Copyright Notice

deeplearning.ai

Siamese Networks

Question Duplicates

How old are you? = What is your age?

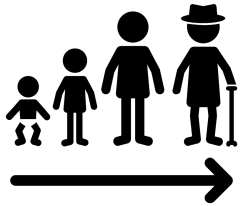
Where are you from? ≠ Where are you going?

What do Siamese Networks learn?

I am happy because I am learning

Classification: categorize things

Siamese Networks: Identify similarity between things

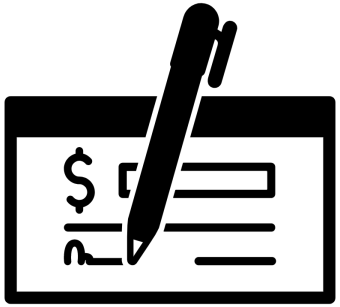


What is your age?

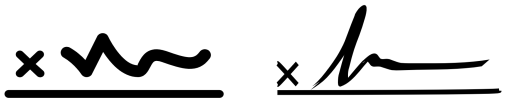
How old are you?

Difference or Similarity

Siamese Networks in NLP



Handwritten checks



What is your age?

How old are you?

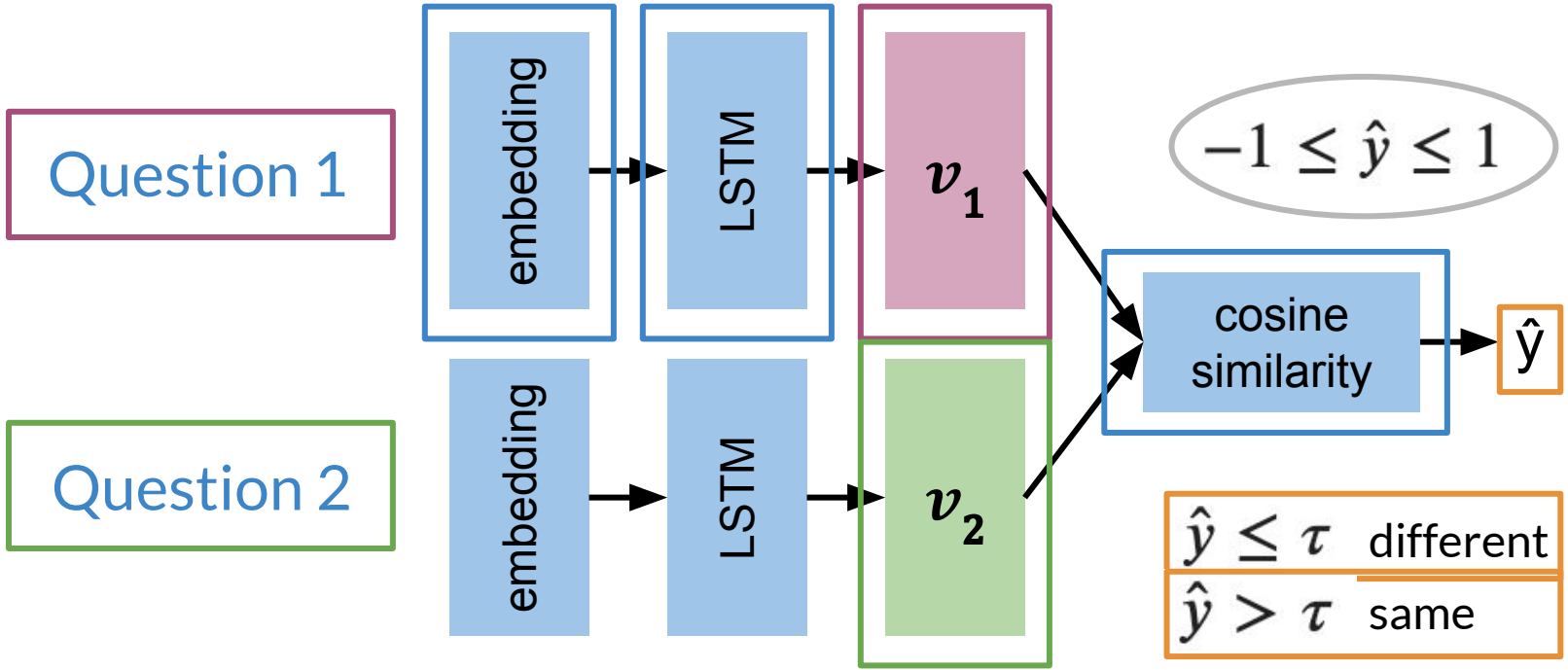
Question duplicates

Queries

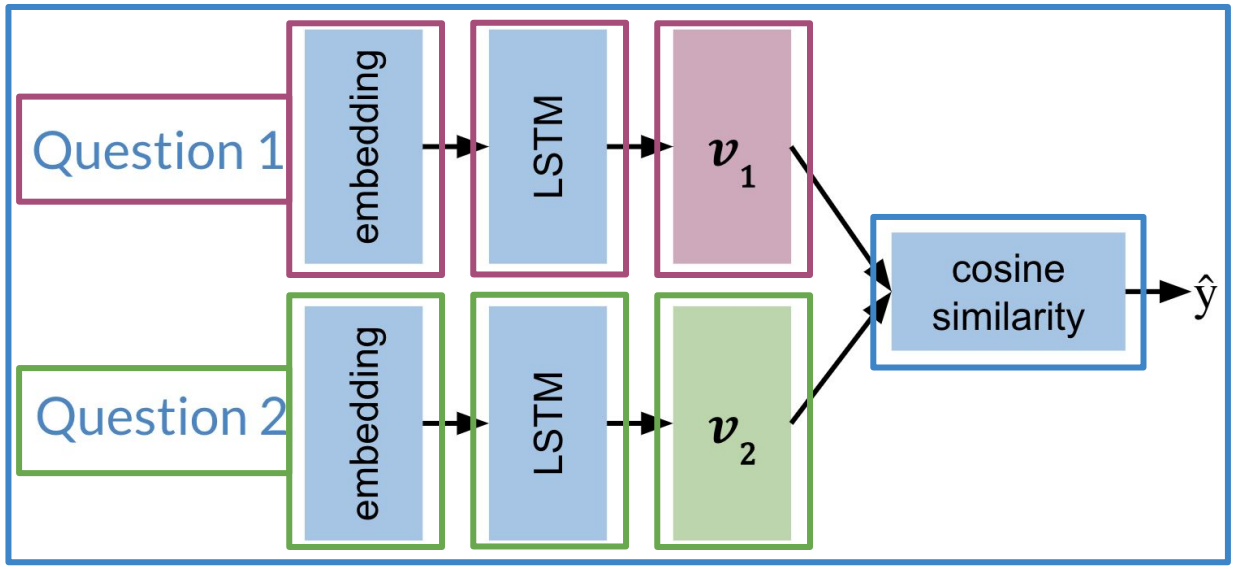
deeplearning.ai

Architecture

Model Architecture



Model Architecture

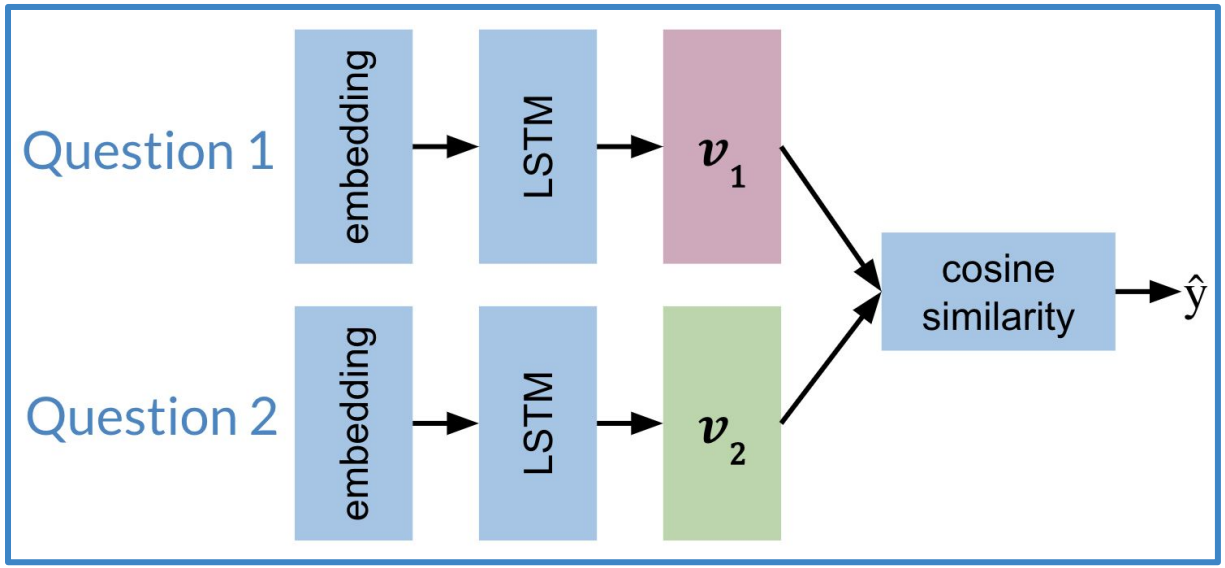


1) Inputs
2) Embedding
3) LSTM
4) Vectors
5) Cosine Similarity

deeplearning.ai

Loss Function

Loss Function



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

Loss Function

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

$$s(v_1, v_2)$$

How old are you? — Anchor

What is your age? — Positive

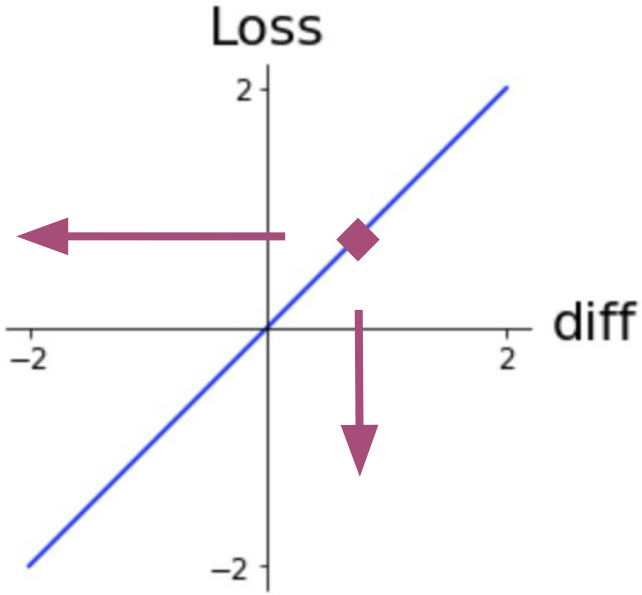
Where are you from? — Negative

$s(A, P)$

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

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

Loss Function



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

deeplearning.ai

Triplets

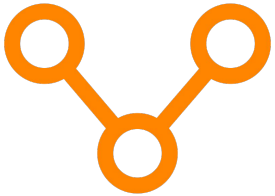
Triplets

How old are you? — Anchor

What is your age? — Positive

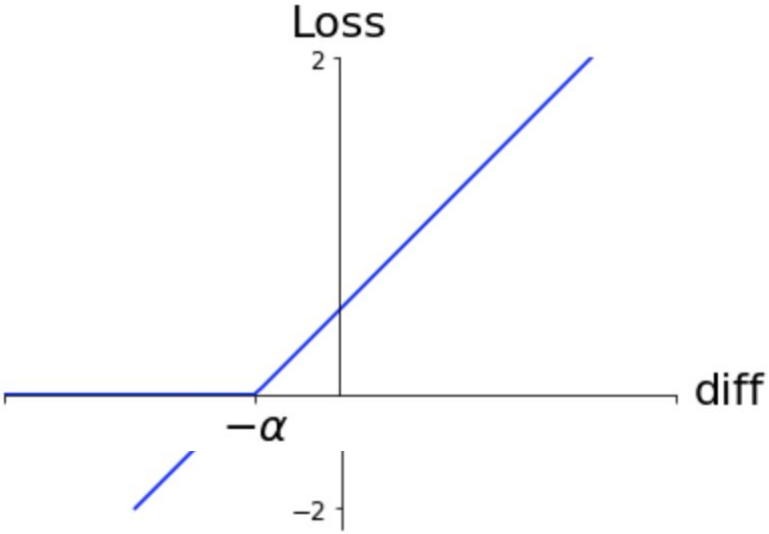
Where are you from? — Negative

} Triplets



Whether or not a question has the same meaning as the anchor

Triplet Loss



Simple loss:

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

Anchor With non-linearity

Positi

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

Negative

With alpha margin

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

Triplet Loss

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

Simplified

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

From the neural network

You can use any similarity function or distance metric

Triplet Selection

Triplet A, P, N

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

Random

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

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

Easy to satisfy. Little to learn

Hard

$$\text{s}(A, N) \approx \text{s}(A, P)$$

Harder to train. More to learn

deeplearning.ai

Computing The Cost I

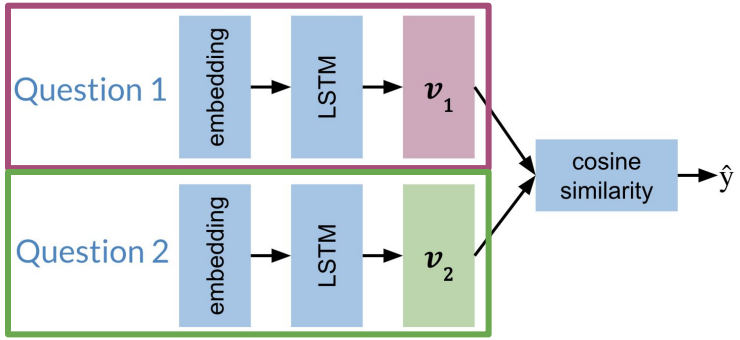
Computing The Cost

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



Batch 1

What is your age?

Can you see me?

Where are thou?

When is the game?

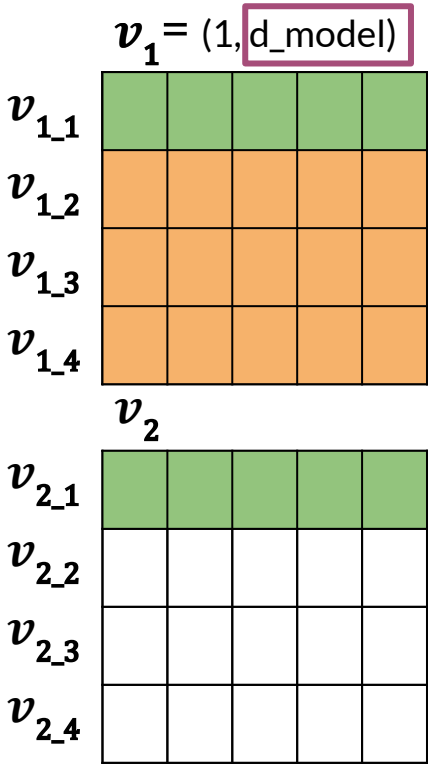
Batch 2

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_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

Computing The Cost

$s(v_1, v_2)$

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

Computing The Cost

$s(v_1, v_2)$

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

Computing The Cost

$s(v_1, v_2)$

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

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

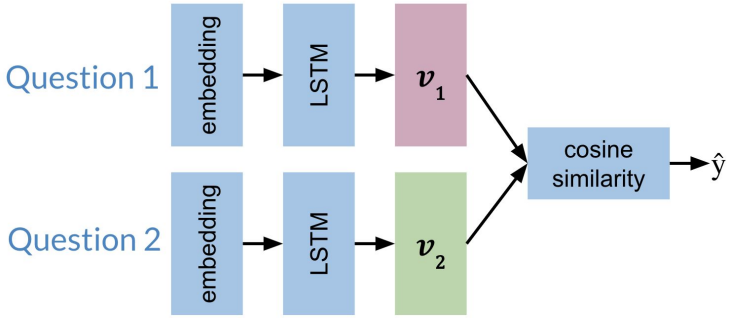
$$\text{diff} = s(A, N) - s(A, P)$$

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

deeplearning.ai

Computing The Cost II

Computing The Cost



Batch 1

What is your age?

Can you see me?

Where are thou?

When is the game?

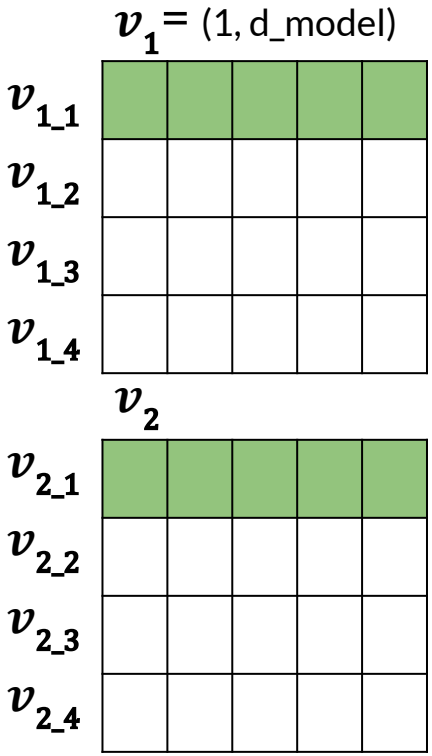
Batch 2

How old are you?

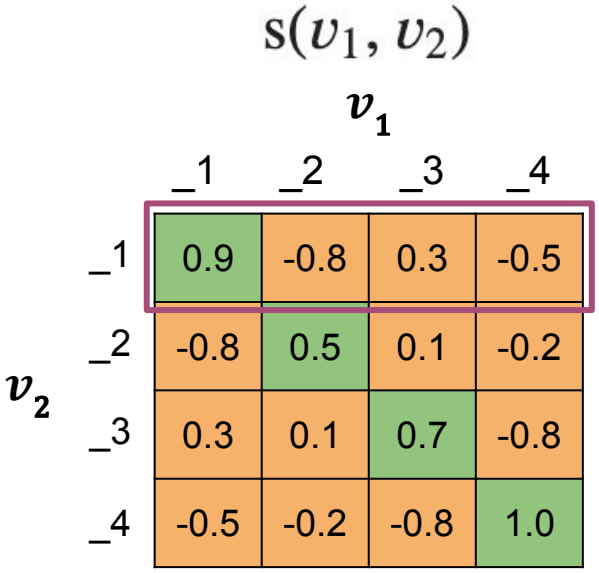
Are you seeing me?

Where are you?

What time is the game?



Hard Negative Mining



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

mean negative: mean of off-diagonal values

closest negative: closest off-diagonal value

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

Hard Negative Mining

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

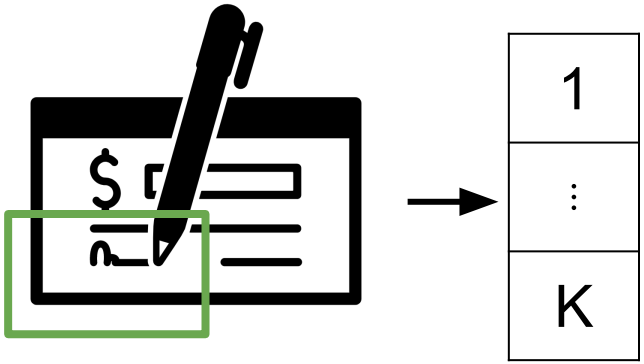
$$\mathcal{J} = \sum_{i=1}^{m} \mathcal{L}_{\text{Full}}(A^{(i)}, P^{(i)}, N^{(i)})$$

deeplearning.ai

One Shot Learning

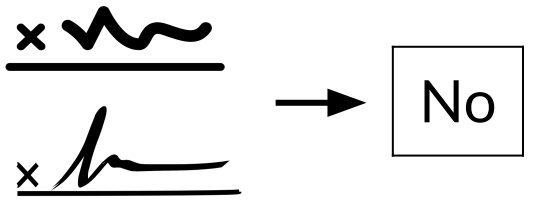
Classification vs One Shot Learning

Classification



Classify as 1 of K classes

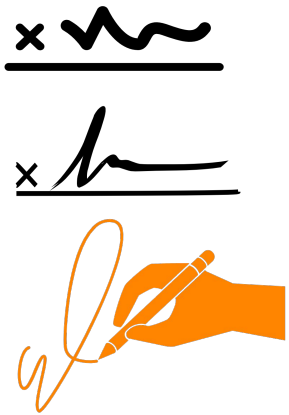
One Shot Learning



Measure similarity between 2 classes

One Shot Learning

No need for retraining !



Learn a similarity score!

$$s(sig1, sig2) > \tau$$ ✅

deeplearning.ai

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

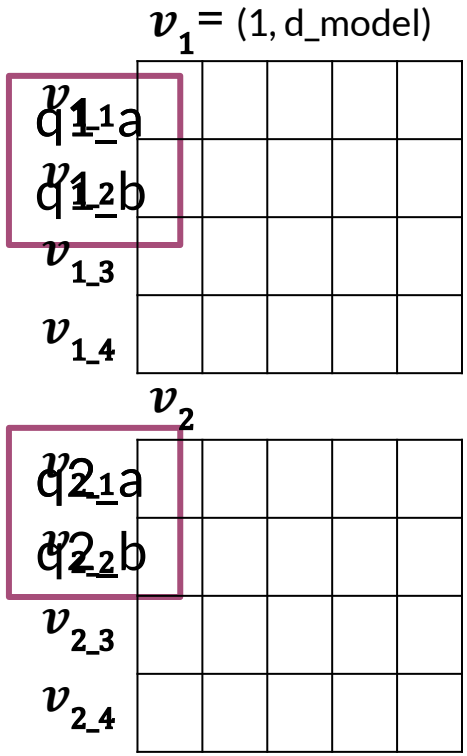
Batch 1

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

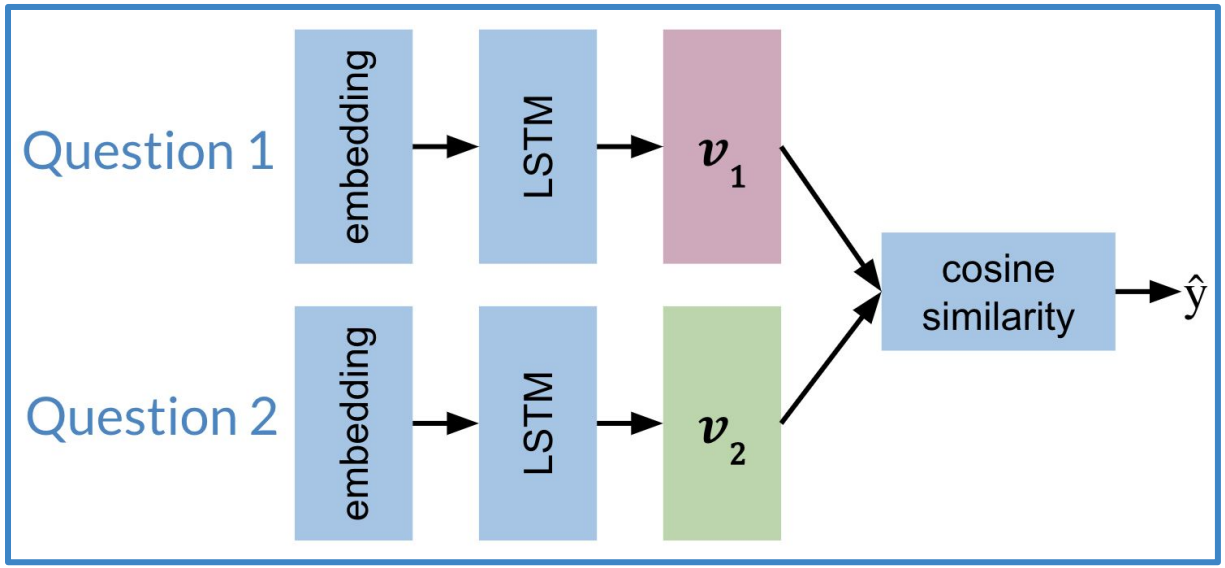
Question 2:
batch size b

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

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