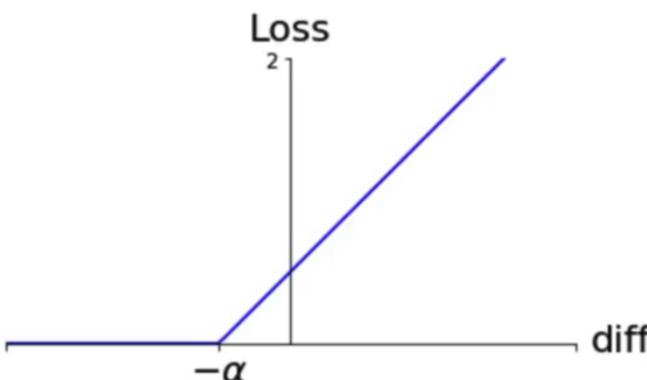
$$\text{diff} = s(A, N) - s(A, P)$$

Non linearity:

$$\mathcal{L} = \begin{cases} 0; & \text{if } \text{diff} \leq 0 \\ \text{diff}; & \text{if } \text{diff} > 0 \end{cases}$$

X Triplet

Triplet Loss



Simple loss:

$$\text{diff} = s(A, N) - s(A, P)$$

Non linearity:

$$\mathcal{L} = \begin{cases} 0; & \text{if } \text{diff} \leq 0 \\ \text{diff}; & \text{if } \text{diff} > 0 \end{cases}$$

Alpha margin:

$$\mathcal{L} = \begin{cases} 0; & \text{if } \text{diff} + \alpha \leq 0 \\ \text{diff} + \alpha; & \text{if } \text{diff} + \alpha > 0 \end{cases}$$

X

Triplets

Triplet Loss

$$\mathcal{L} = \begin{cases} 0; & \text{if } diff + \alpha \leq 0 \\ diff; & \text{if } diff + \alpha > 0 \end{cases}$$

X

Triplets

Triplet Loss

$$s(v_1, v_2)$$

$$\mathcal{L} = \begin{cases} 0; & \text{if } diff + \alpha \leq 0 \\ diff; & \text{if } diff + \alpha > 0 \end{cases}$$

$$\mathcal{L}(A, P, N) = \max (diff + \alpha, 0)$$

X

Triplets

Triplet Loss

s(v_1, v_2) d(v_1, v_2) 

$$\mathcal{L} = \begin{cases} 0; & \text{if } diff + \alpha \leq 0 \\ diff; & \text{if } diff + \alpha > 0 \end{cases}$$

$$\mathcal{L}(A, P, N) = \max (diff + \alpha, 0)$$

Triplet Selection

Hard triplets are better for training !

Triplet A, P, N:



duplicate set: A, P
non-duplicate set: A, N

Random:

$$\mathcal{L} = \max (diff + \alpha, 0)$$

$$diff = s(A, N) - s(A, P)$$

Easy to satisfy. Little to learn



Triplet Selection

Hard triplets are better for training !

Triplet A, P, N:



duplicate set: A, P
non-duplicate set: A, N

Random:

$$\mathcal{L} = \max(\text{diff} + \alpha, 0)$$

$$\text{diff} = s(A, N) - s(A, P)$$

Easy to satisfy. Little to learn



Hard:

$$s(A, N) \approx s(A, P)$$

Harder to train. More to learn



X

Computing The Cost

Prepare the batches as follows:

What is your age?

How old are you?

Can you see me?

Are you seeing me?

Where are thou?

Where are you?

When is the game?

What time is the game?

X

Computing The Cost

Prepare the batches as follows:

What is your age?

How old are you?

Can you see me?

Are you seeing me?

Where are thou?

Where are you?

When is the game?

What time is the game?

$$b = 4$$

Computing The Cost

Prepare the batches as follows:

What is your age?

How old are you?



Can you see me?

Are you seeing me?

Where are thou?

Where are you?

When is the game?

What time is the game?

$$b = 4$$



Computing The Cost

Prepare the batches as follows:



What is your age?

Can you see me?

Where are thou?

When is the game?

How old are you?

Are you seeing me?

Where are you?

What time is the game?

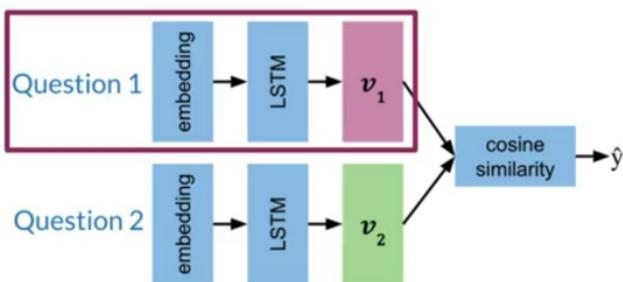


$$b = 4$$

X

Computing The Cost I

Computing The Cost



Batch 1
What is your age?
Can you see me?
Where are thou?
When is the game?

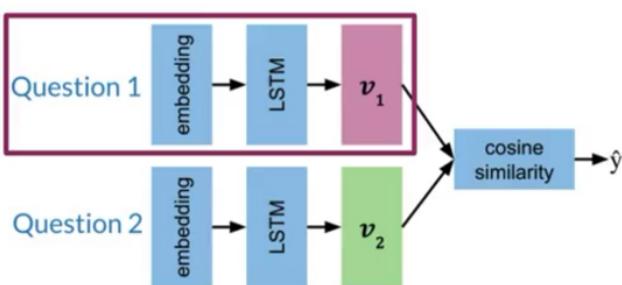
$$\mathbf{v}_1 = (1, d_{\text{model}})$$

--	--	--	--	--

X

Computing The Cost I

Computing The Cost



Batch 1

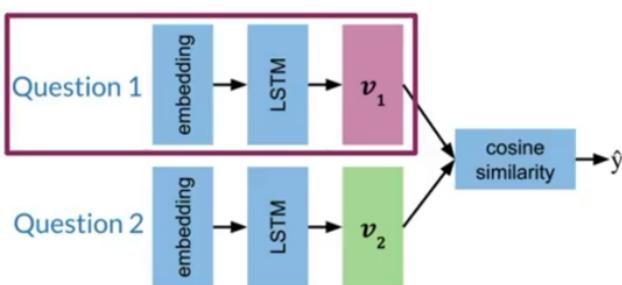
What is your age?
Can you see me?
Where are thou?
When is the game?

$$\mathbf{v}_1 = (1, d_{\text{model}})$$

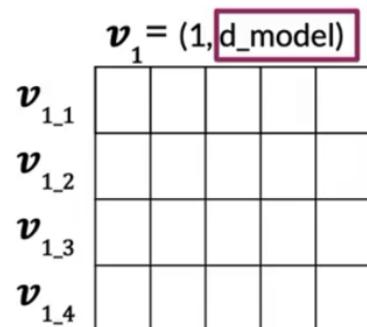
X

Computing The Cost I

Computing The Cost

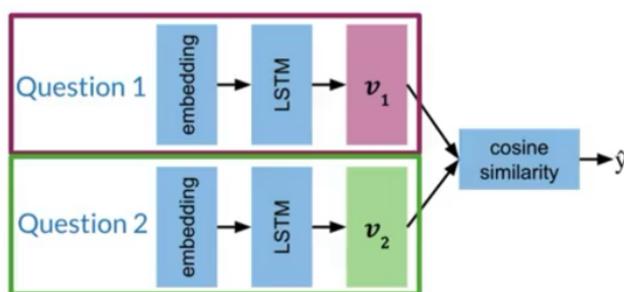


Batch 1
What is your age?
Can you see me?
Where are thou?
When is the game?

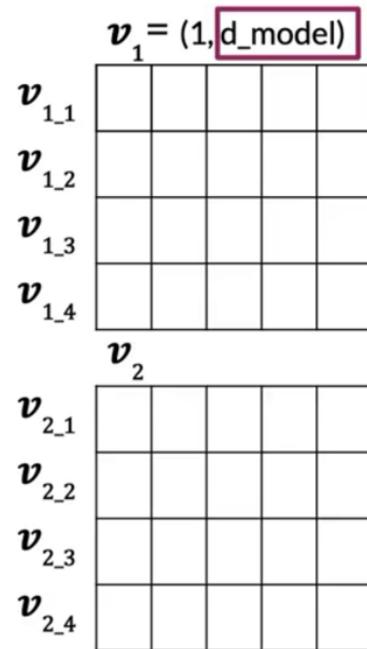




Computing The Cost



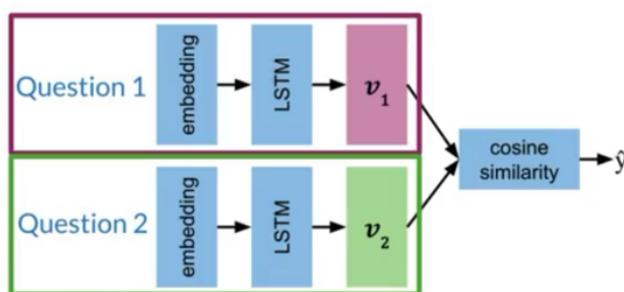
- Batch 1
- What is your age?
Can you see me?
Where are thou?
When is the game?
- Batch 2
- How old are you?
Are you seeing me?
Where are you?
What time is the game?



X

Computing The Cost I

Computing The Cost

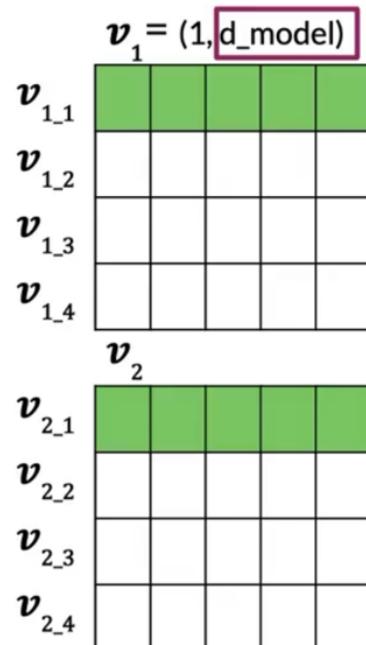


Batch 1

- What is your age?
- Can you see me?
- Where are thou?
- When is the game?

Batch 2

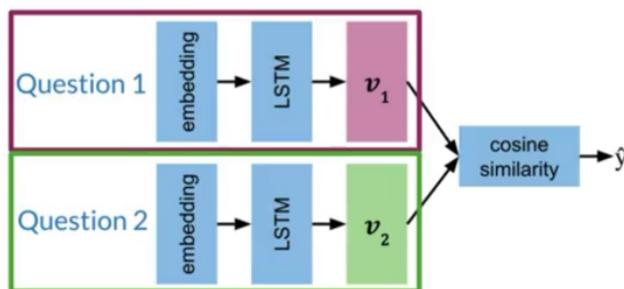
- How old are you?
- Are you seeing me?
- Where are you?
- What time is the game?



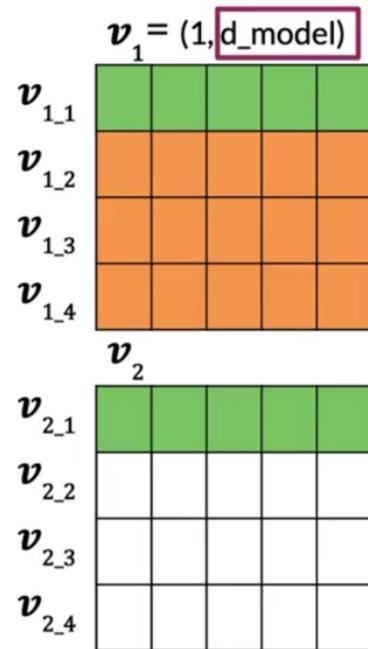
X

Computing The Cost I

Computing The Cost



- Batch 1
- What is your age?
Can you see me?
Where are thou?
When is the game?
- Batch 2
- How old are you?
Are you seeing me?
Where are you?
What time is the game?



Computing The Cost

$s(v_1, v_2)$

		v_1				
		-1	-2	-3	-4	
		-1	0.9	-0.8	0.3	-0.5
		-2	-0.8	0.5	0.1	-0.2
		-3	0.3	0.1	0.7	-0.8
		-4	-0.5	-0.2	-0.8	1.0

X Computing The Cost I

Computing The Cost

 $s(v_1, v_2)$

		v_1				
		-1	-2	-3	-4	
		-1	0.9	-0.8	0.3	-0.5
		-2	-0.8	0.5	0.1	-0.2
		-3	0.3	0.1	0.7	-0.8
		-4	-0.5	-0.2	-0.8	1.0

X

Computing The Cost I

Computing The Cost

 $s(v_1, v_2)$ v_1 $-1 \quad -2 \quad -3 \quad -4$

v_2	-1	-2	-3	-4
-1	0.9	-0.8	0.3	-0.5
-2	-0.8	0.5	0.1	-0.2
-3	0.3	0.1	0.7	-0.8
-4	-0.5	-0.2	-0.8	1.0

$$\mathcal{L}(A, P, N) = \max (diff + \alpha, 0)$$

$$diff = s(A, N) - s(A, P)$$

X

Computing The Cost I

Computing The Cost

$s(v_1, v_2)$

		v_1			
		-1	-2	-3	-4
v_2		0.9	-0.8	0.3	-0.5
-1		-0.8	0.5	0.1	-0.2
-2		0.3	0.1	0.7	-0.8
-3		-0.5	-0.2	-0.8	1.0

$$\mathcal{L}(A, P, N) = \max (diff + \alpha, 0)$$
$$diff = s(A, N) - s(A, P)$$

$$\mathcal{J} = \sum_{i=1}^m \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

X

Computing The Cost II

Computing The Cost

Batch 1

What is your age?

Can you see me?

Where are thou?

When is the game?

Batch 2

How old are you?

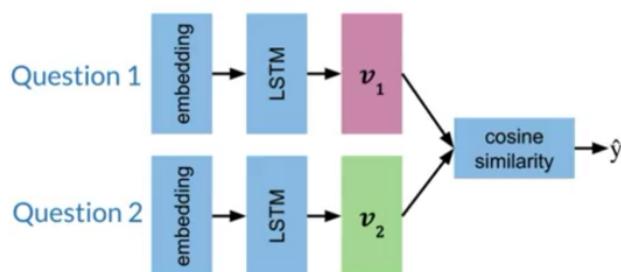
Are you seeing me?

Where are you?

What time is the game?

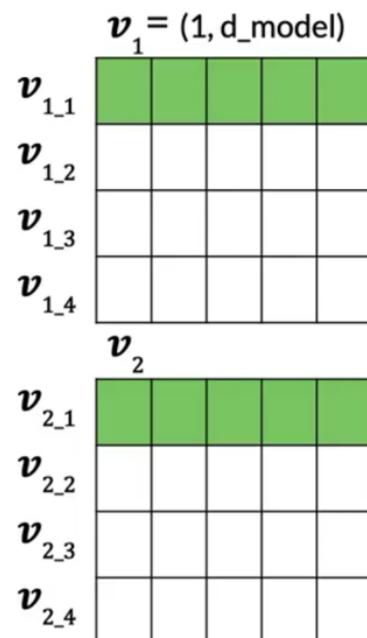
X Computing The Cost II

Computing The Cost



Batch 1
What is your age?
Can you see me?
Where are thou?
When is the game?

Batch 2
How old are you?
Are you seeing me?
Where are you?
What time is the game?



X

Hard Negative Mining

$s(v_1, v_2)$

		v_1			
		-1	-2	-3	-4
v_2		-1	-0.8	0.3	-0.5
-2		-0.8	0.5	0.1	-0.2
-3		0.3	0.1	0.7	-0.8
-4		-0.5	-0.2	-0.8	1.0

mean negative:

mean of off-diagonal values in each row

X

Hard Negative Mining

mean negative:

mean of off-diagonal values in each row

$$s(v_1, v_2)$$

$$v_1$$

$$\begin{matrix} -1 & -2 & -3 & -4 \end{matrix}$$

v_2	-1	0.9	-0.8	0.3	-0.5
	-2	-0.8	0.5	0.1	-0.2
	-3	0.3	0.1	0.7	-0.8
	-4	-0.5	-0.2	-0.8	1.0

X

Hard Negative Mining

$$s(v_1, v_2)$$

$$v_1$$

_1 _2 _3 _4

v_2	_1	0.9	-0.8	0.3	-0.5
	_2	-0.8	0.5	0.1	-0.2
	_3	0.3	0.1	0.7	-0.8
	_4	-0.5	-0.2	-0.8	1.0

mean negative:

mean of off-diagonal values in each row

closest negative:

off-diagonal value closest to (but less than) the value on diagonal in each row

Hard Negative Mining

$$\mathcal{L}_{\text{Original}} = \max (\underbrace{s(A, N) - s(A, P)}_{\text{diff}} + \alpha, 0)$$

Hard Negative Mining

$$\mathcal{L}_{\text{Original}} = \max (\underbrace{s(A, N) - s(A, P) + \alpha}_{\text{diff}}, 0)$$



Hard Negative Mining

$$\mathcal{L}_{\text{Original}} = \max (\underbrace{s(A, N) - s(A, P) + \alpha}_{\text{diff}}, 0)$$

$$\mathcal{L}_1 = \max (mean_neg - s(A, P) + \alpha, 0)$$



Hard Negative Mining

mean negative: mean of off-diagonal values

$$\mathcal{L}_{\text{Original}} = \max (\underbrace{s(A, N) - s(A, P)}_{\text{diff}} + \alpha, 0)$$

$$\mathcal{L}_1 = \max (\text{mean_neg} - s(A, P) + \alpha, 0)$$



Hard Negative Mining

mean negative: mean of off-diagonal values

$$\mathcal{L}_{\text{Original}} = \max (\underbrace{s(A, N) - s(A, P)}_{\text{diff}} + \alpha, 0)$$

$$\mathcal{L}_1 = \max (\text{mean_neg} - s(A, P) + \alpha, 0)$$

$$\mathcal{L}_2 = \max (\text{closest_neg} - s(A, P) + \alpha, 0)$$



Hard Negative Mining

mean negative: mean of off-diagonal values

closest negative: closest off-diagonal value

$$\mathcal{L}_{\text{Original}} = \max (\underbrace{s(A, N) - s(A, P)}_{\text{diff}} + \alpha, 0)$$

$$\mathcal{L}_1 = \max (\text{mean_neg} - s(A, P) + \alpha, 0)$$

$$\mathcal{L}_2 = \max (\text{closest_neg} - s(A, P) + \alpha, 0)$$



Hard Negative Mining

mean negative: mean of off-diagonal values

closest negative: closest off-diagonal value

$$\mathcal{L}_{\text{Original}} = \max (\underbrace{s(A, N) - s(A, P)}_{\text{diff}} + \alpha, 0)$$

$$\mathcal{L}_1 = \max (\text{mean_neg} - s(A, P) + \alpha, 0)$$

$$\mathcal{L}_2 = \max (\text{closest_neg} - s(A, P) + \alpha, 0)$$

$$\mathcal{L}_{\text{Full}} = \mathcal{L}_1 + \mathcal{L}_2$$



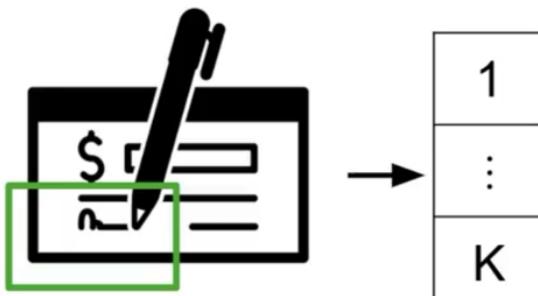
Hard Negative Mining

$$\mathcal{L}_{\text{Full}}(A, P, N) = \mathcal{L}_1 + \mathcal{L}_2$$

$$\mathcal{J} = \sum_{i=1}^m \mathcal{L}_{\text{Full}}(A^{(i)}, P^{(i)}, N^{(i)})$$

Classification vs One Shot Learning

Classification



Classify as 1 of K classes

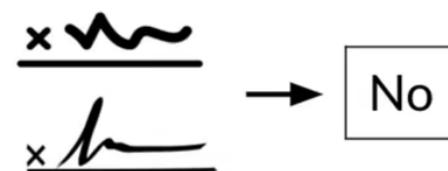
Classification vs One Shot Learning

Classification



Classify as 1 of K classes

One Shot Learning



Measure similarity between
2 classes

One Shot Learning

No need for retraining !



Learn a similarity score!

One Shot Learning

No need for retraining !



Learn a similarity score!

$$s(sig1, sig2) > \tau$$

One Shot Learning

No need for retraining !



Learn a similarity score!

$s(sig1, sig2) > \tau$

$s(sig1, sig2) \leq \tau$

One Shot Learning

No need for retraining !



Learn a similarity score!

$s(sig1, sig2) > \tau$

$s(sig1, sig2) \leq \tau$

X

Training / Testing

Dataset

Question 1	Question 2	is_duplicate
What is your age?	How old are you?	true
Where are you from?	Where are you going?	false
:	:	:



Prepare Batches

Question 1:
batch size b

Batch 1

What is your age?
Can you see me?
Where are thou?
When is the game?

q1_a
q1_b

Question 2:
batch size b

Batch 2

How old are you? q2_a
Are you seeing me? q2_b
Where are you?
What time is the game?



Prepare Batches

Question 1:
batch size b

Batch 1

{ What is your age? q1_a
Can you see me? q1_b
Where are thou?
When is the game?

Question 2:
batch size b

Batch 2

{ How old are you?
Are you seeing me?
Where are you?
What time is the game?

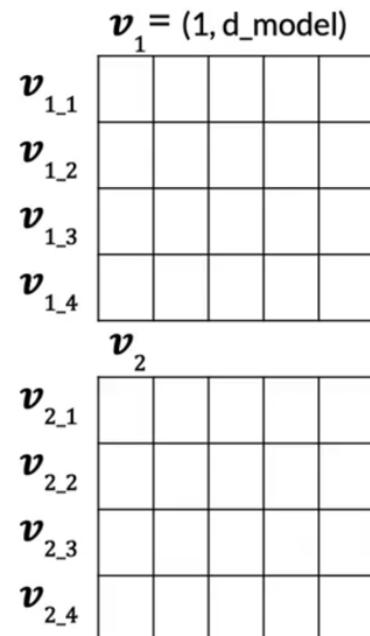
q2_a
q2_b

Prepare Batches

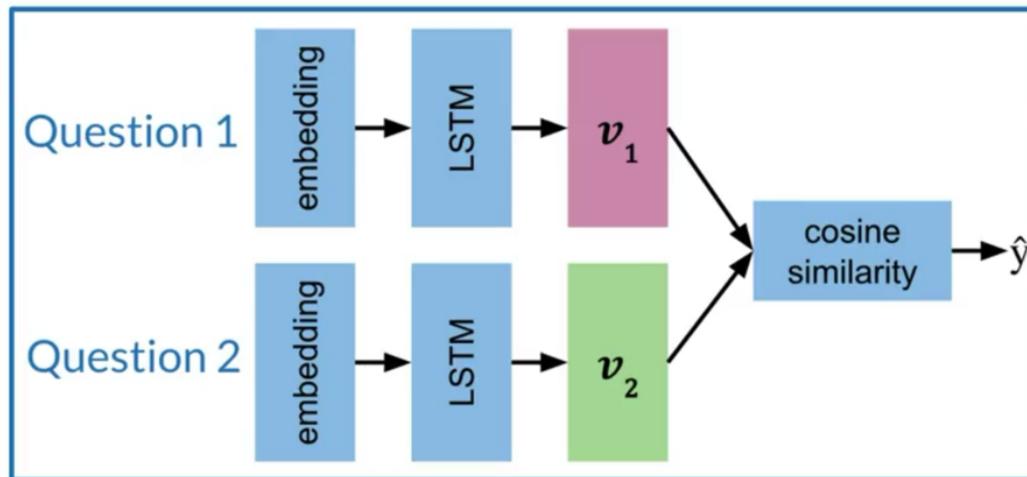
Question 1:
batch size b

Question 2:
batch size b

- {
 - Batch 1
 - What is your age?
 - Can you see me?
 - Where are thou?
 - When is the game?
 - Batch 2
 - How old are you?
 - Are you seeing me?
 - Where are you?
 - What time is the game?



Siamese Model



Create a subnetwork:

- 1) Embedding
- 2) LSTM
- 3) Vectors
- 4) Cosine Similarity

X

Testing

1. Convert each input into an array of numbers
2. Feed arrays into your model
3. Compare v_1, v_2 using cosine similarity

X

Testing

1. Convert each input into an array of numbers
2. Feed arrays into your model
3. Compare v_1, v_2 using cosine similarity
4. Test against a threshold τ