

Transformer (for NLP)

Text-to-Image Creation

Vision Transformer (for CV)

Lecture by Margrit Betke, CS 585, April 16, 2024
with many slides from Steve Seitz' videos:

[Part 1](#) & [Part 2](#)

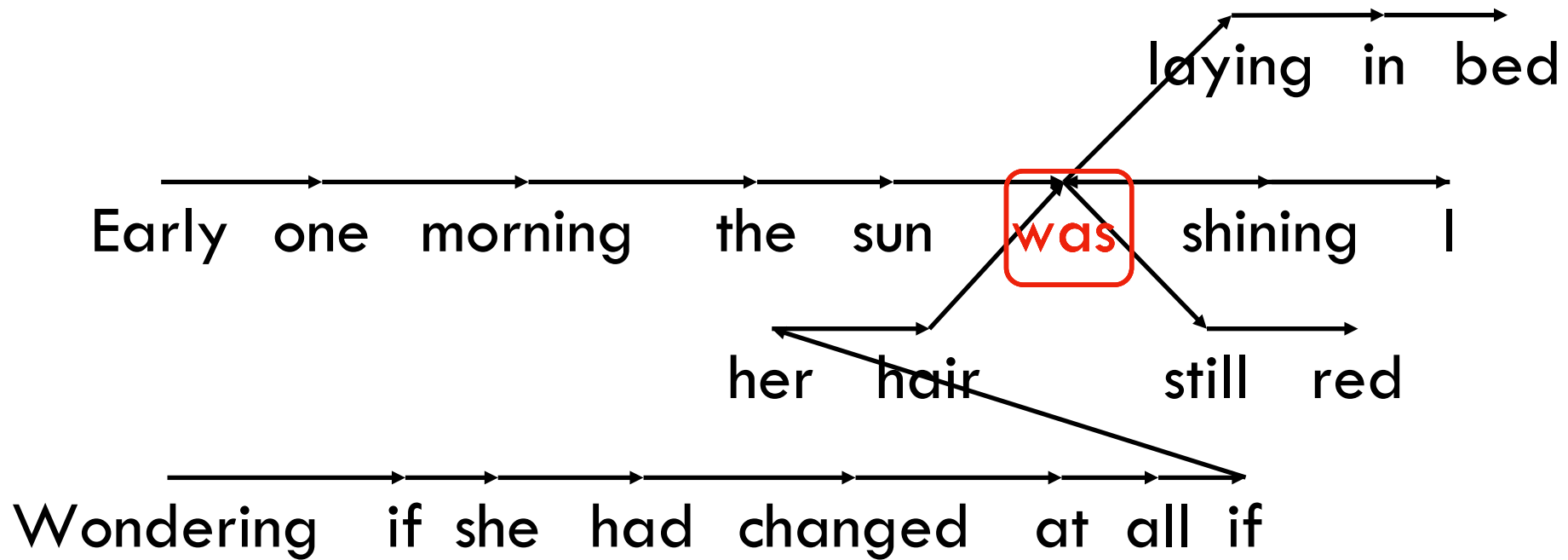
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still red

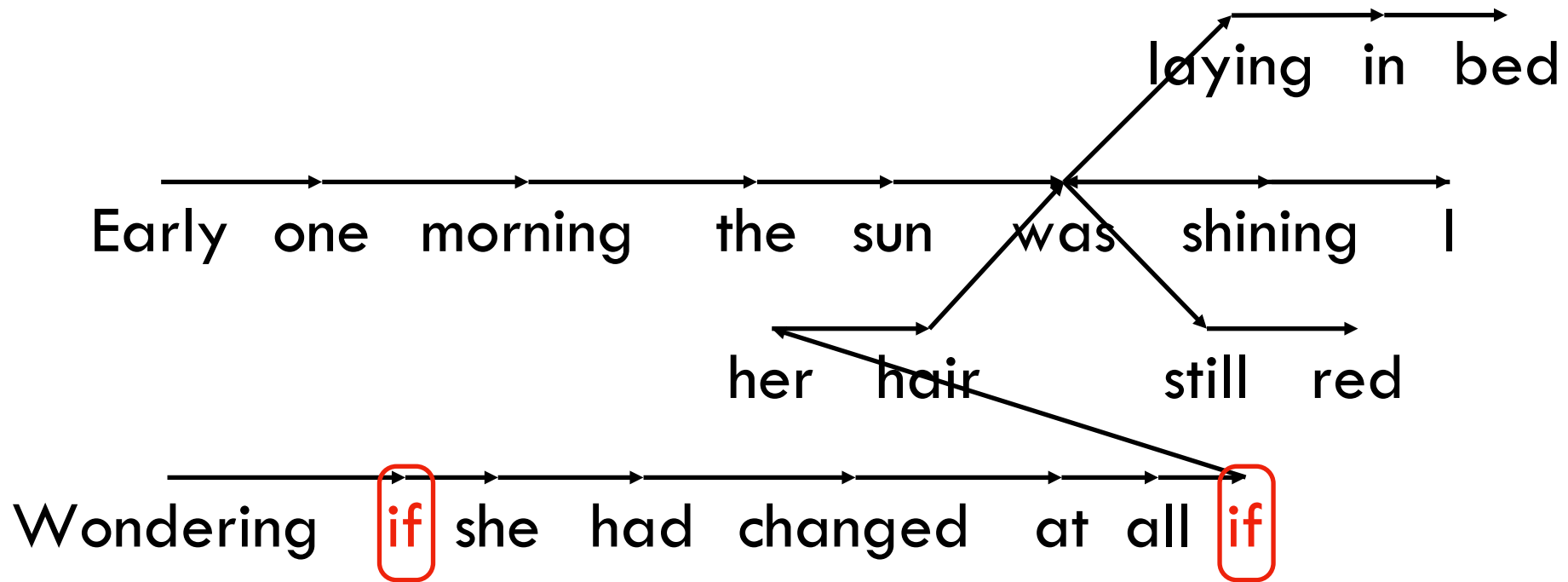
Bob Dylan, *Tangled up in Blue*

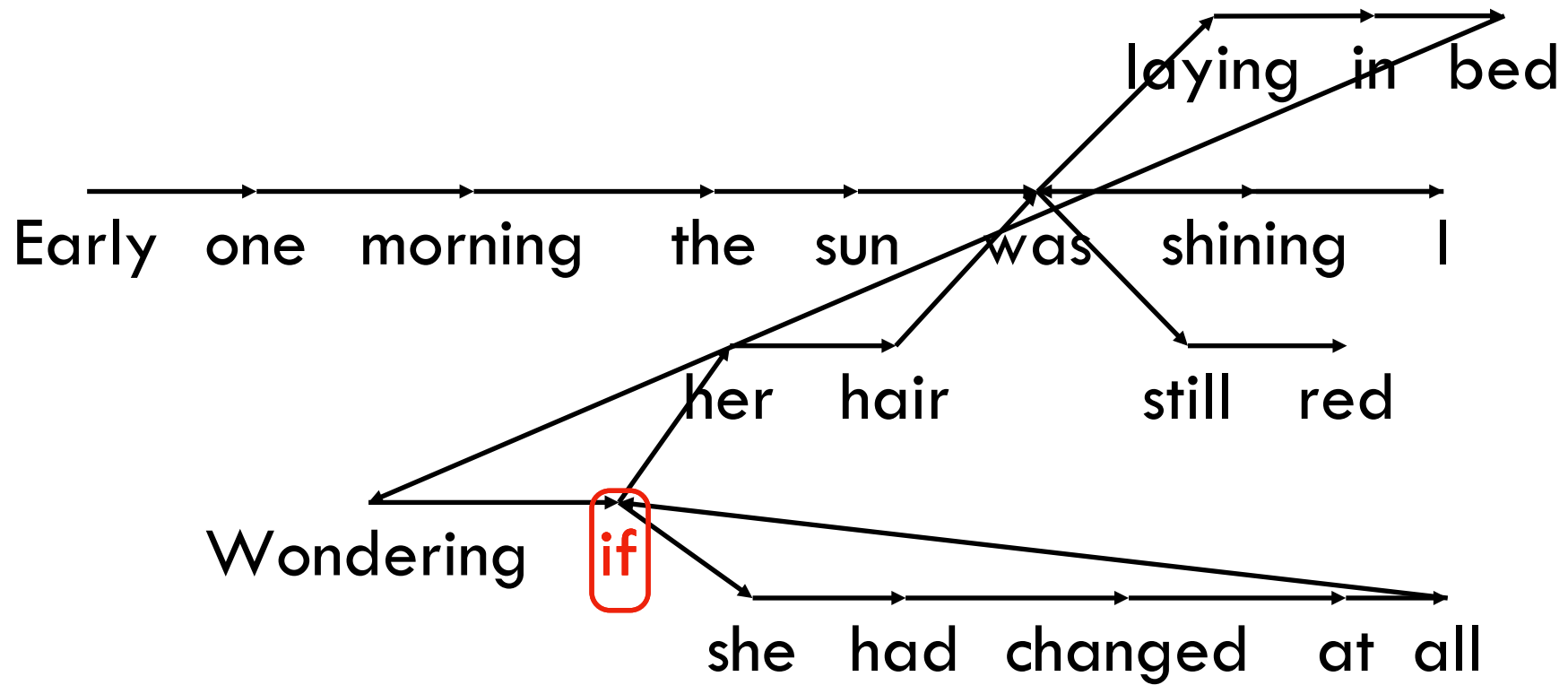
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still red

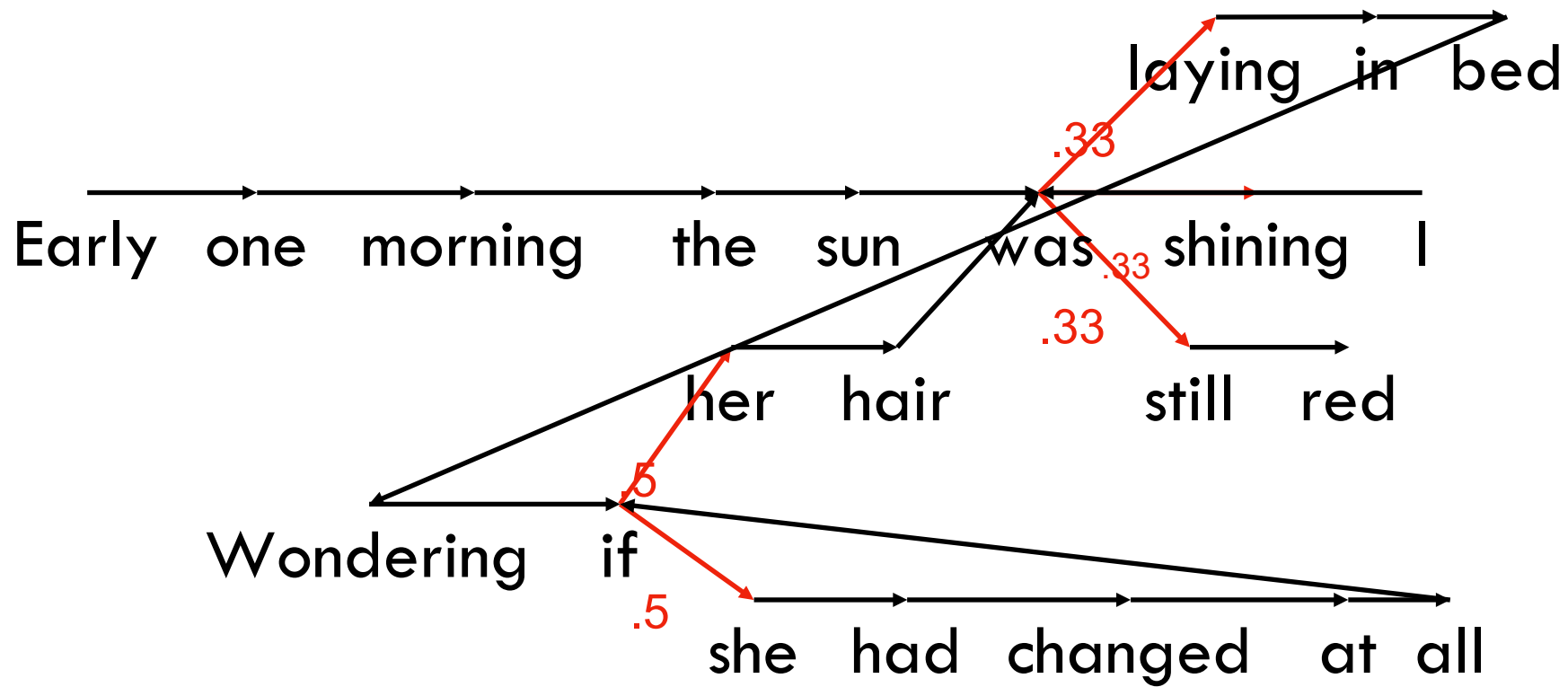
The diagram consists of two horizontal lines of arrows. The top line starts at the beginning of the first sentence and ends at the end of the first sentence. The bottom line starts at the beginning of the second sentence and ends at the end of the second sentence. A single line connects the right end of the top line to the left end of the bottom line, crossing over the space between the two sentences.

Early one morning the sun **was** shining I **was** laying in bed
Wondering if she had changed at all if her hair **was** still red



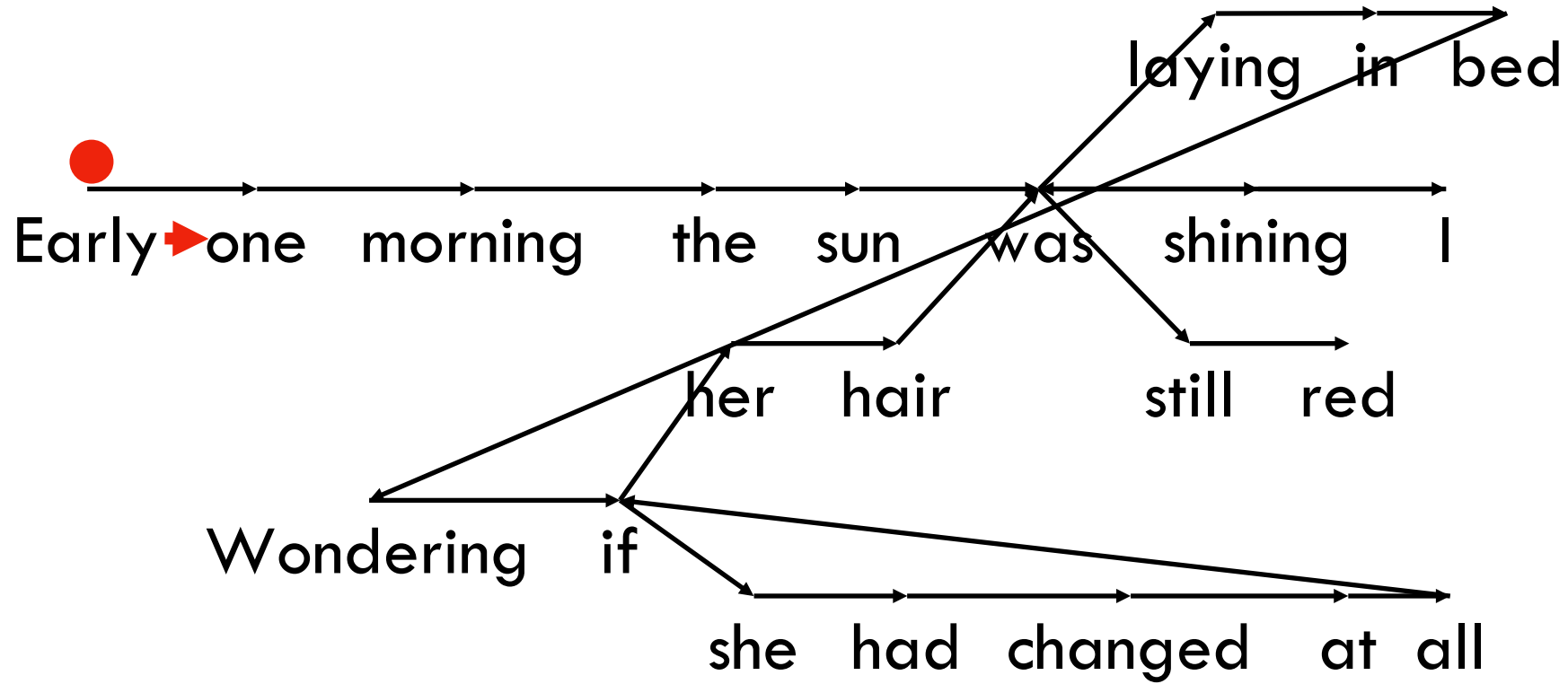




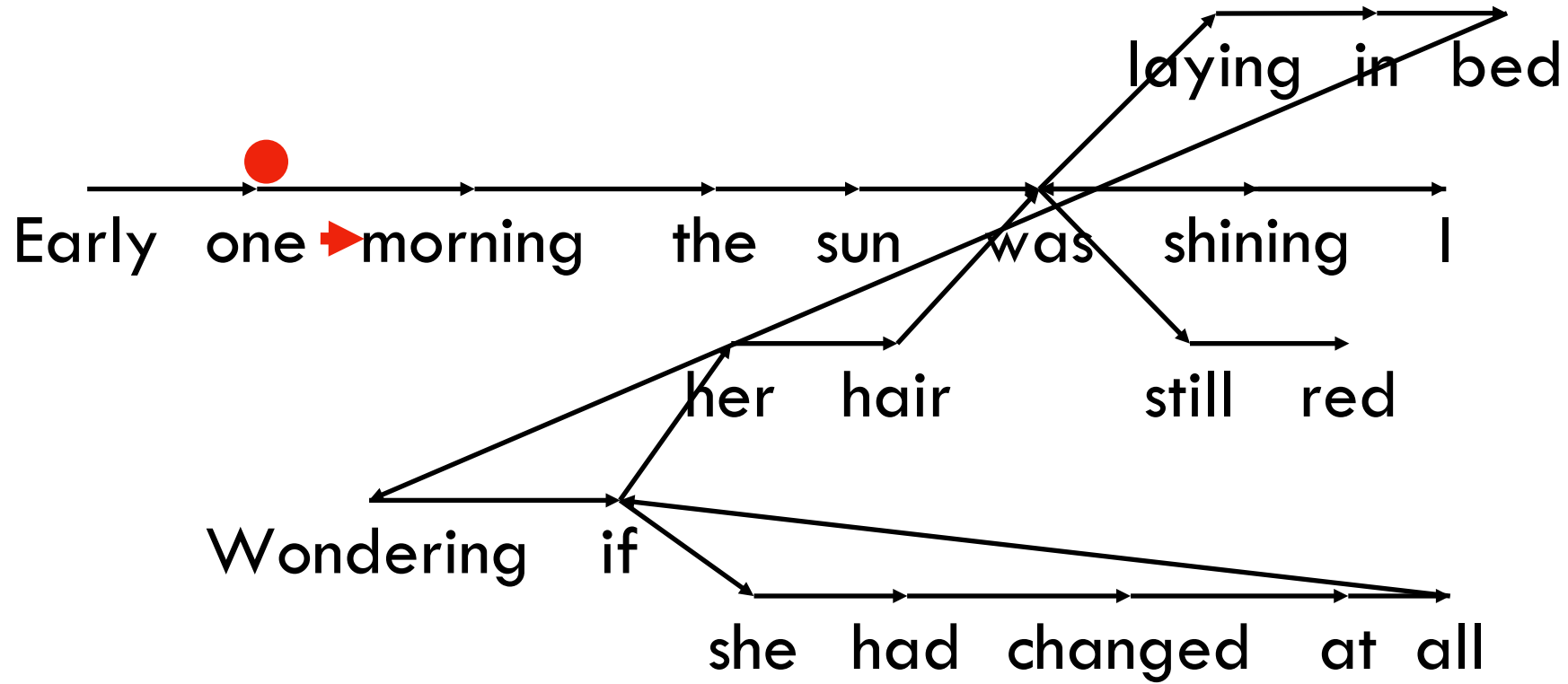


Language Model

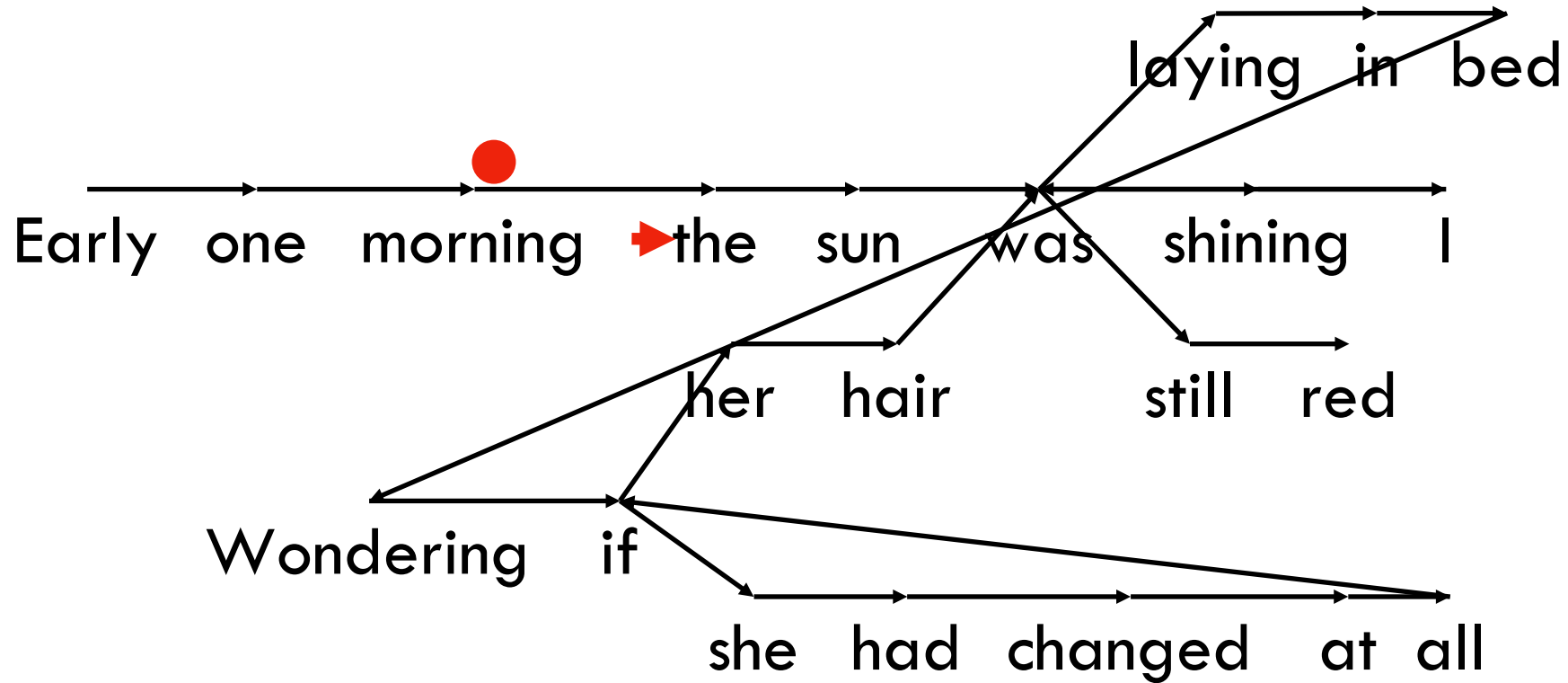
Early



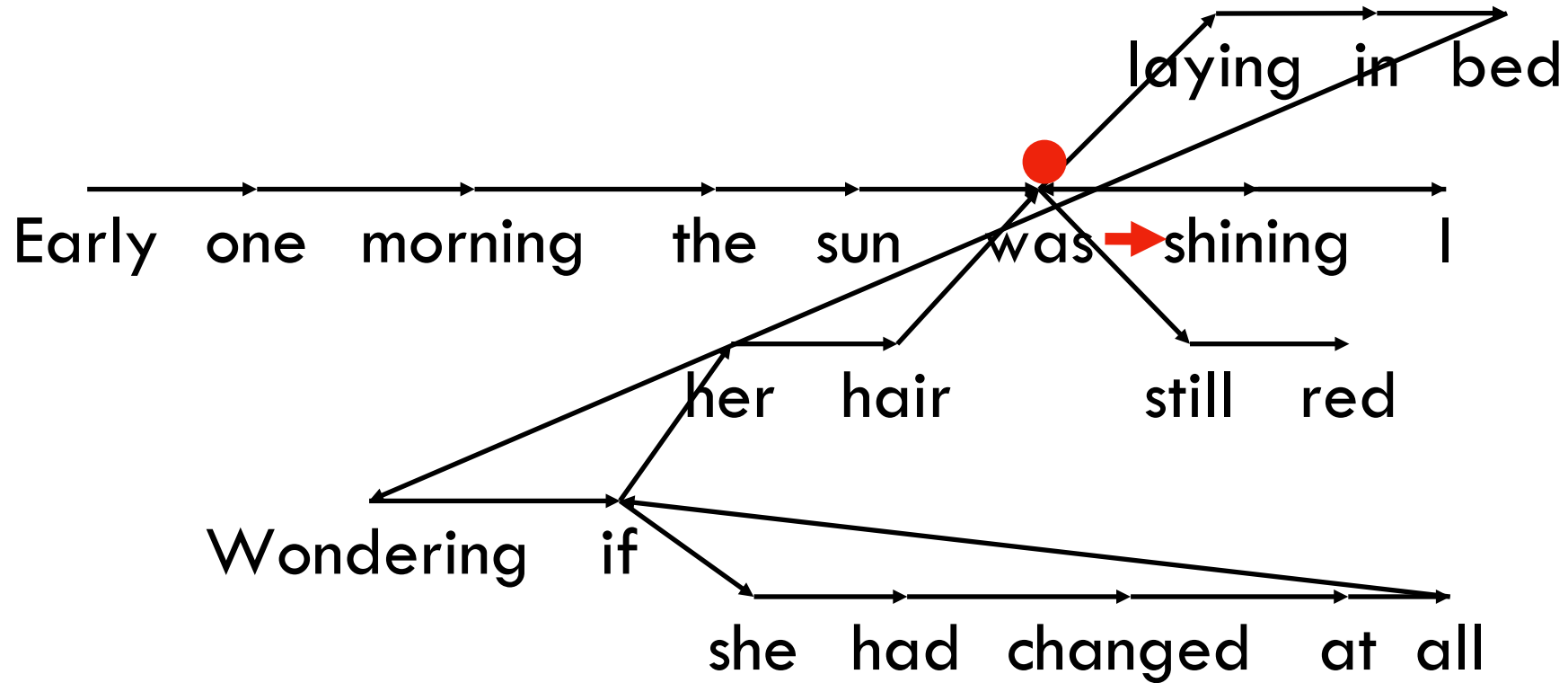
Early one



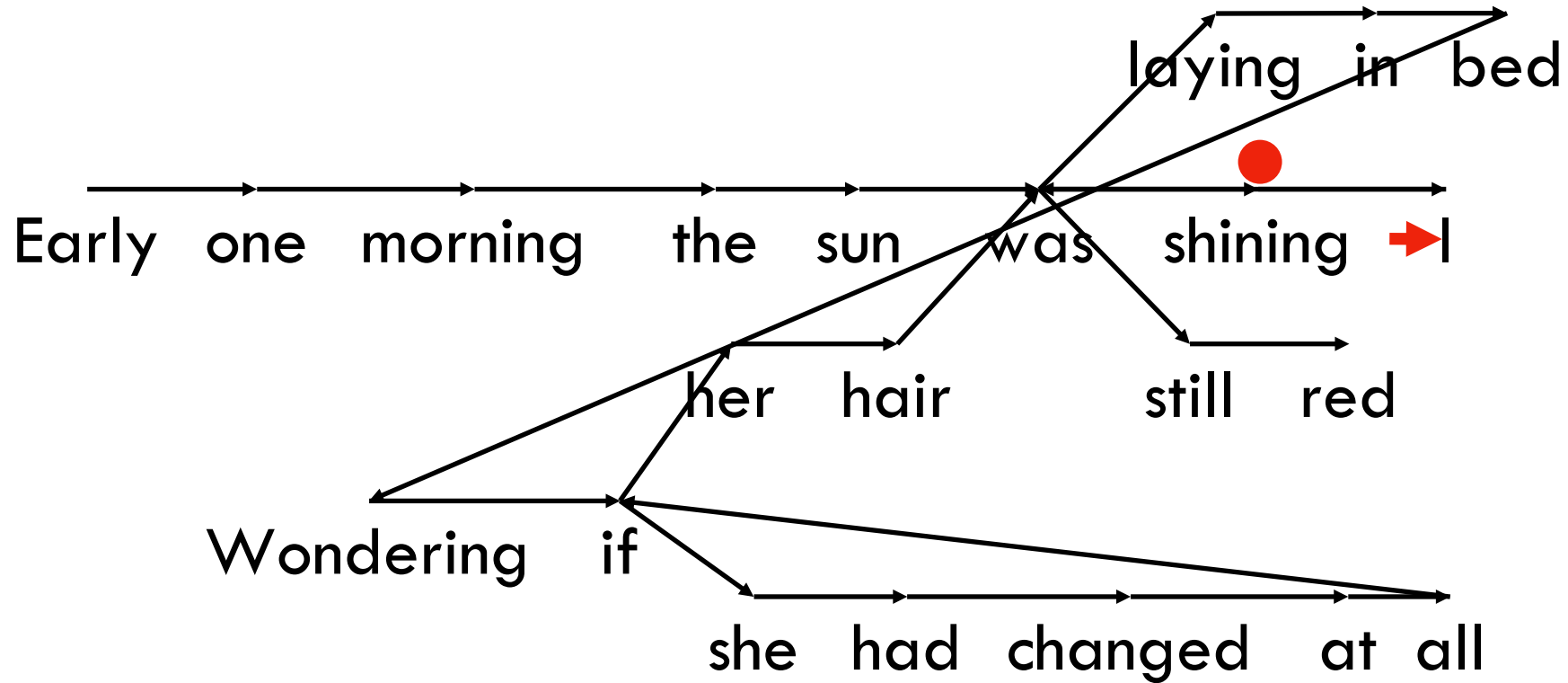
Early one morning



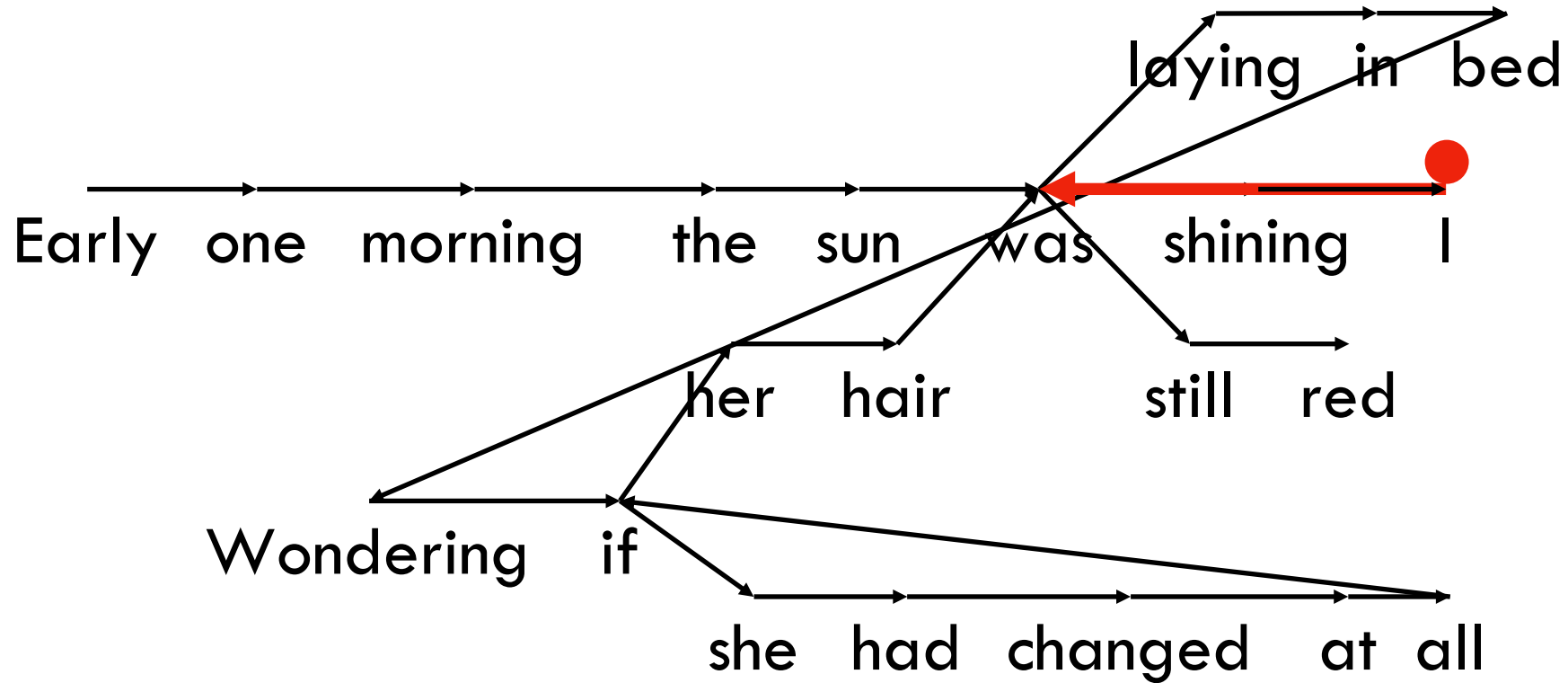
Early one morning the sun was



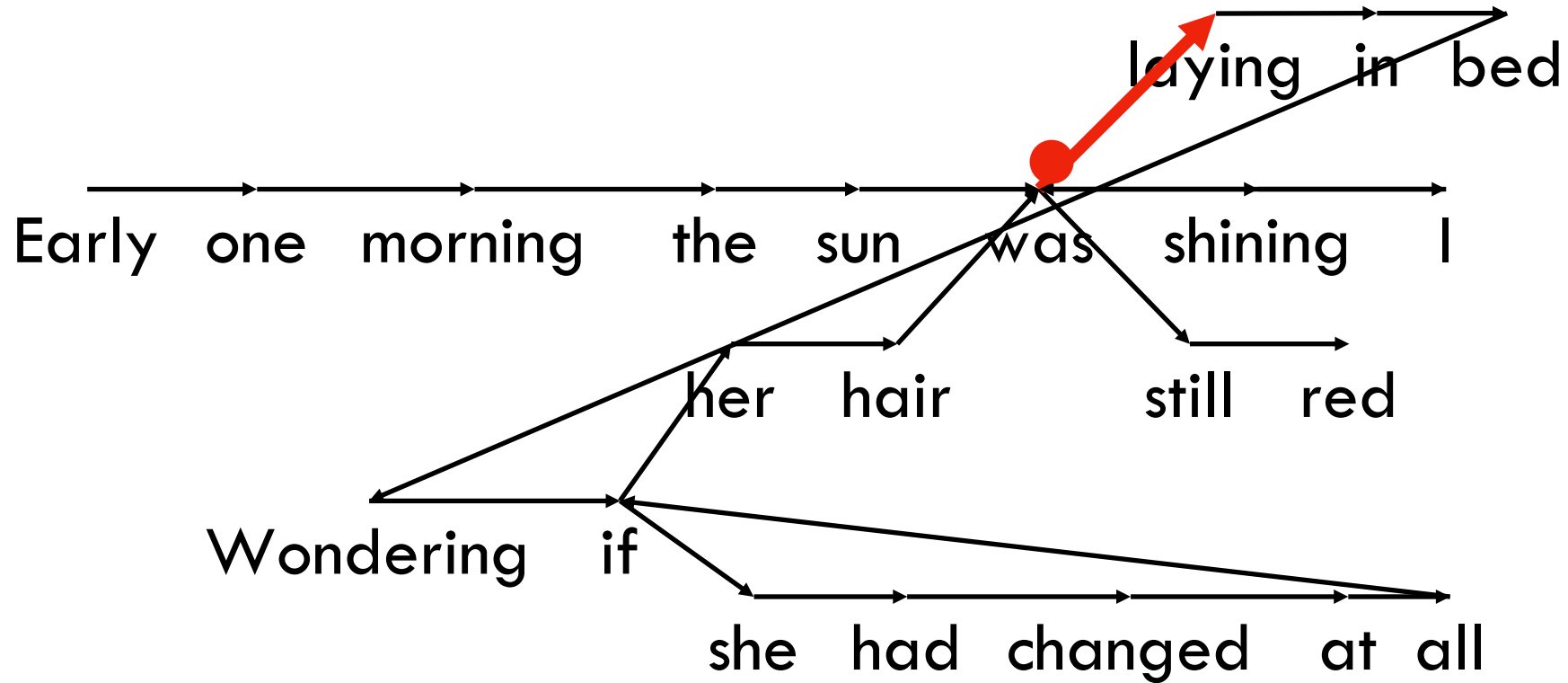
Early one morning the sun was shining



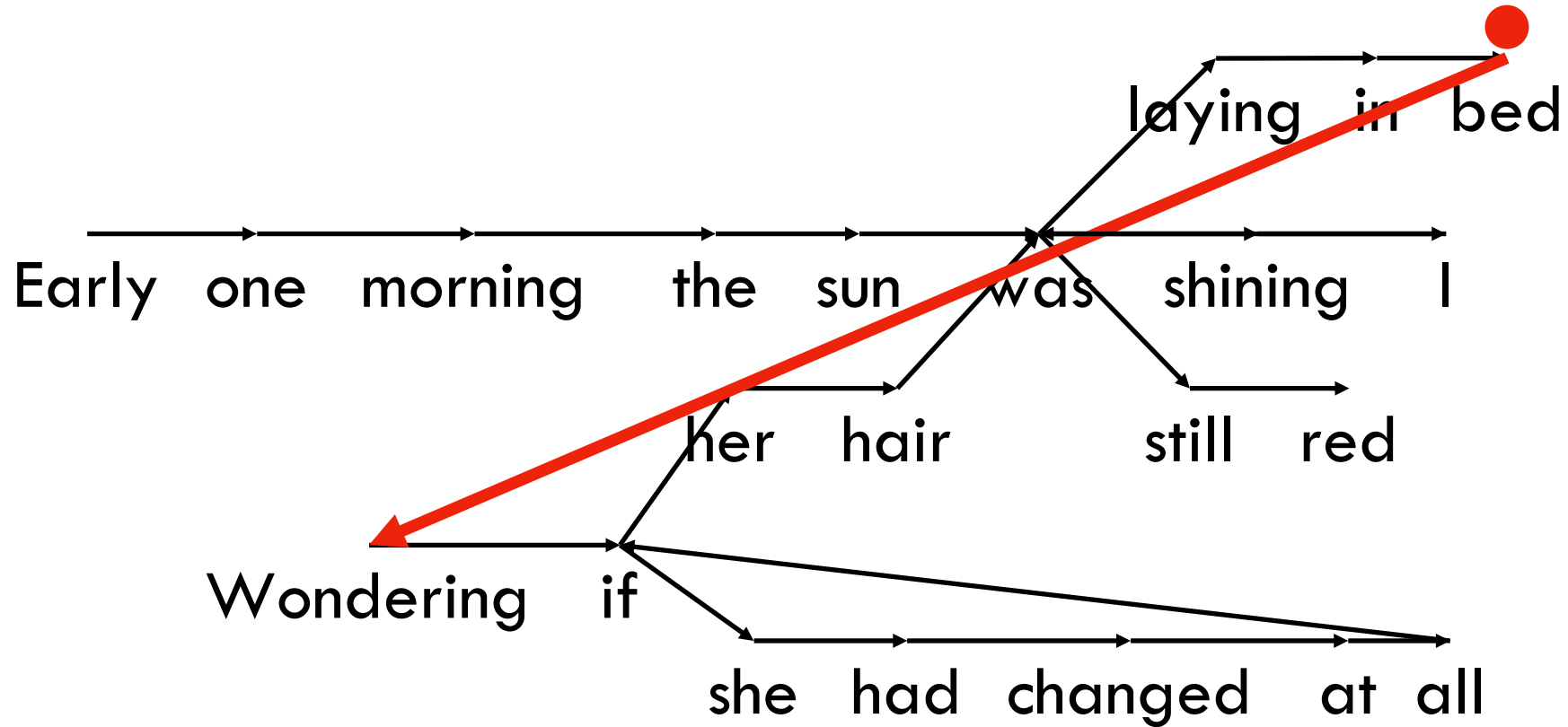
Early one morning the sun was shining I



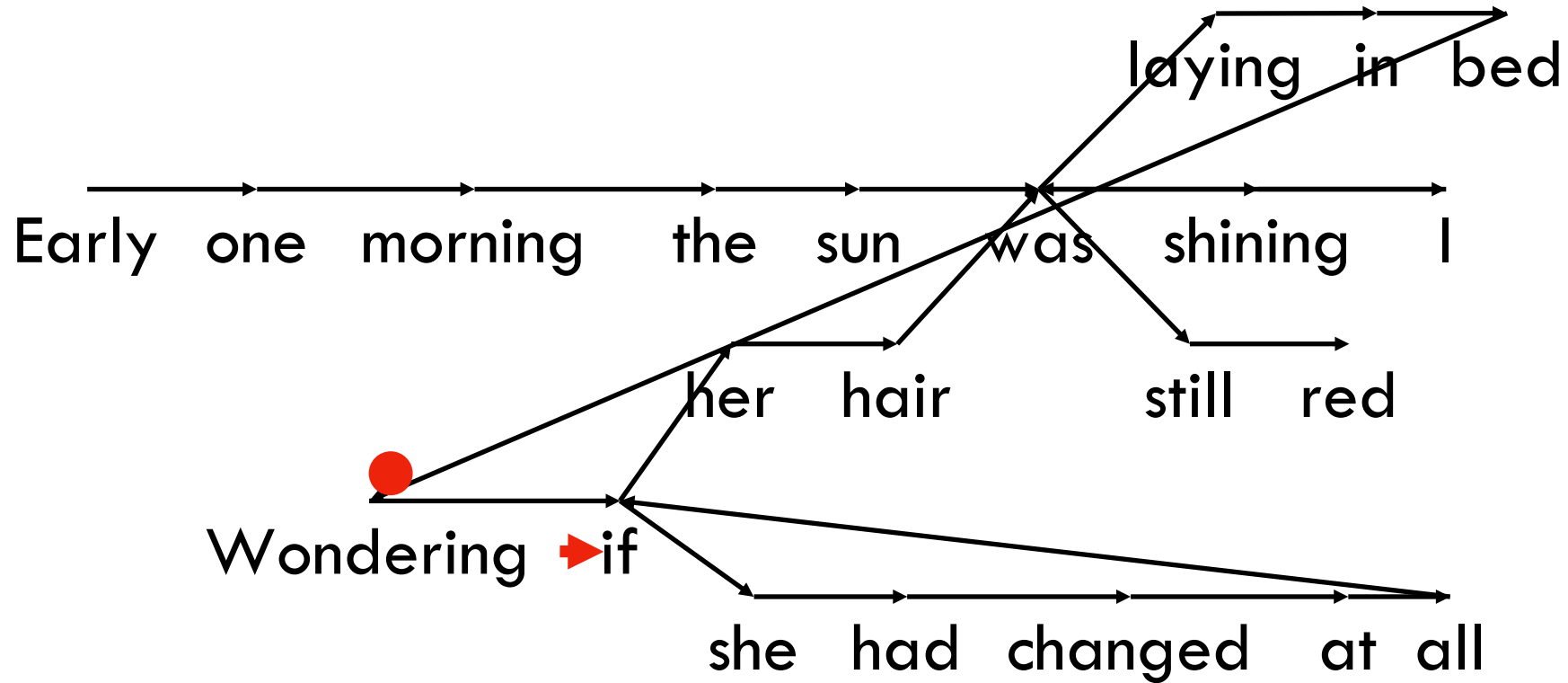
Early one morning the sun was shining I was



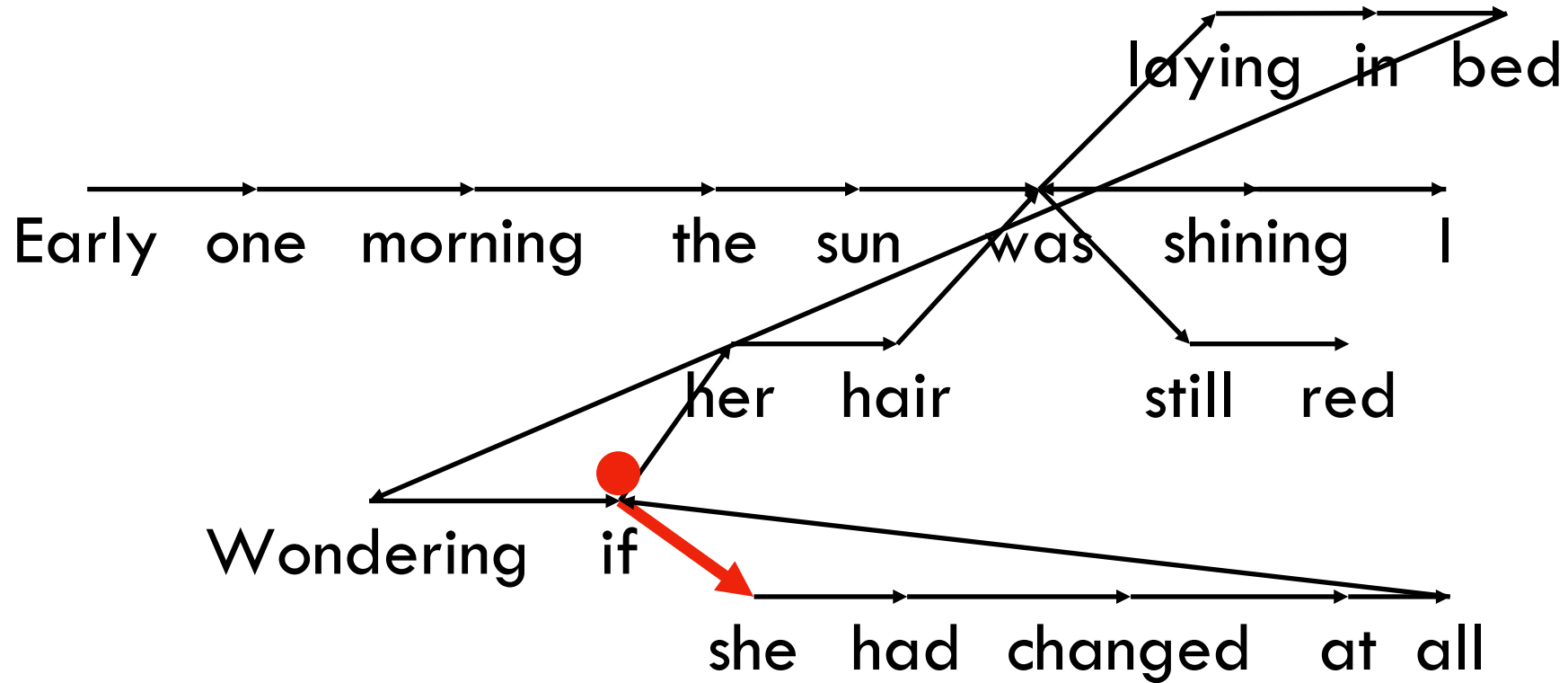
Early one morning the sun was shining I was laying in bed



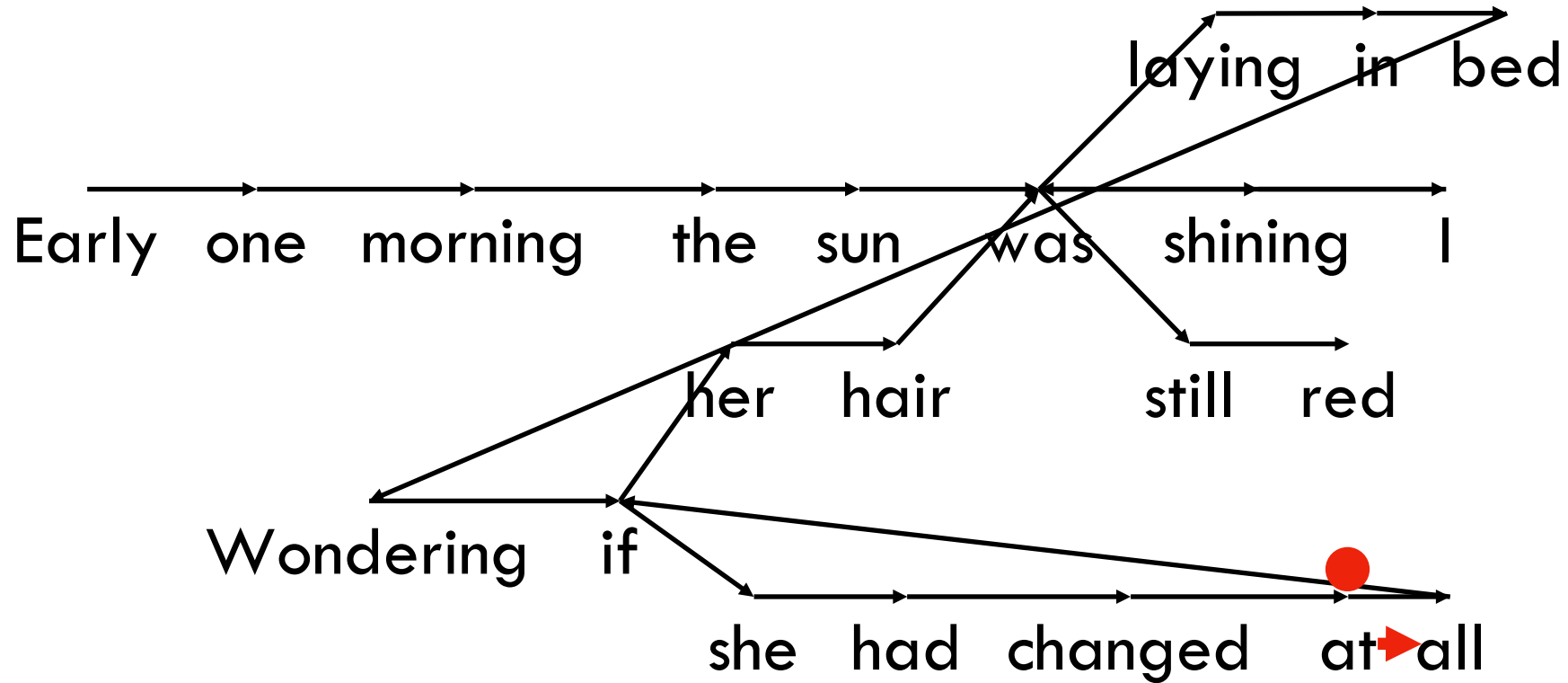
Early one morning the sun was shining I was laying in bed
Wondering



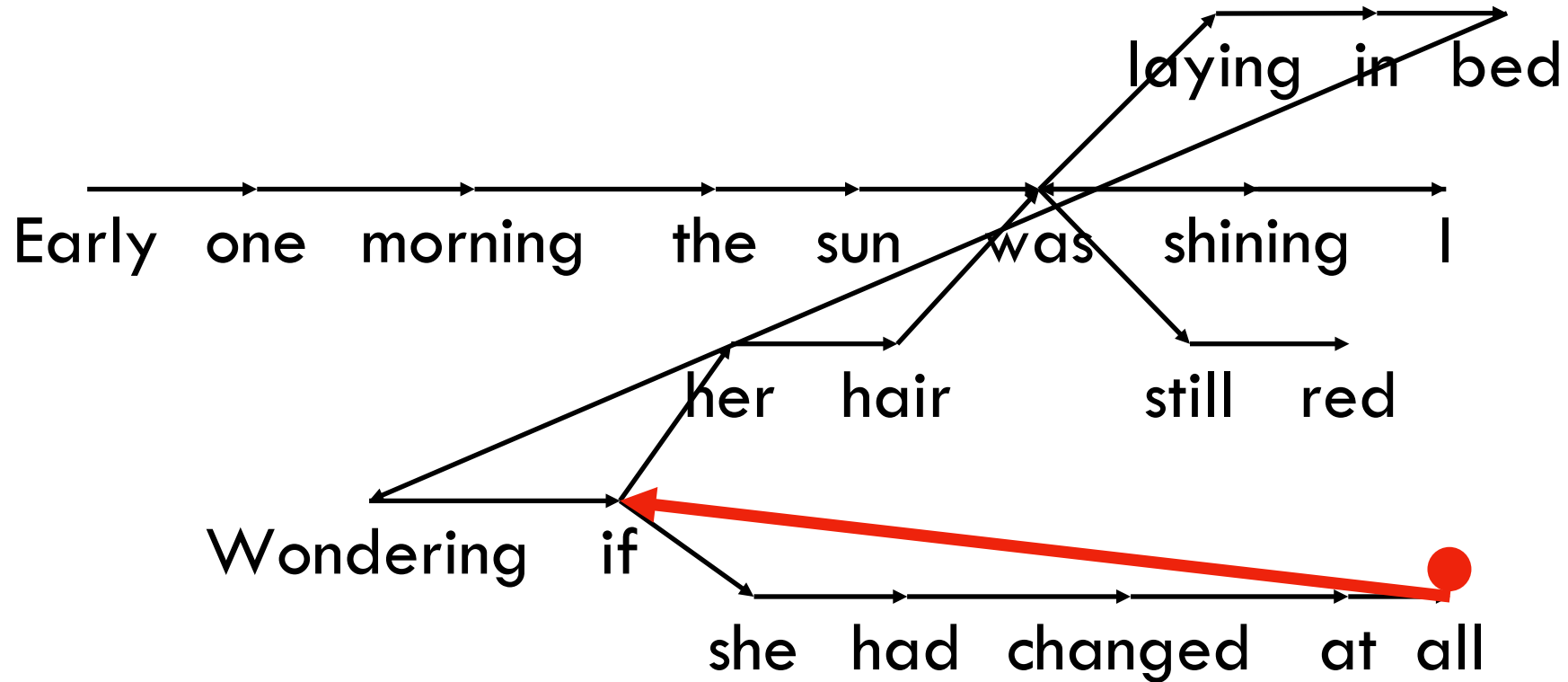
Early one morning the sun was shining I was laying in bed
Wondering if



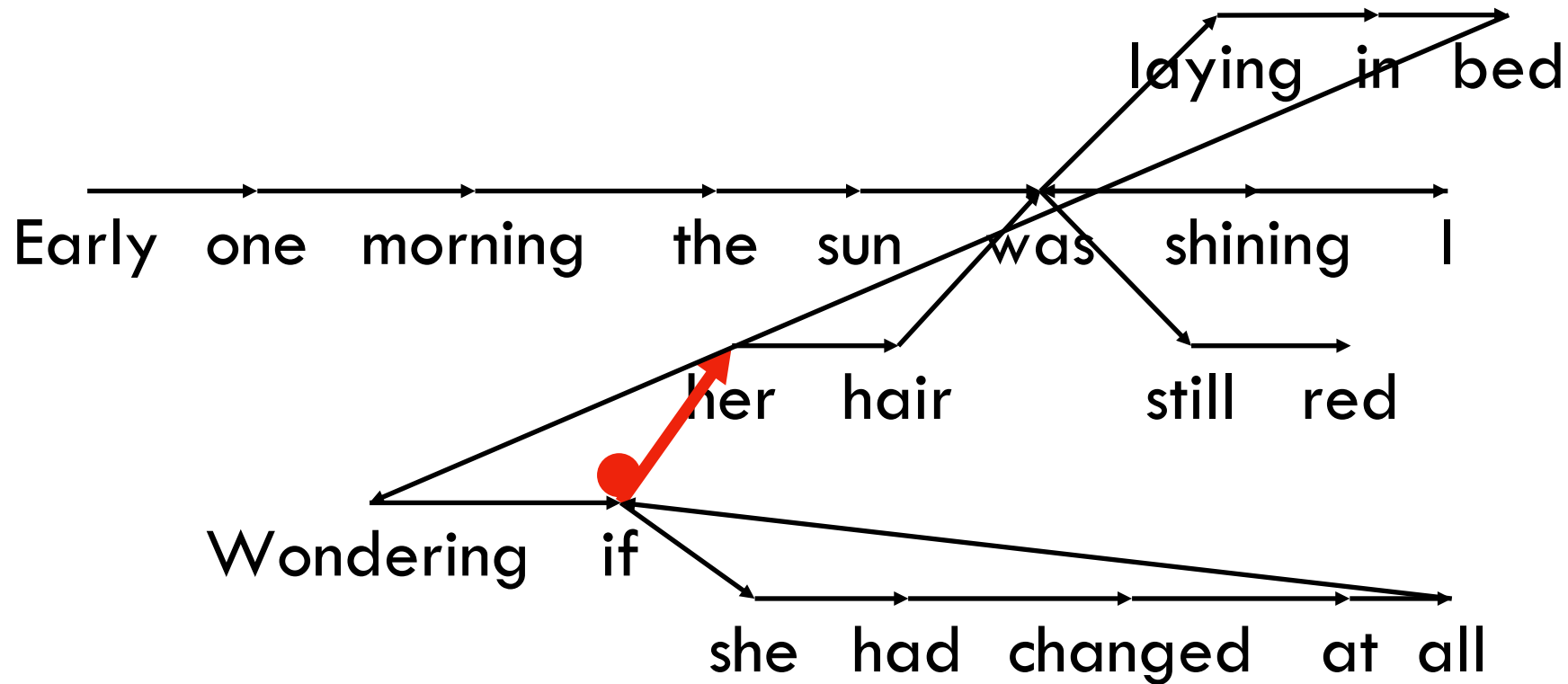
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at



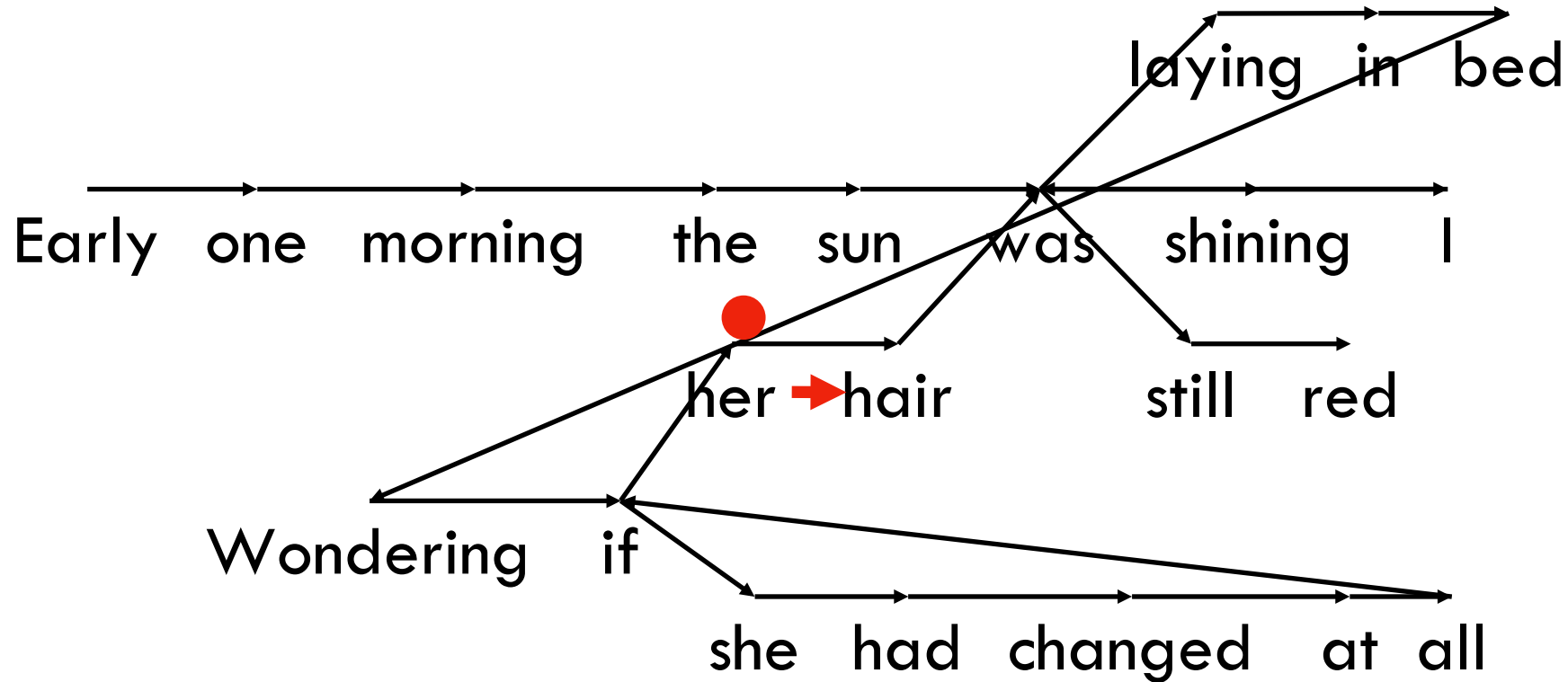
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all



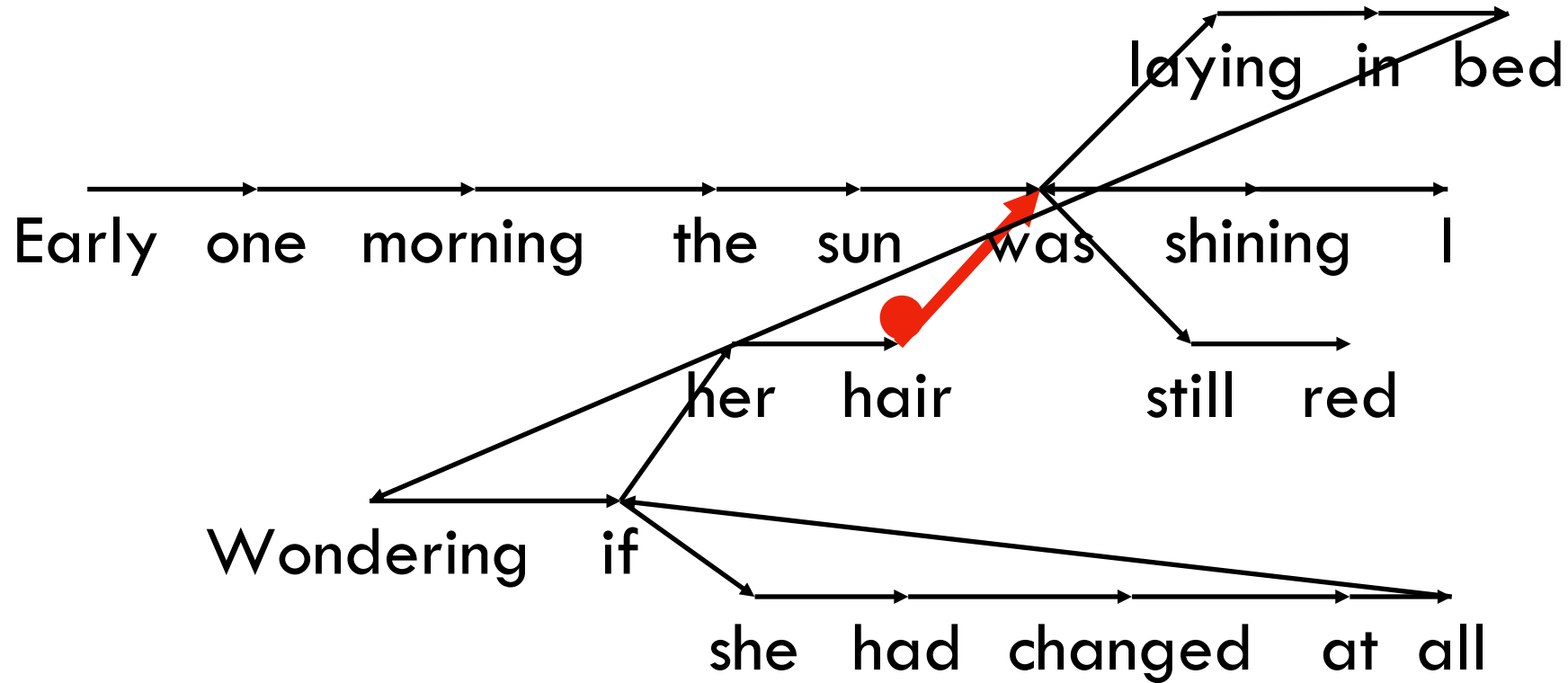
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if



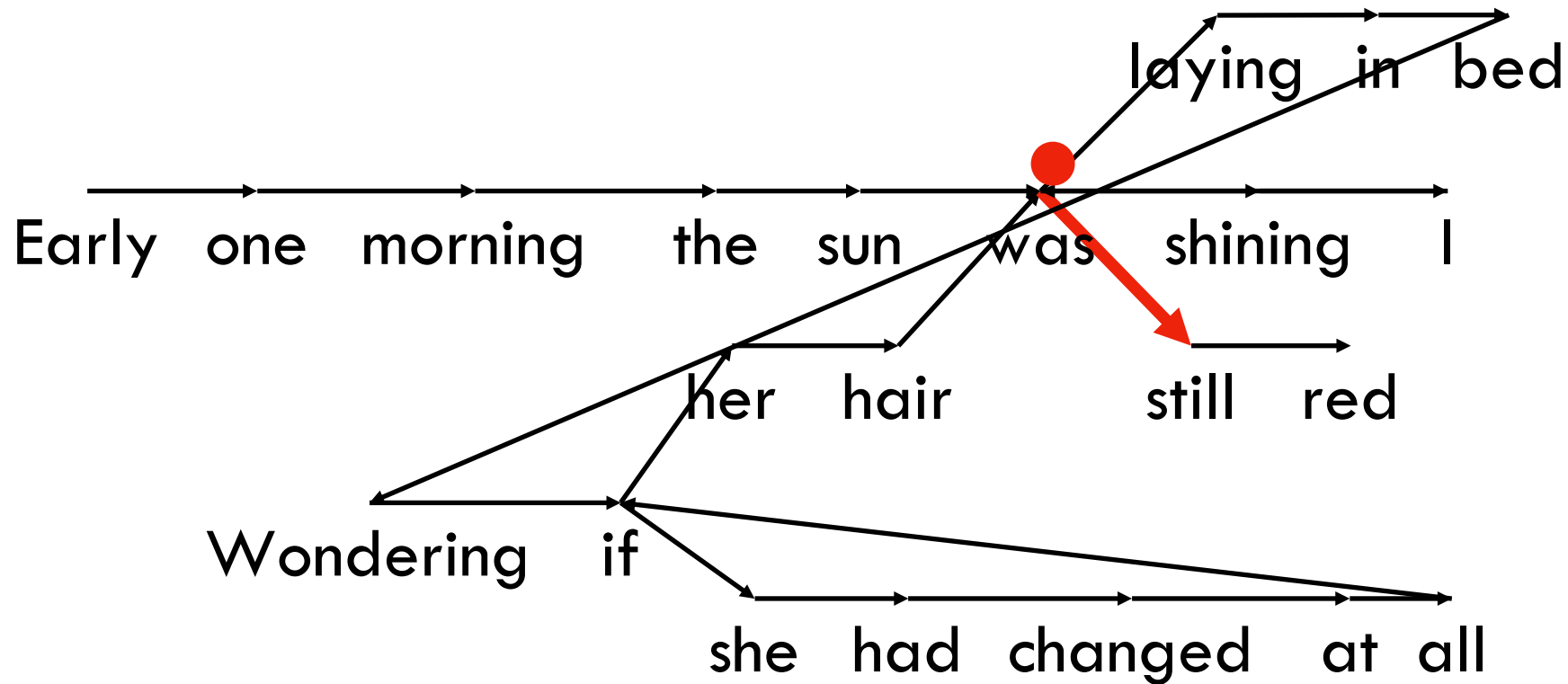
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her



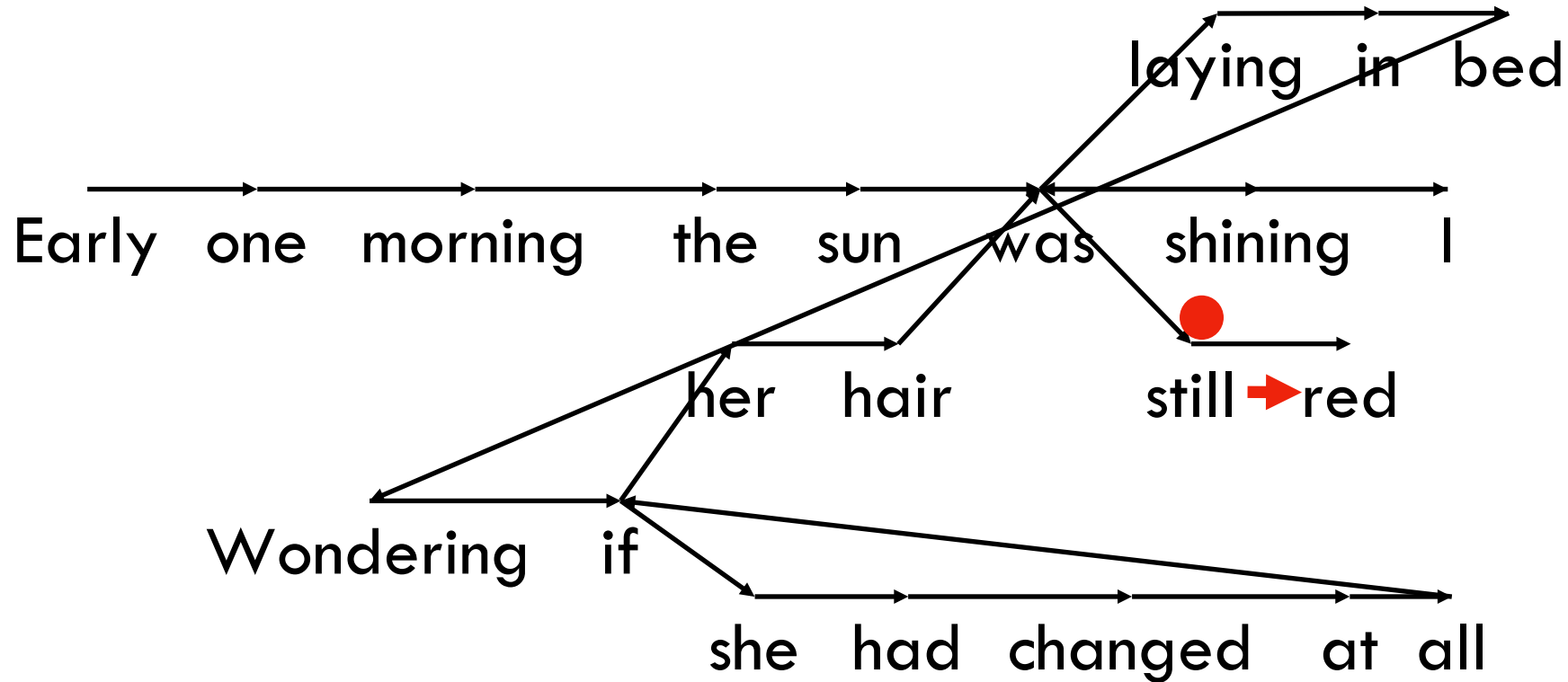
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair



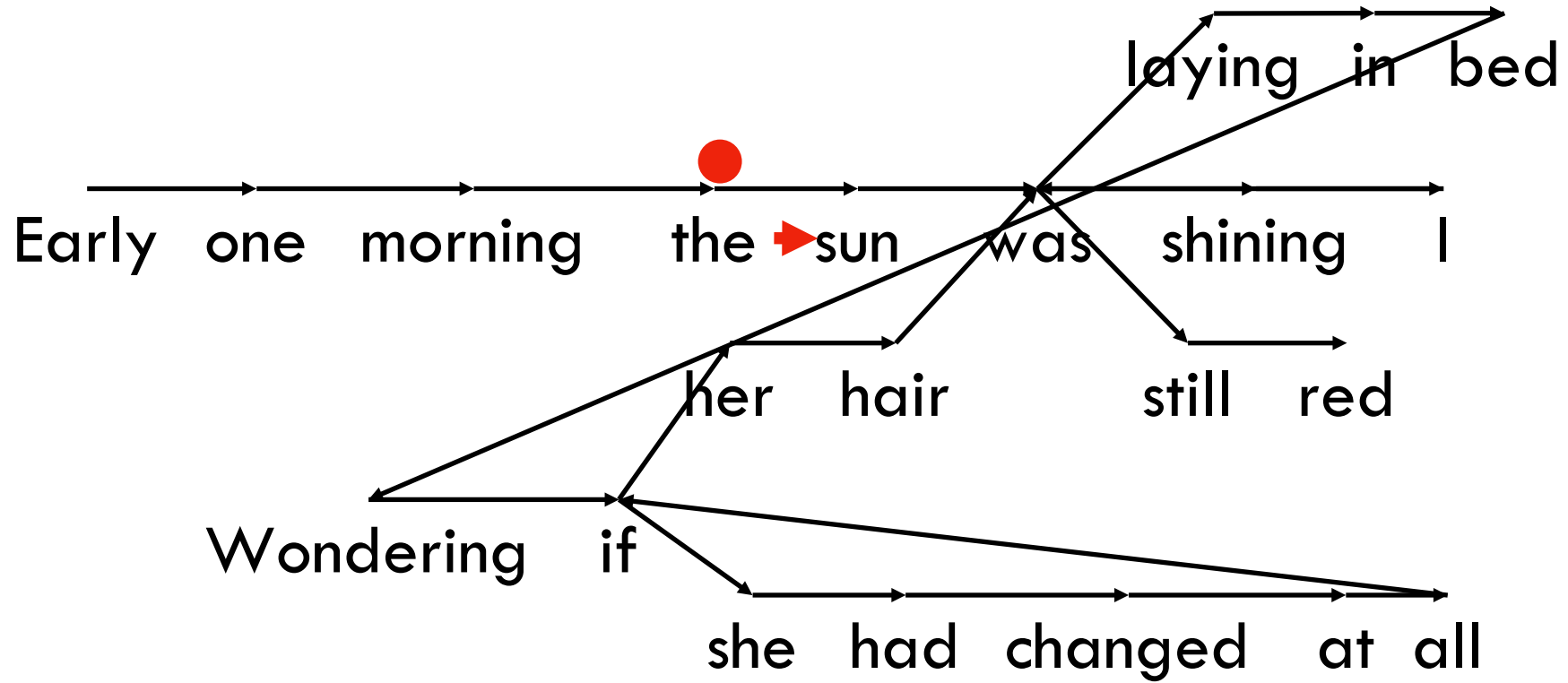
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was



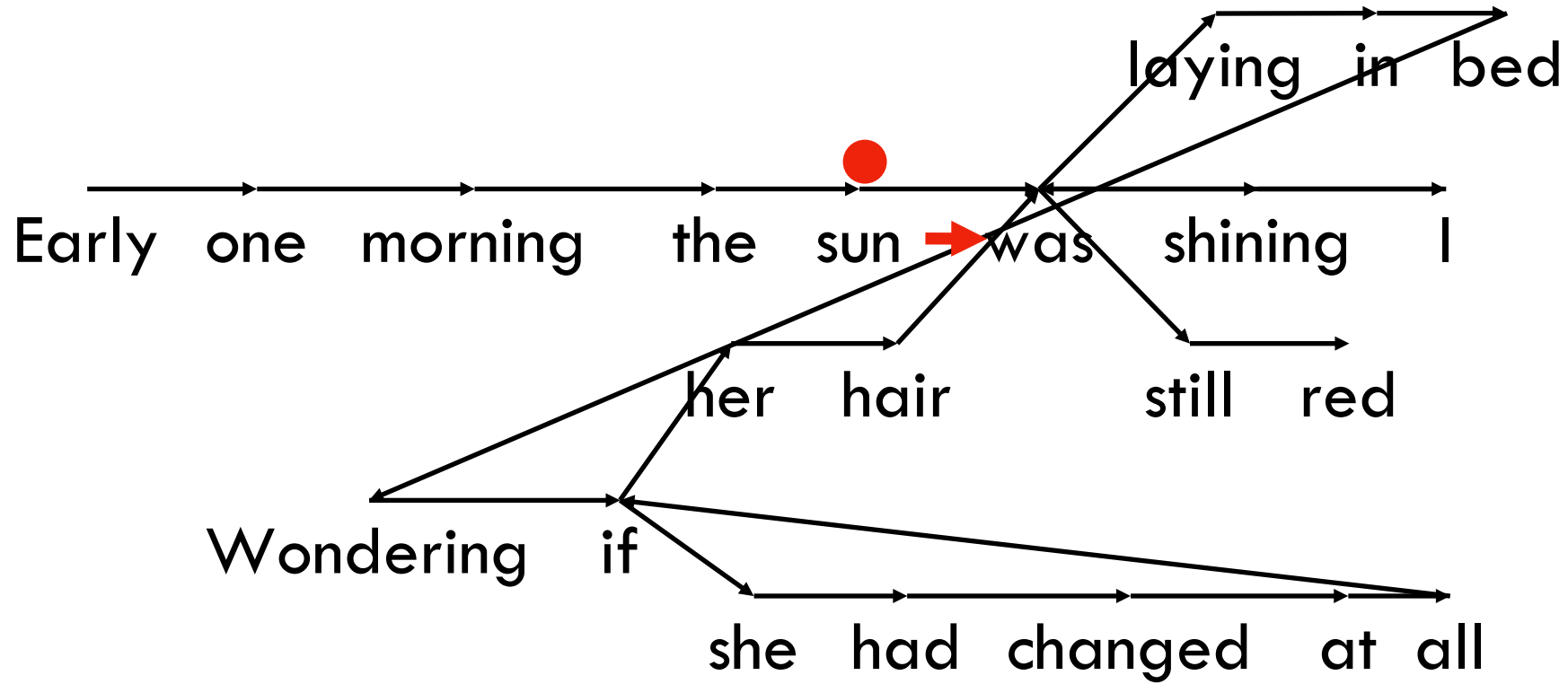
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still



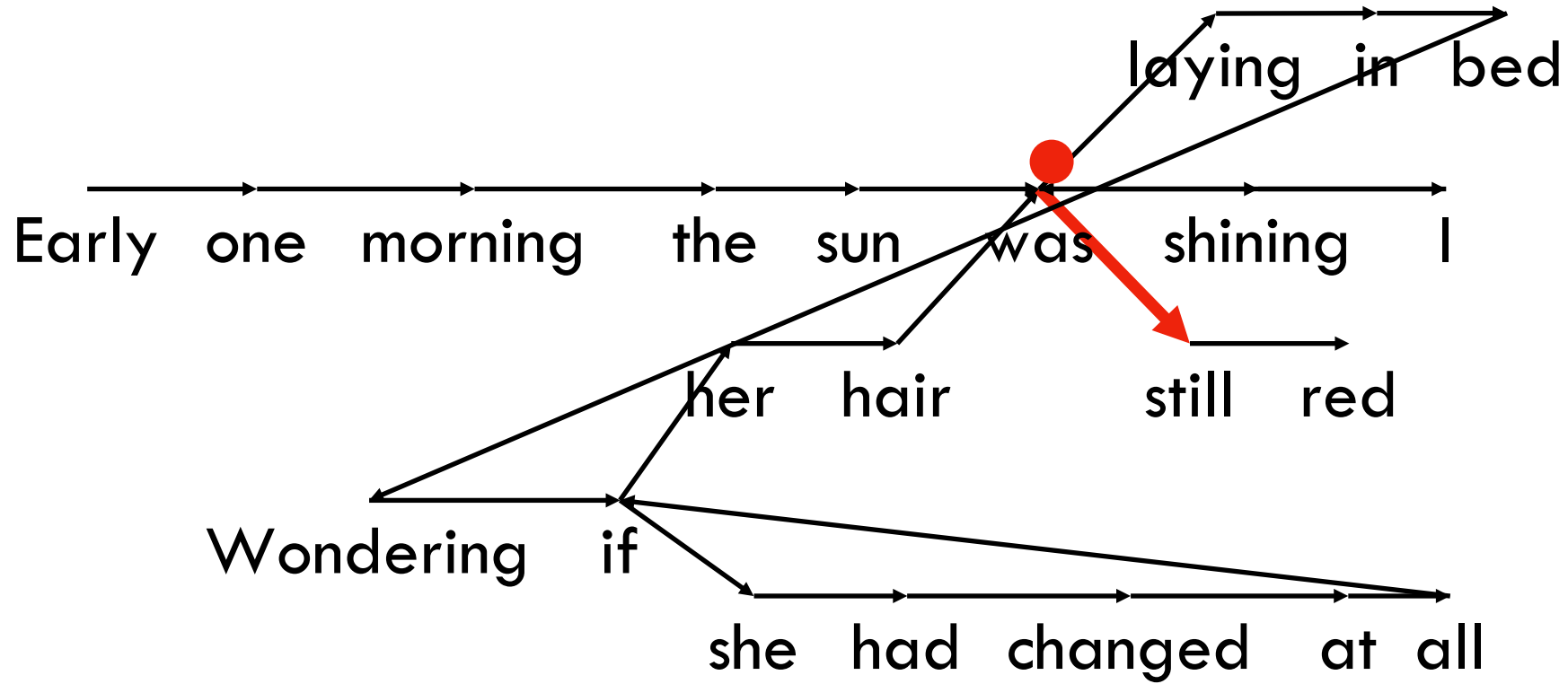
the



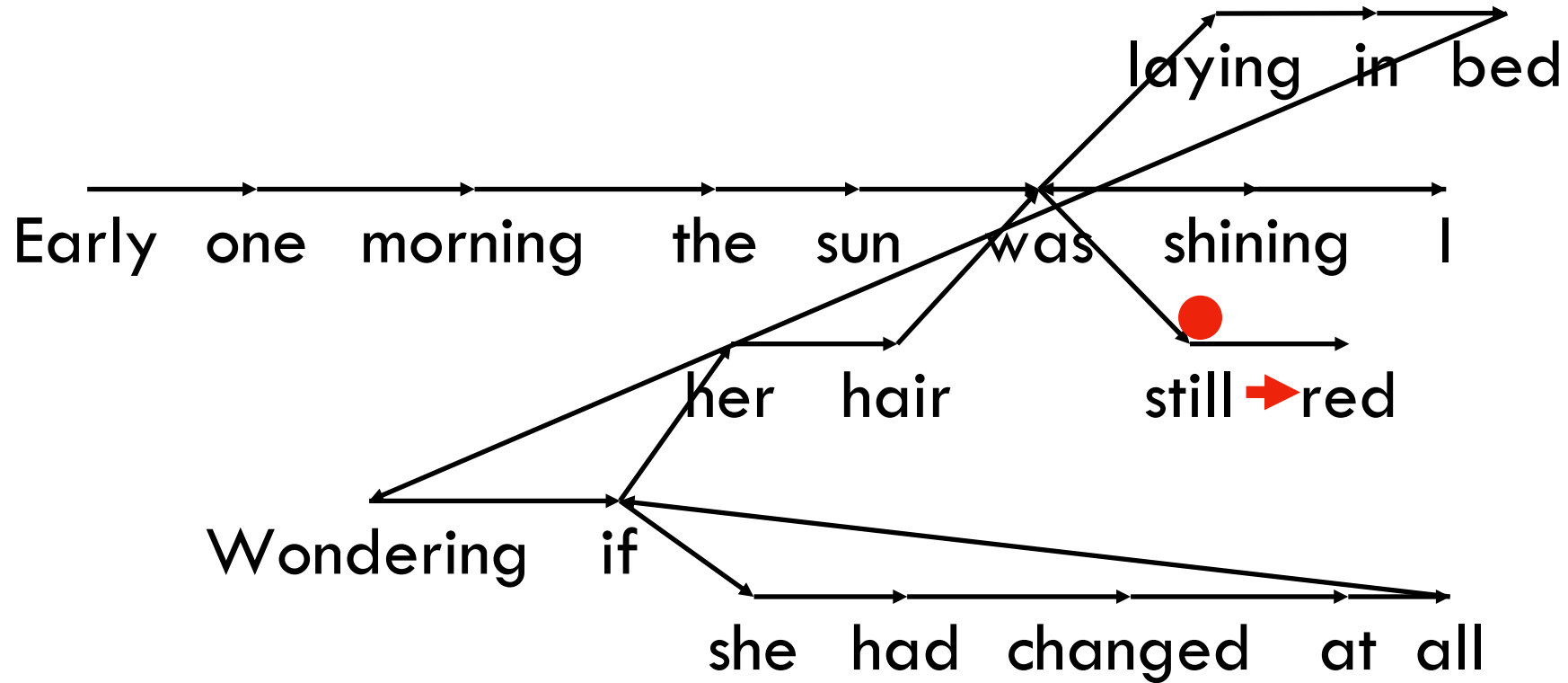
the sun



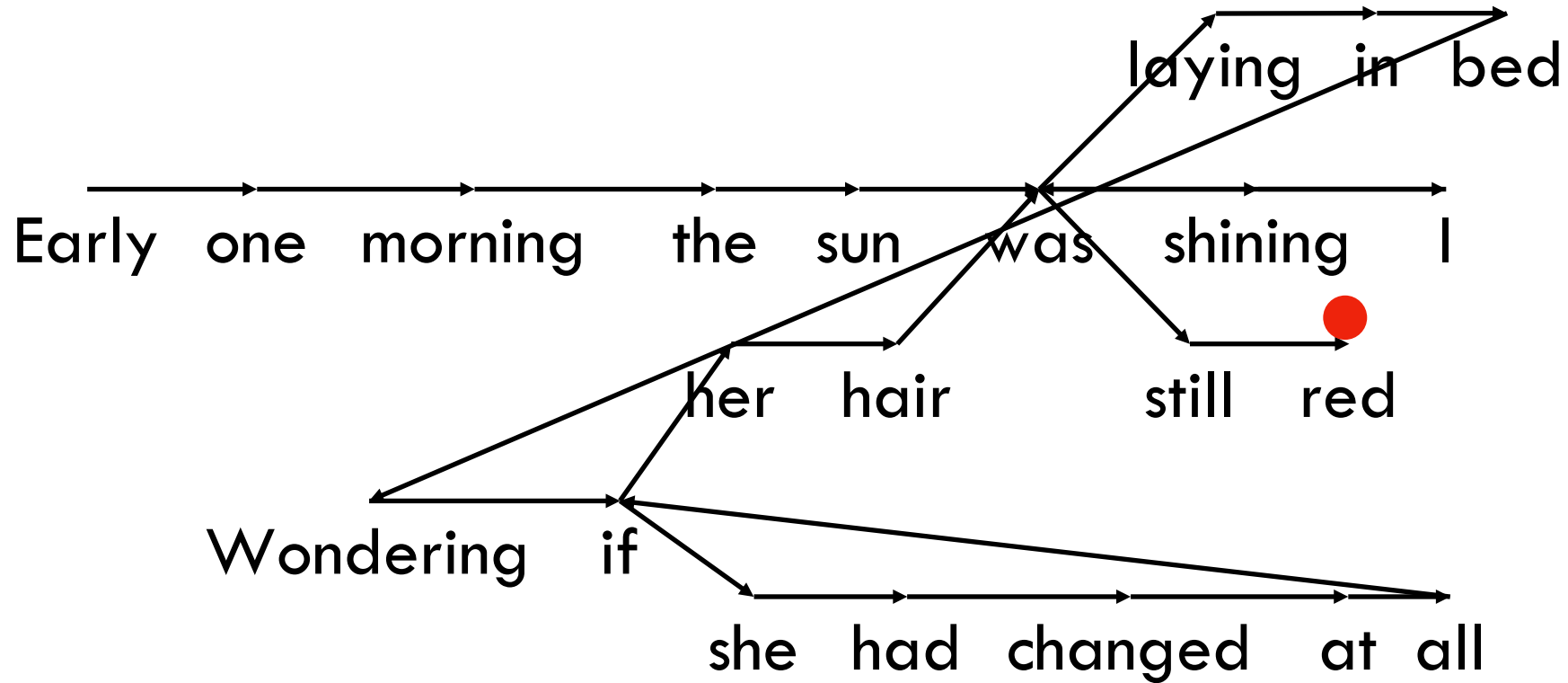
the sun was



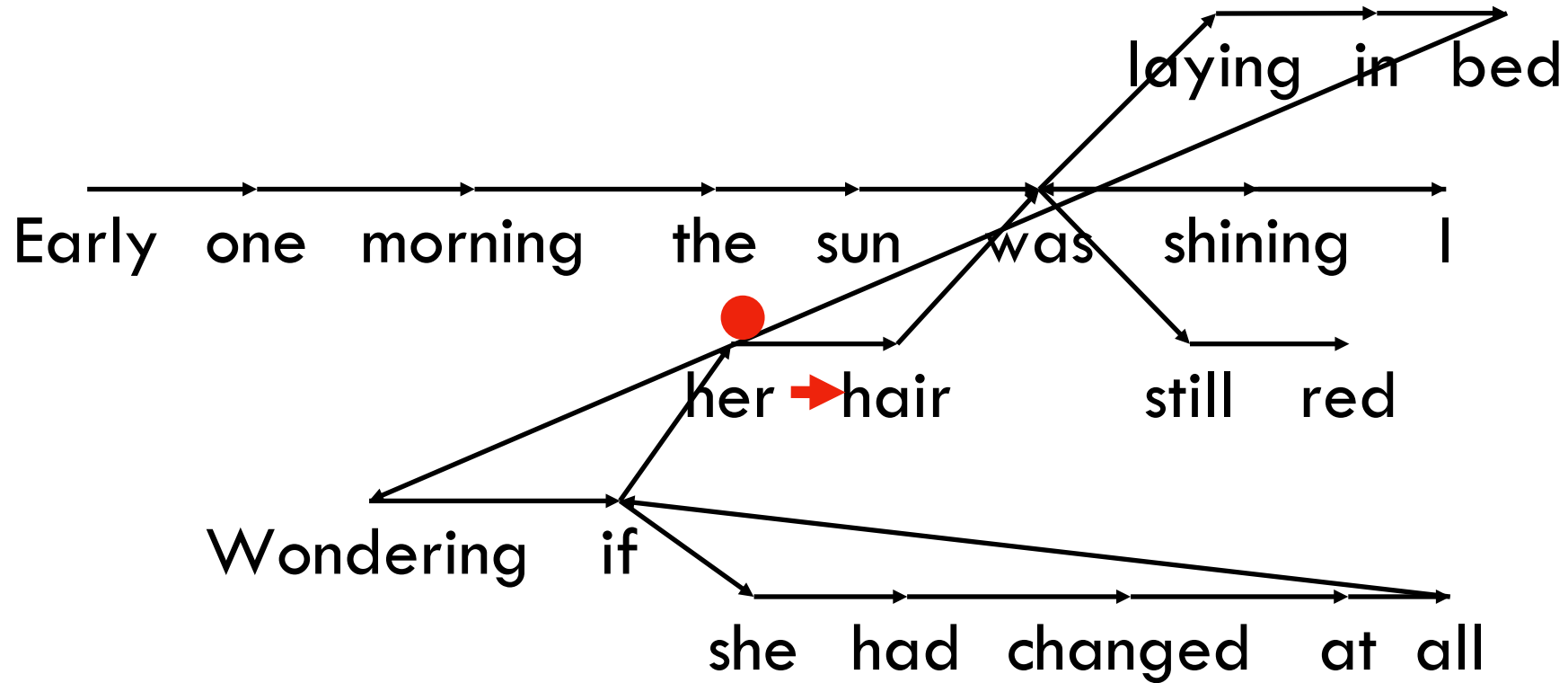
the sun was still



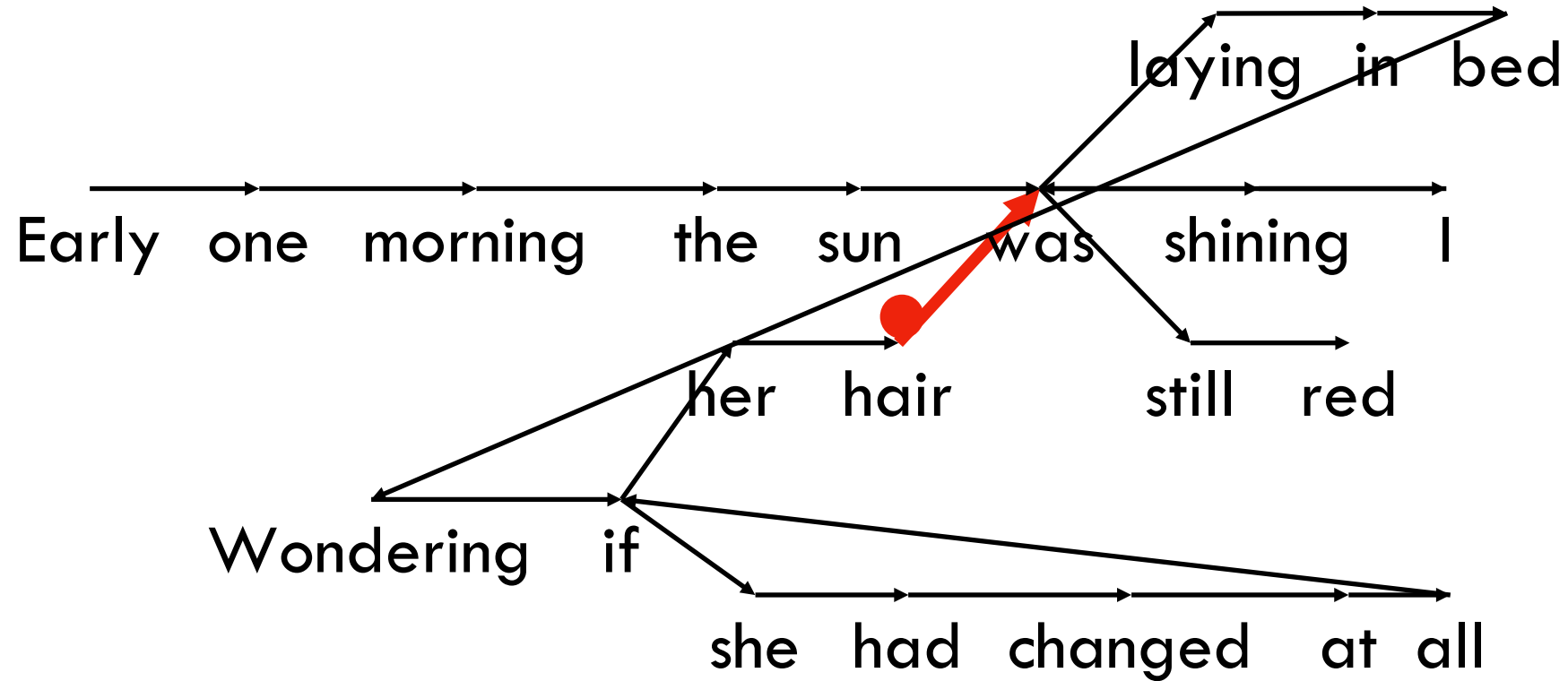
the sun was still red



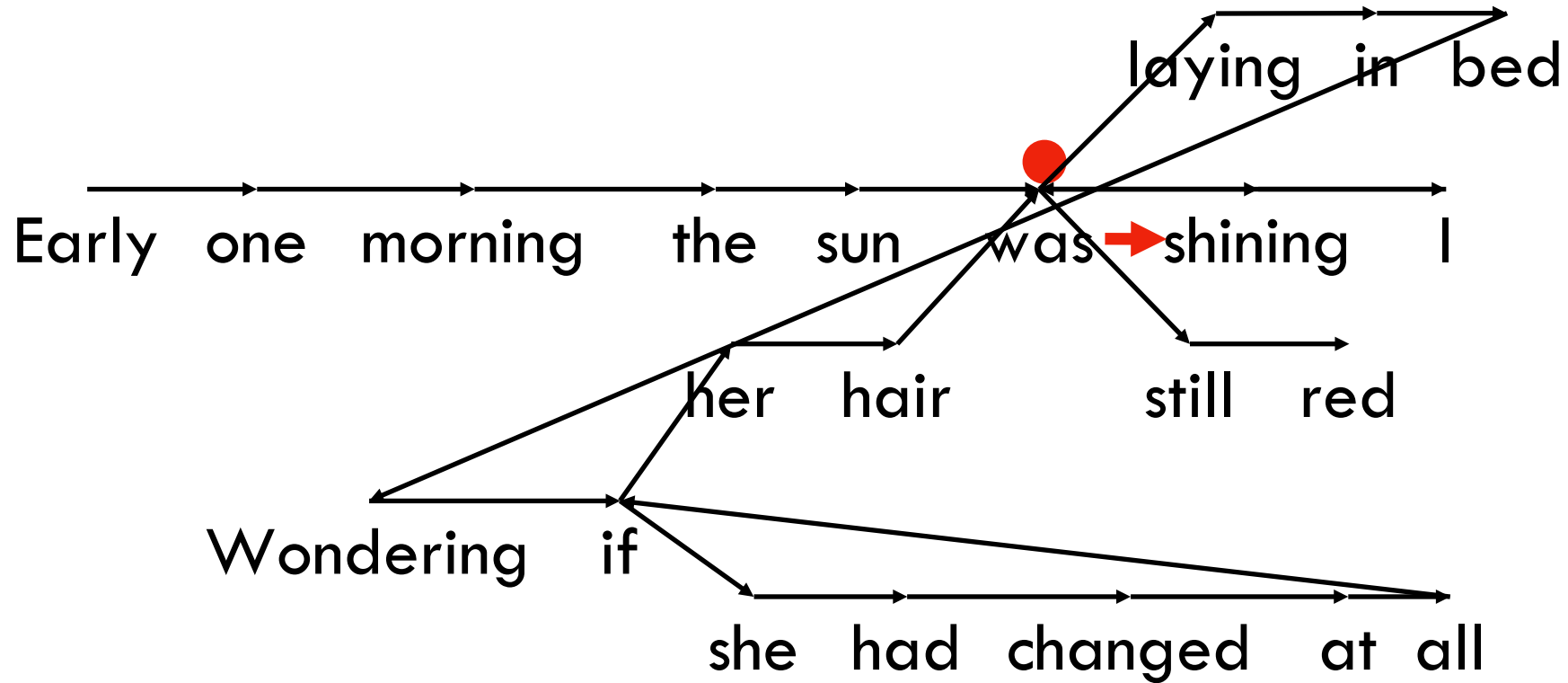
the sun was still red
her



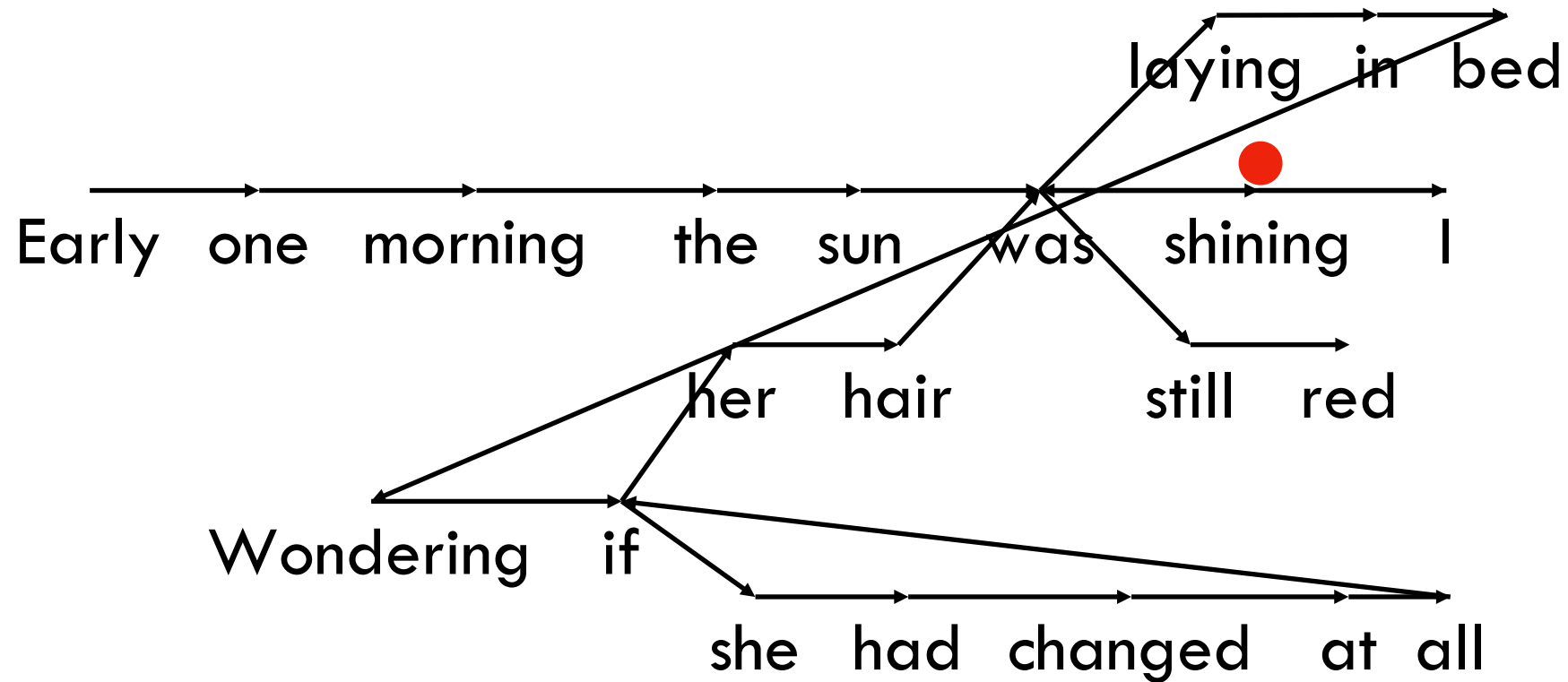
the sun was still red
her hair



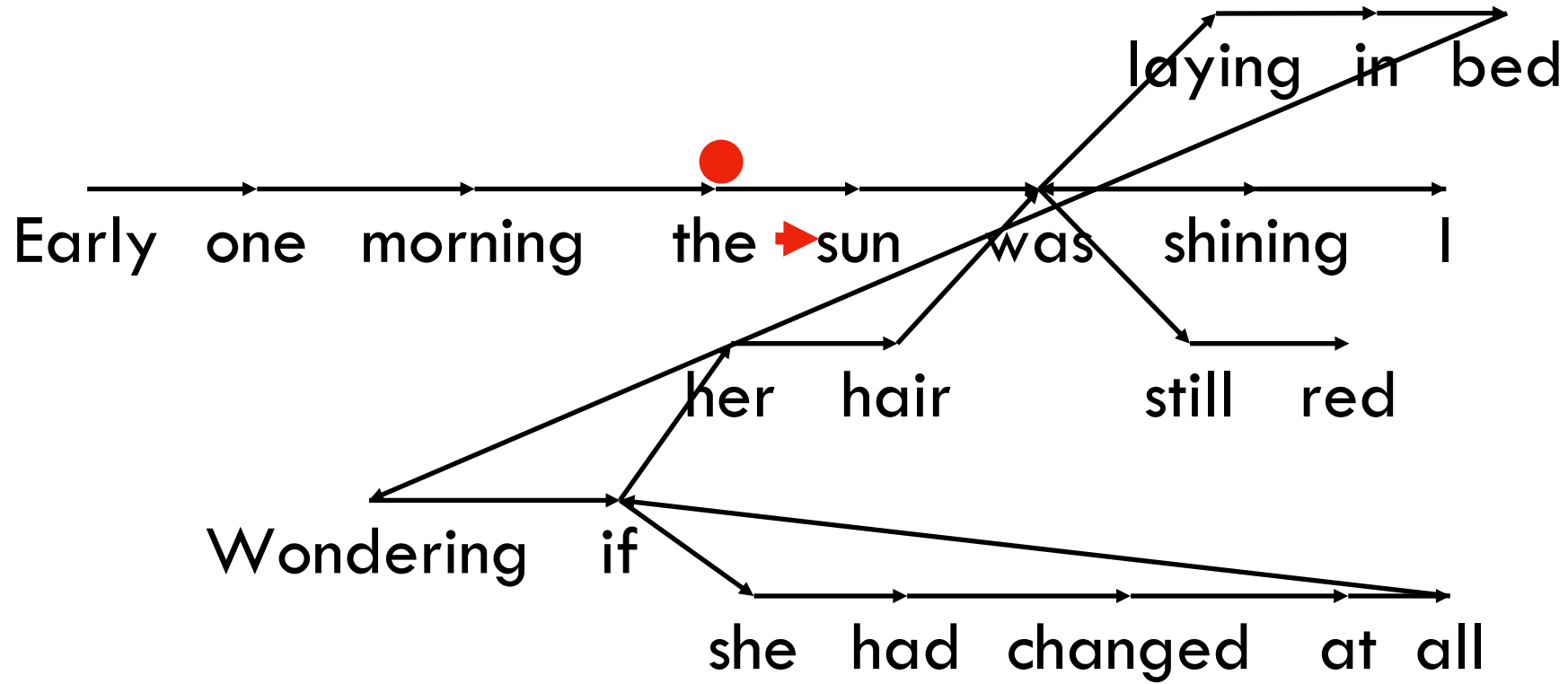
the sun was still red
her hair was



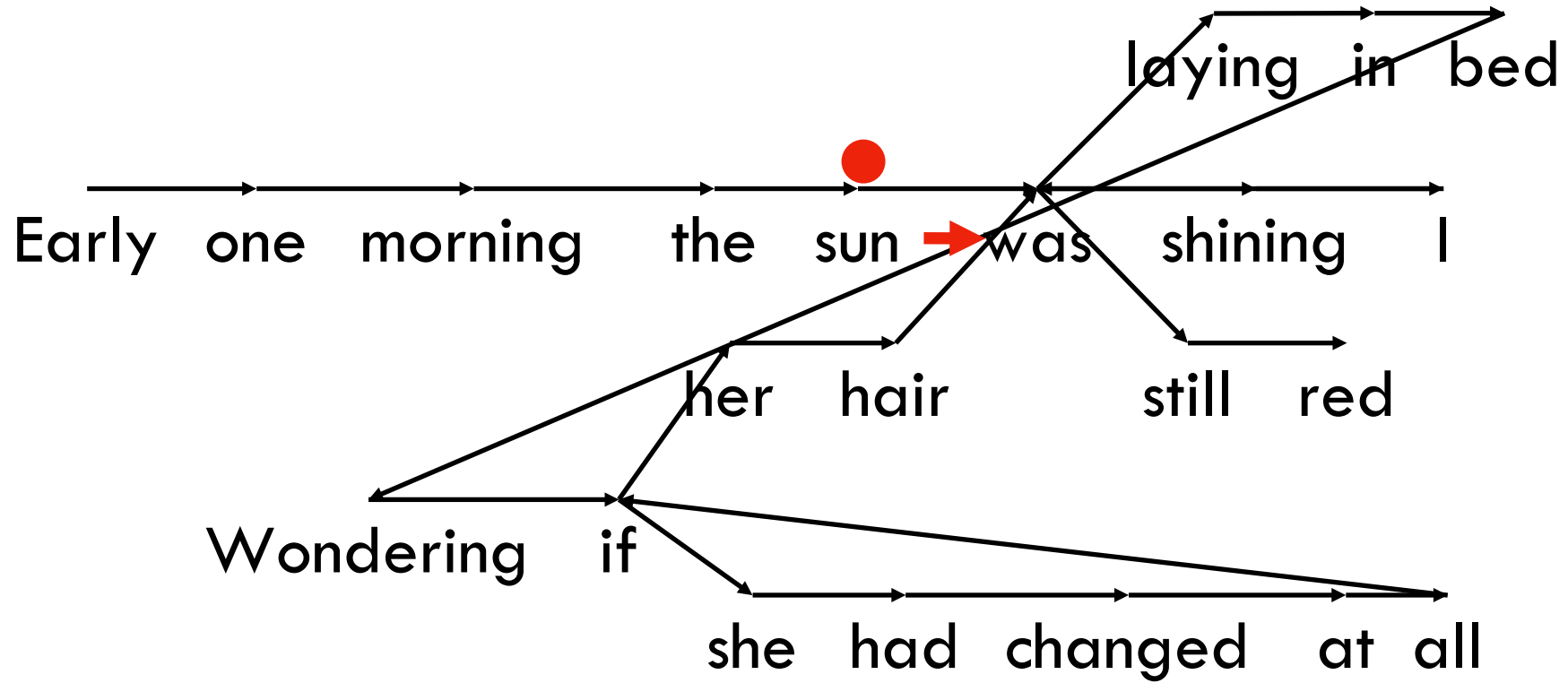
the sun was still red
her hair was shining



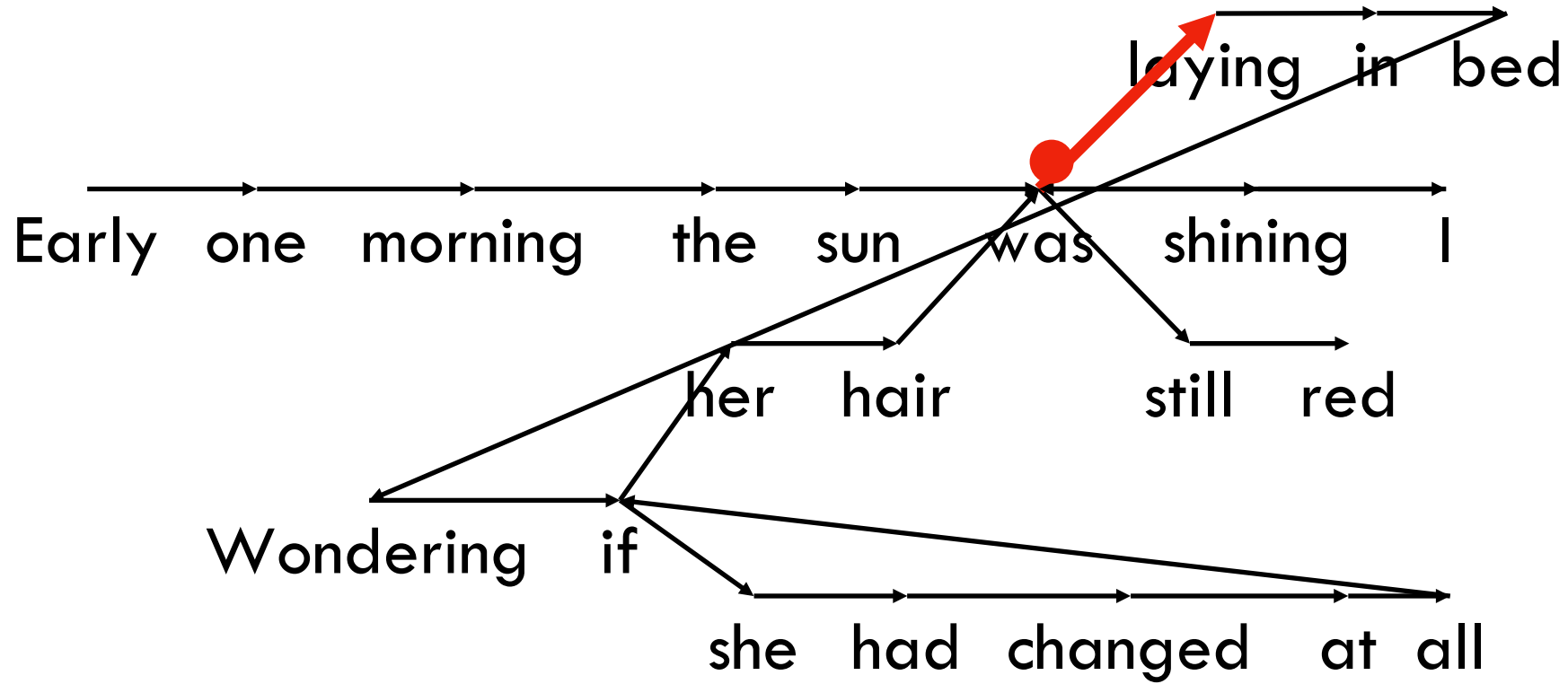
the



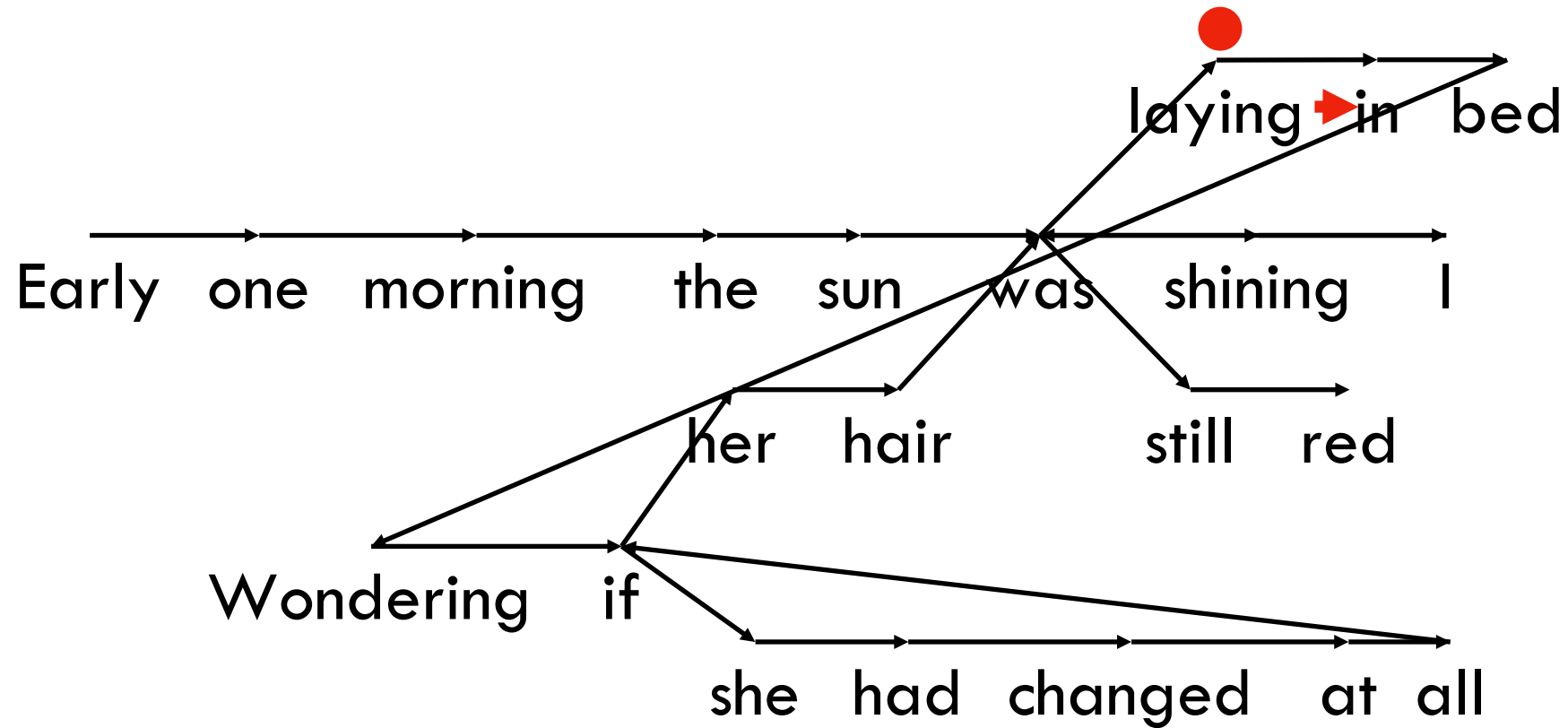
the sun



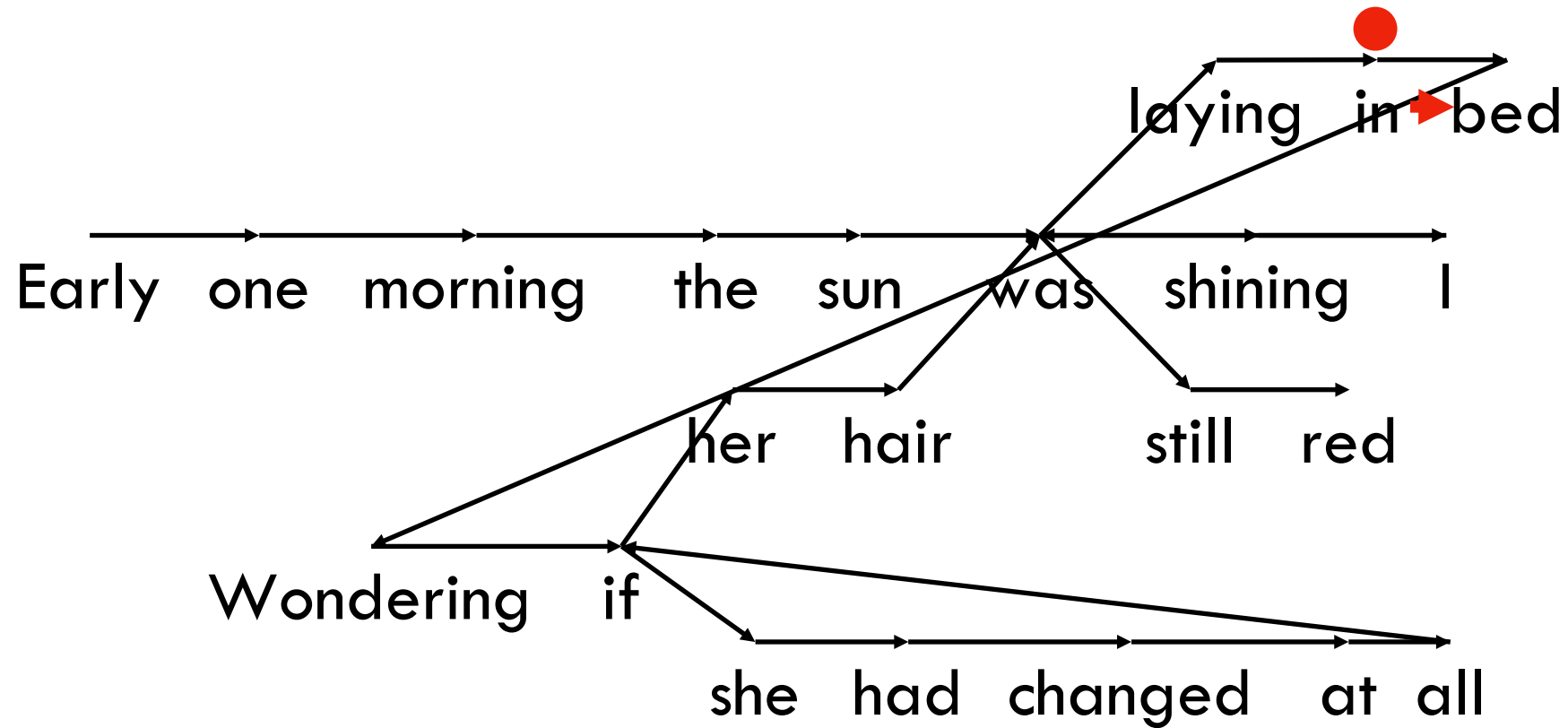
the sun was



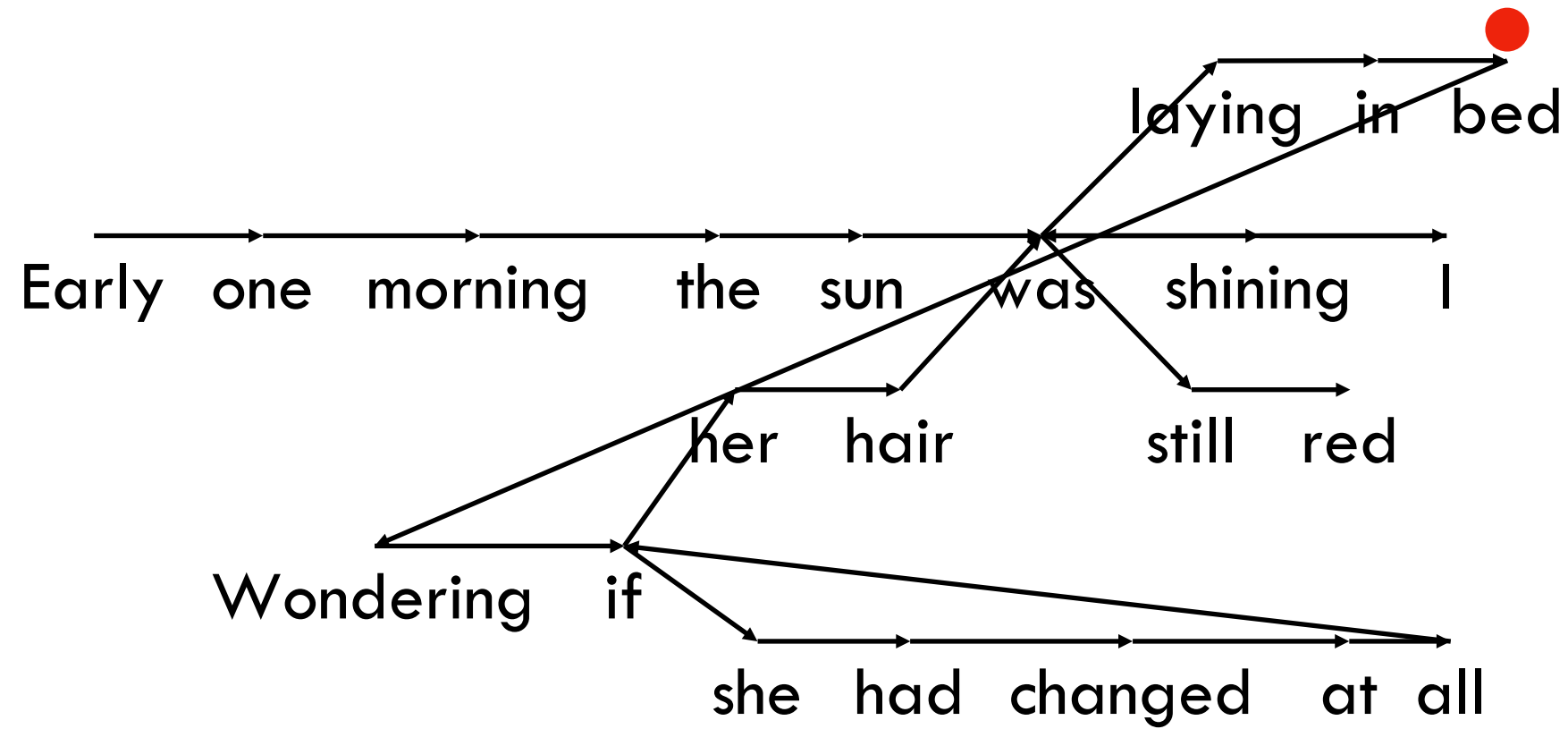
the sun was laying



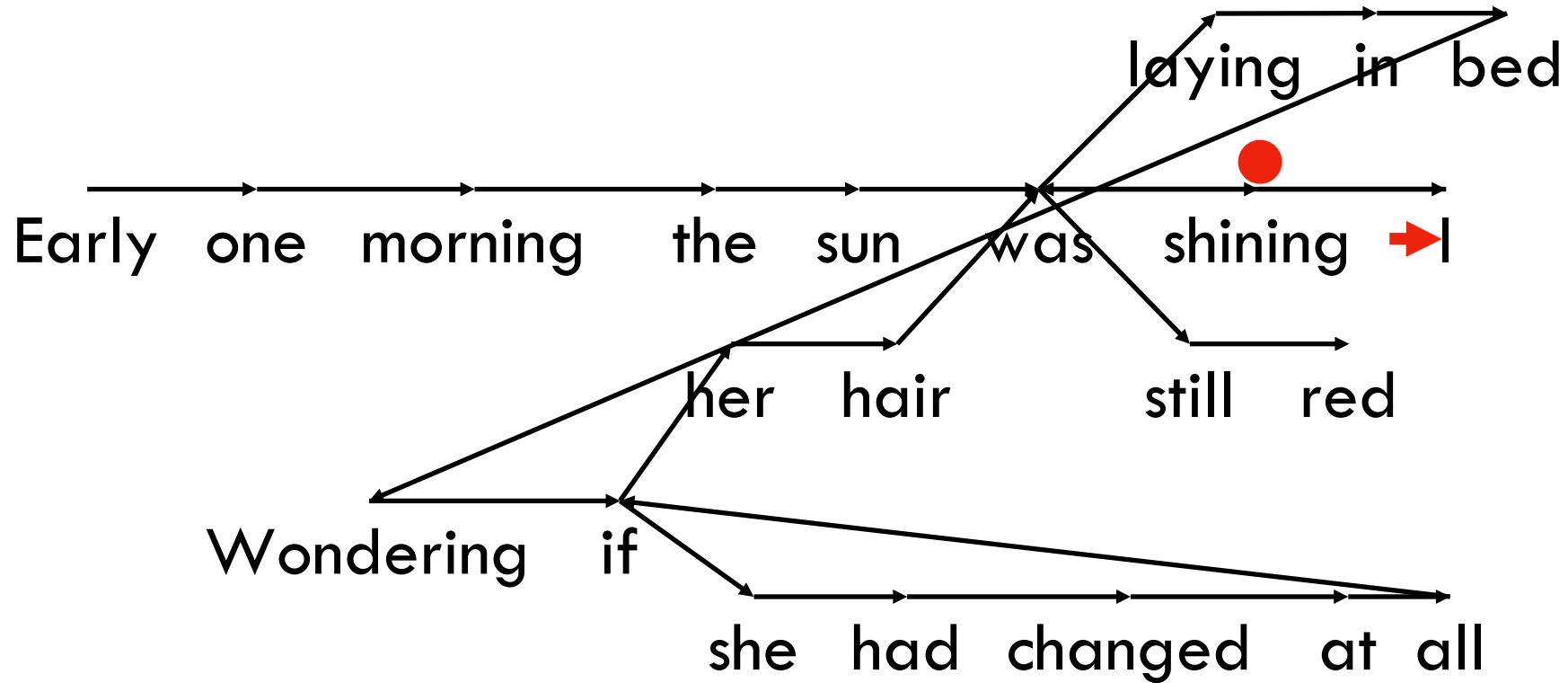
the sun was laying in



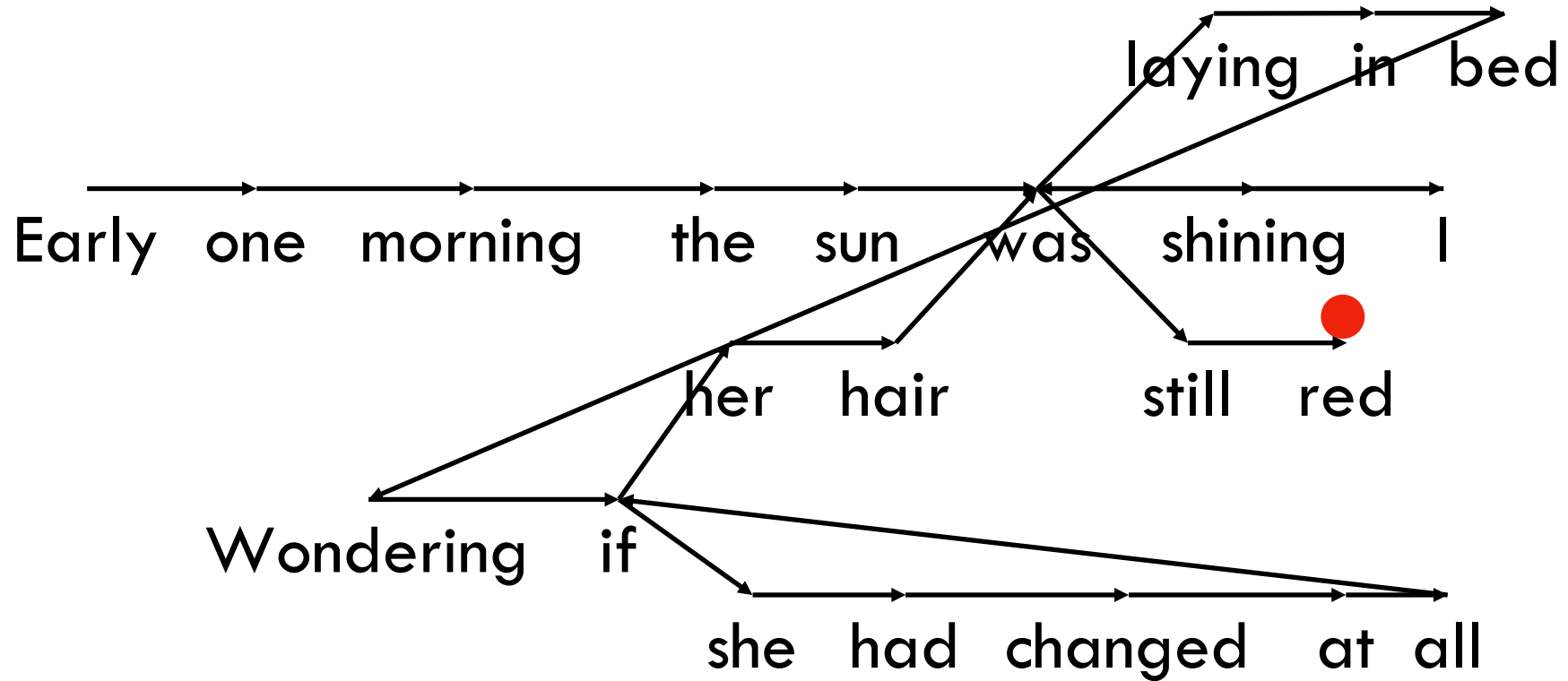
the sun was laying in bed



I was shining I was shining



I was shining I was shining I was still red



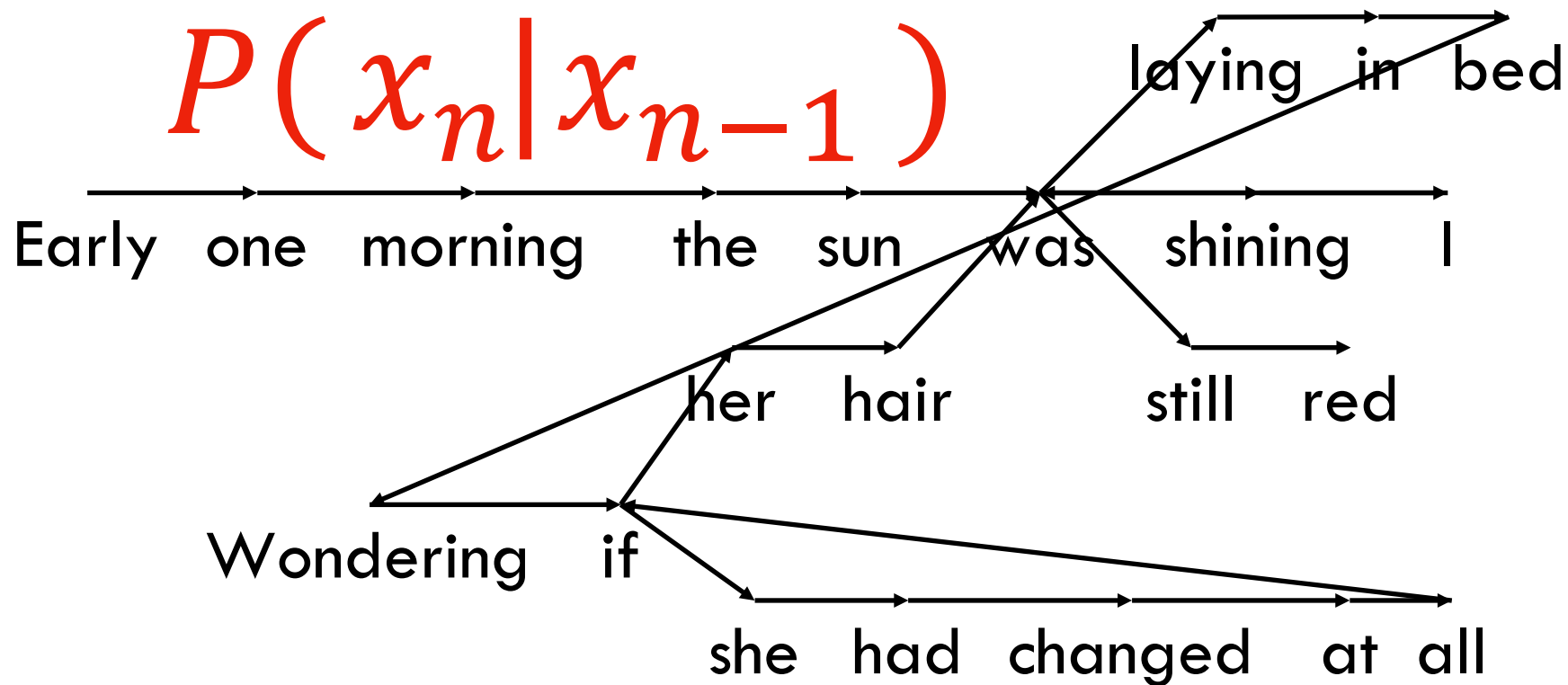
It's early one morning the sun was shining
I was happy at first
Wondering if time changed at all
if our hair was still the
the little they used our lives together
Sure was gonna be tough
They never did like Martin homemade dress
Purple hairbands wasn't big enough
And I was standing on the side of the road
feeling out my shoes
Heading out for the first time
Lord knows the road some times getting through
Tangled up in blue
She was married when we first met
Seem to be divorced
I helped her out of a jam I guess
But I used a little too much force
the dress that ran as far as we could
Admitted it our shoes
Spit up on a dark red light
Both agreeing I was lord
She turned around to look at me
As I was walking away
I found her eye on my shoulder
With most open hostility on the street
Tangled up in blue
I had a job in the great north woods
Working as a clerk for a year
But I never did the it all that much
And one day she so just fell
So I shifted over to New Orleans
Where I happened to be unemployed
Working for a while on a fishing boat
Wright outside of Louisiana
But all the while I was alone
The great was close behind
I seen a lot of women
But she never occupied my mind and I just grew
Tangled up in blue
She was working in a laptop place
And I stopped in for a beer
I just kept looking at the side of her face
In the spotlight in blue
And later on as the crowd thinned out
I a just about to be the stars
She was dancing there in back of my chair
Lead to me then I know your name
I muttered something underneath my breath
She studied the lines on my face
I must admit I felt a little uneasy
When she bent down to be the back of my shoes
Tangled up in blue
She'll be a burner on the street
And offered me a pipe
I thought you'd never say hello she said
You look like the stars type
Then she opened up a book of poems
And handed it to me
Written by an Italian poet
From the thirteenth century
And every one of them words ring true
And glowed like burning coal
Pouring off every page
Like it was written in my soul from the top
Tangled up in blue
I lived with them on Montague Street
In a basement down the stairs
There was music in the walls of night
And music in the air
Then he started this dealing with glass
And something inside of him died
She had to see everything she wanted
And broke up needs
And when finally the bottom fell out
I became withdrawn
The only thing I knew how to do
Was to keep on keeping on like a line that runs
Tangled up in blue
So now we going back again
I got to get to her somehow
All the people we used to know
There's an illusion in the rain
Some are mathematicians
Some are carpenters wives
Don't know how it all got started
I don't know what they're doing with their lives
But me in still on the road
Heading for another joint
We always did love the stars
We just see it from a different point of view
Tangled up in blue

she was standing on the side of my mind ...

side of my shoes heading out of my face ...

one of my chair said our lives together ...

$$P(x_n | x_{n-1})$$



Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still red

Early one morning

trigrams

Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still red

Early one morning
one morning the

trigrams

Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still red

Early one morning
one morning the
morning the sun

trigrams

Early one morning **the sun was** shining I was laying in bed
Wondering if she had changed at all if her hair was still red

Early one morning
one morning the
morning the sun
the sun was

trigrams

$$P(x_n | x_{n-1}, x_{n-2})$$

Early one → morning

one morning → the

morning the → sun

the sun → was

sun was → shining

was shining → I

shining I → was

I was → laying

...

Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still red

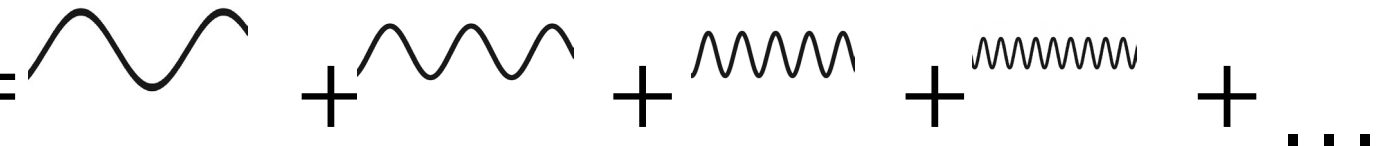
Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still red

Early one morning the sun was shining I was laying in **bed**
Wondering if she had changed at all if her **hair** was still **red**

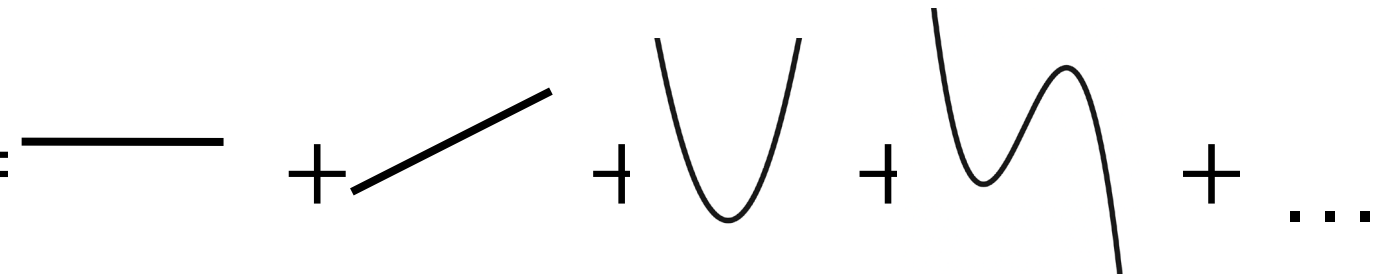
$$P(x_n | x_{n-1}, x_{n-2}, x_{n-3}, x_{n-4}, x_{n-5}, x_{n-6}, x_{n-7}, x_{n-8}, x_{n-9}, x_{n-10}, x_{n-11}, x_{n-12}, x_{n-13})$$

10^{70} combinations

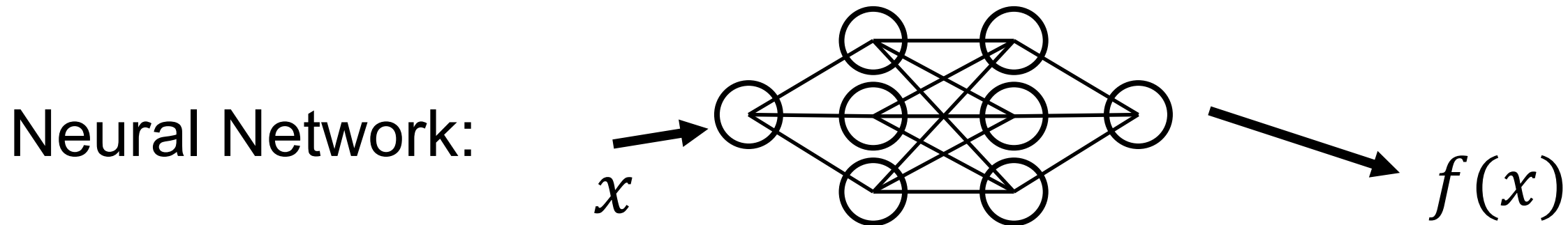
Function Approximation

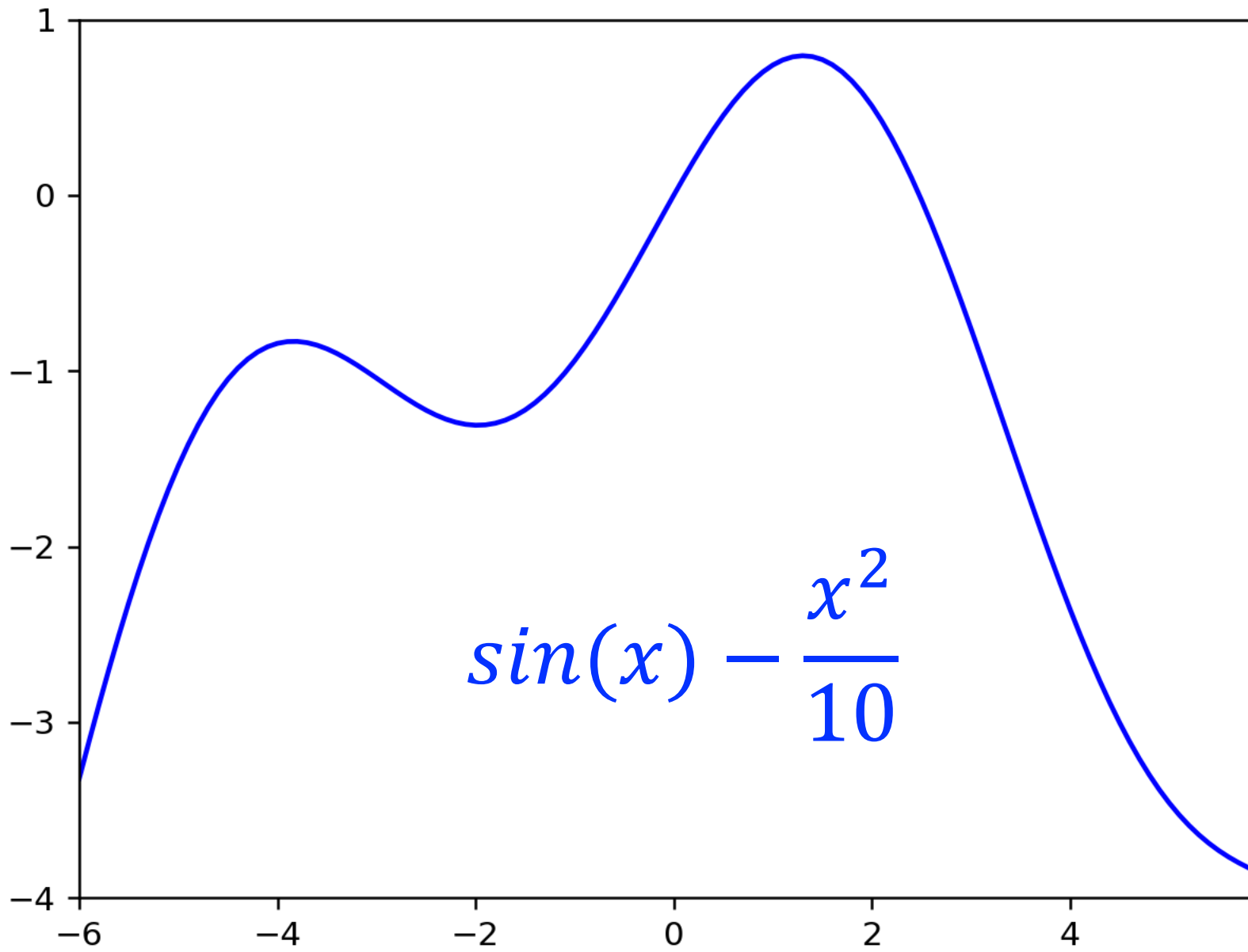
Fourier Series: $f(x) =$  $+ \dots$

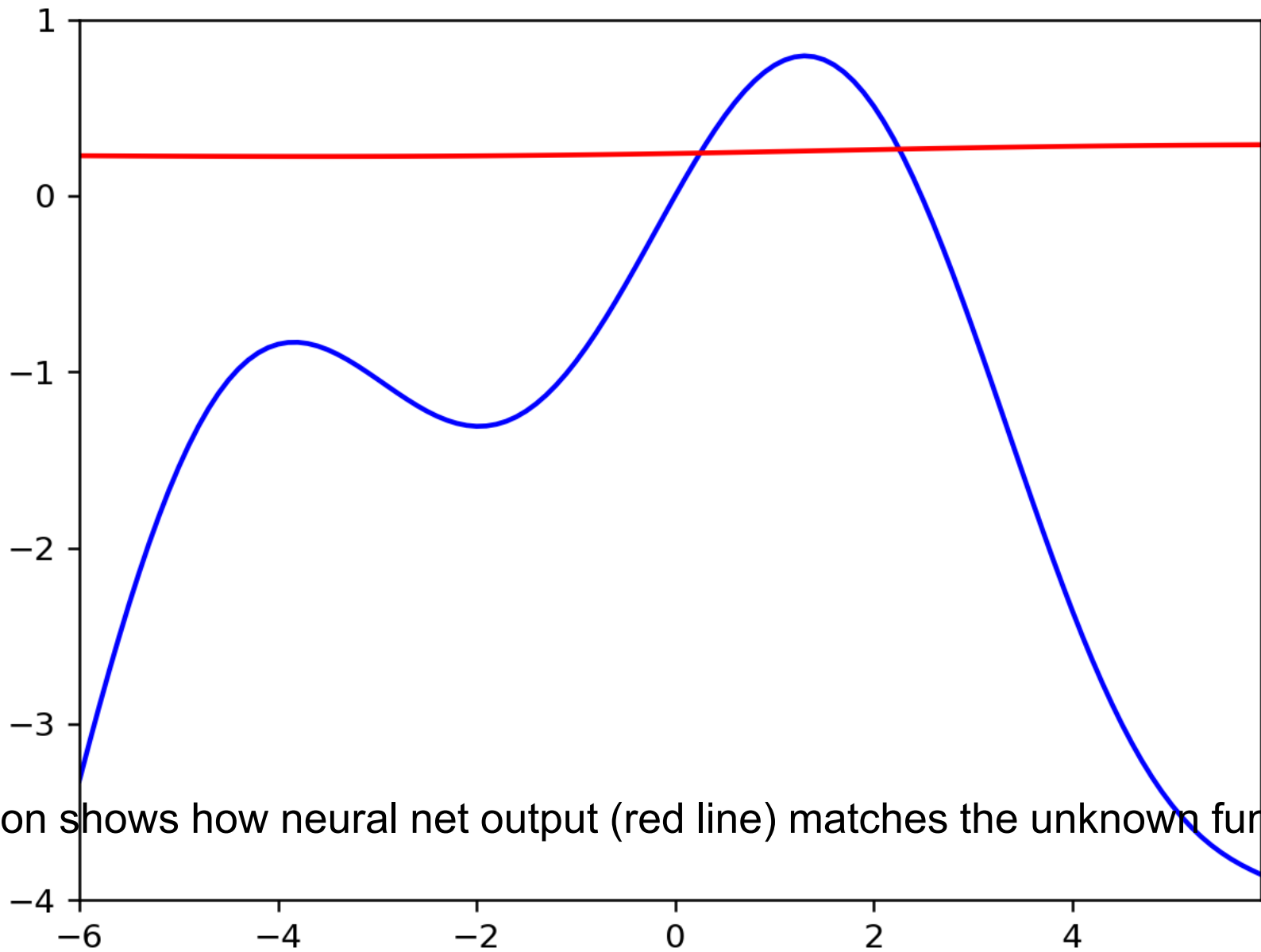
The diagram shows the function $f(x)$ as a sum of four sine waves with increasing frequencies, followed by an ellipsis. The first wave has the lowest frequency and amplitude, and each subsequent wave has a higher frequency and smaller amplitude.

Taylor Series: $f(x) =$  $+ \dots$

The diagram shows the function $f(x)$ as a sum of four terms: a horizontal line, a straight line with a positive slope, a parabola opening upwards, and a cubic curve, followed by an ellipsis.







Animation shows how neural net output (red line) matches the unknown function (blue line)

$$P(x_n | x_{n-1}, x_{n-2}, x_{n-3}, x_{n-4}, x_{n-5}, x_{n-6}, x_{n-7}, \dots)$$

Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still red

red



Early one morning the sun was shining I was laying in bed

Wondering if she had changed at all if her hair was still

red

neural network

Early one morning the sun was shining I was laying in bed wondering if she had changed at all if her hair was still

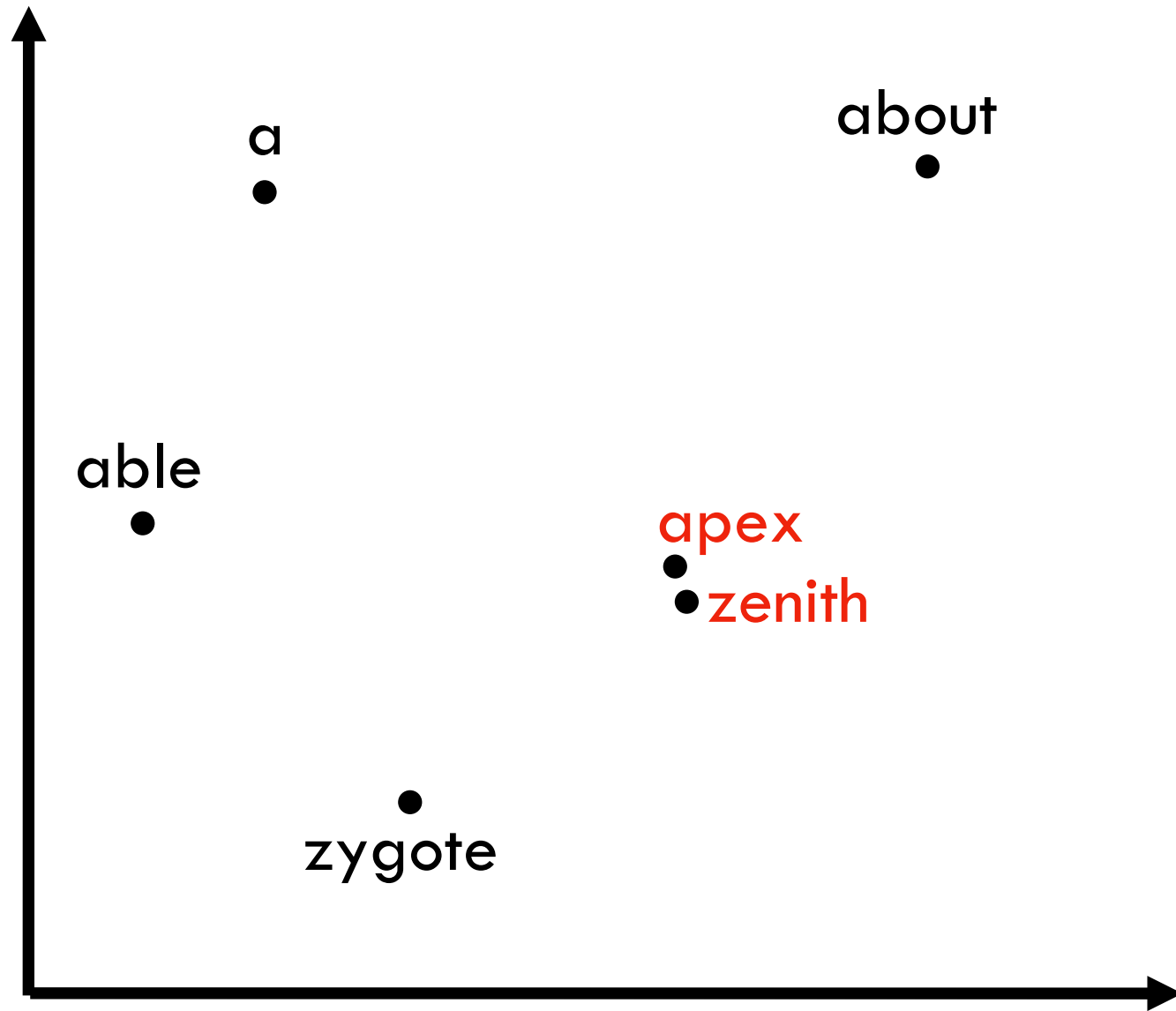
0	a
1	able
2	about
	⋮
39	apex
	⋮
56,356	zenith
	⋮
92,487	zygote

word2vec

[Collobert & Weston 2008; Mikolov et al. 2013]

house, where the professor lived without his wife and child; or so he said jokingly sometimes: "Here's where I live. My house." His daughter often added, without resentment, for the visitor's information, "It started out to be for me, but it's really his." And she might reach in to bring forth an inch-high table lamp with fluted shade, or a blue dish the size of her little fingernail, marked "Kitty" and half full of eternal milk; but she was sure to replace these, after they had been admired, pretty near exactly where they had been. The little house was very orderly, and just big enough for all it contained, though to some tastes the bric-à-brac in the parlor might seem excessive. The daughter's preference was for the store-bought gimmicks and appliances, the toasters and carpet sweepers of Lilliput, but she knew that most adult visitors would

Deep
Net

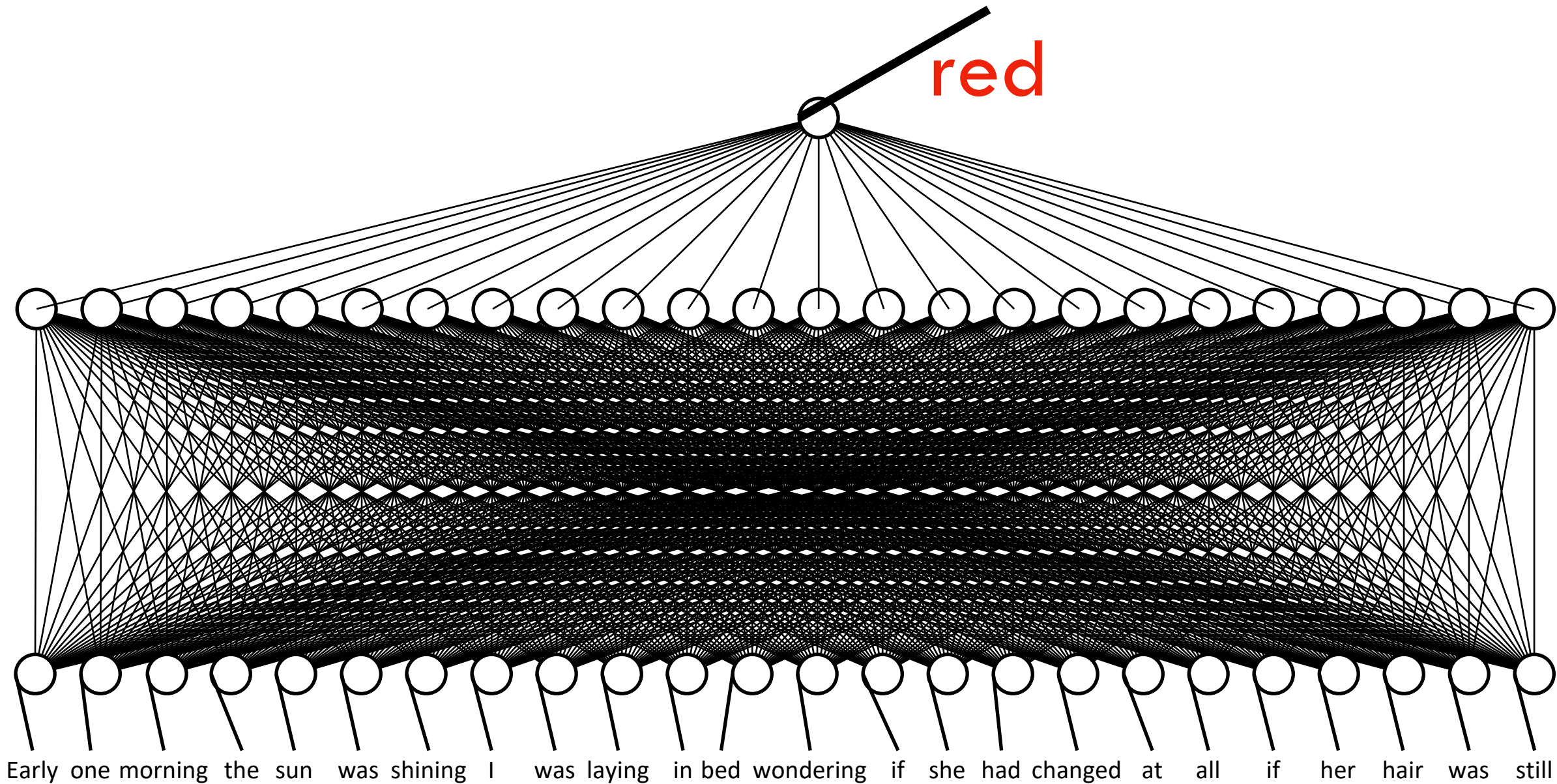


Word Embedding (e.g., word2Vec, GloVe)

red

neural network

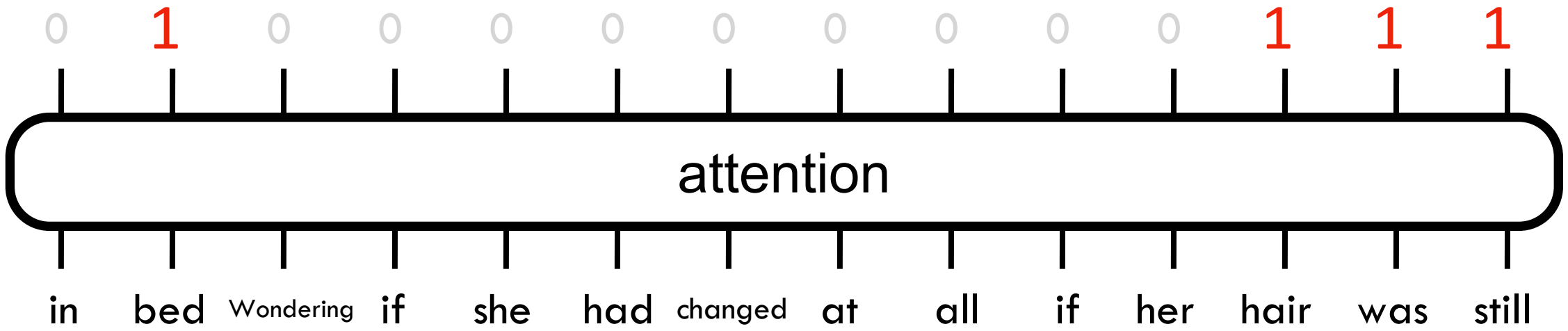
Early one morning the sun was shining I was laying in bed wondering if she had changed at all if her hair was still



Early one morning the sun was shining I was laying in bed
Wondering if she had changed at all if her hair was still ?

bed

hair was still red



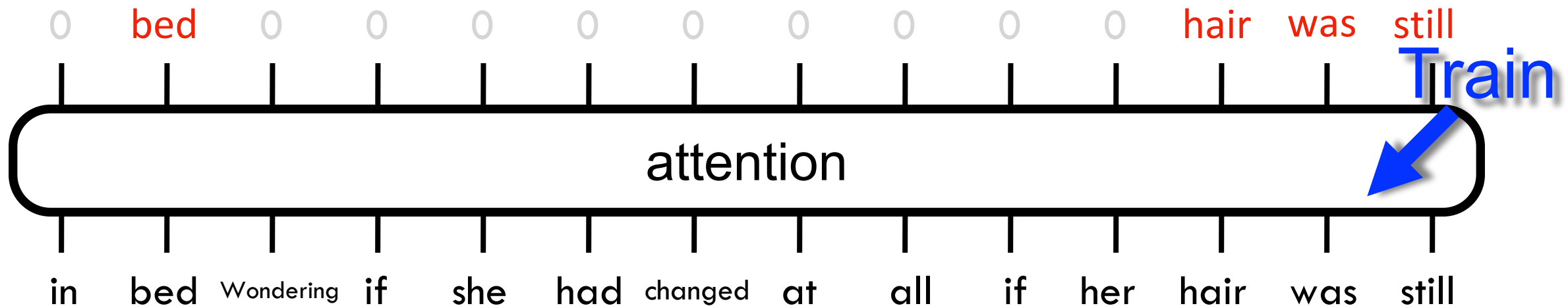
red

next word prediction

0 bed 0 0 0 0 0 0 0 0 0 hair was still

attention

in bed Wondering if she had changed at all if her hair was still



Two roads diverted in a yellow wood
And sorry I could not travel both
And be one traveler, long I stood
And looked down as far as I could
To where it bent in the undergrowth;

Robert Frost, *Road Not Taken*

slide from Steve Seitz's [video](#)

Train



red

next word prediction

0 bed 0 0 0 0 0 0 0 0 0 hair was still

attention

in bed Wondering if she had changed at all if her hair was still

Train



brown

next word prediction

0 bed 0 0 0 0 0 0 0 0 0 hair was still

attention

in bed Wondering if she had changed at all if her hair was still

Train

brown

next word prediction

0 bed 0 0 0 0 0 0 0 0 0 hair was still



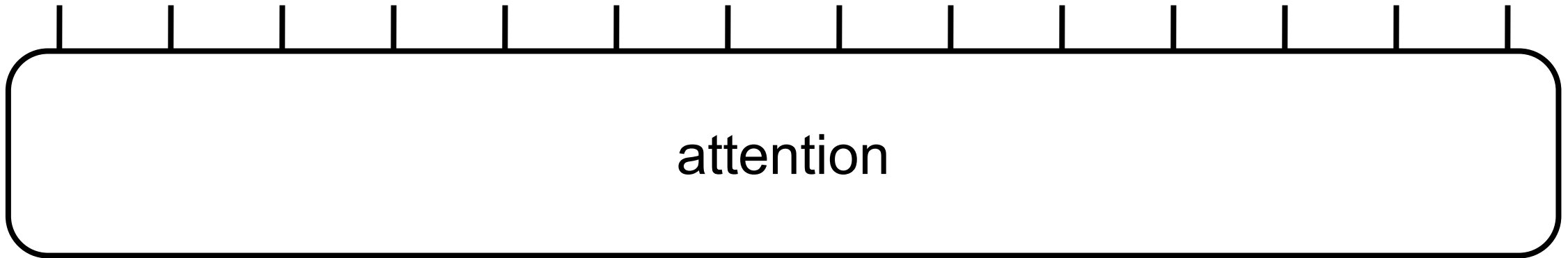
attention

in bed Wondering if she had changed at all if her hair was still

red

Transformer

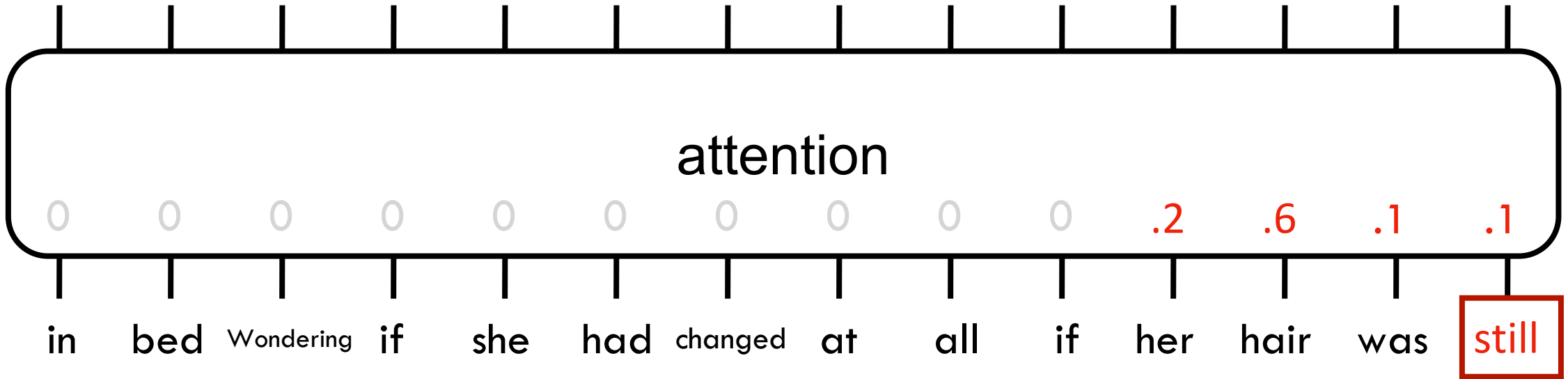
in bed Wondering if she had changed at all if her hair was still

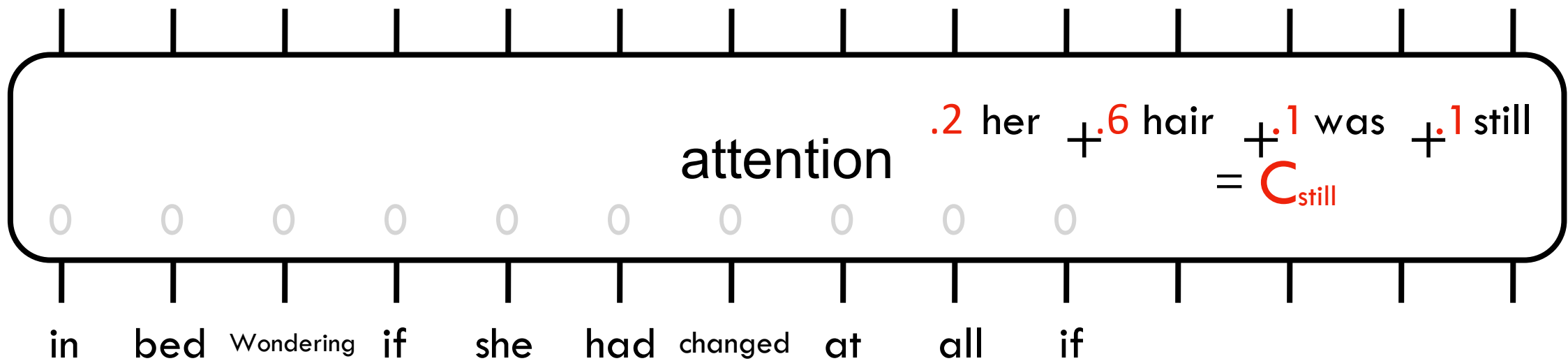


in bed Wondering if she had changed at all if her hair was still

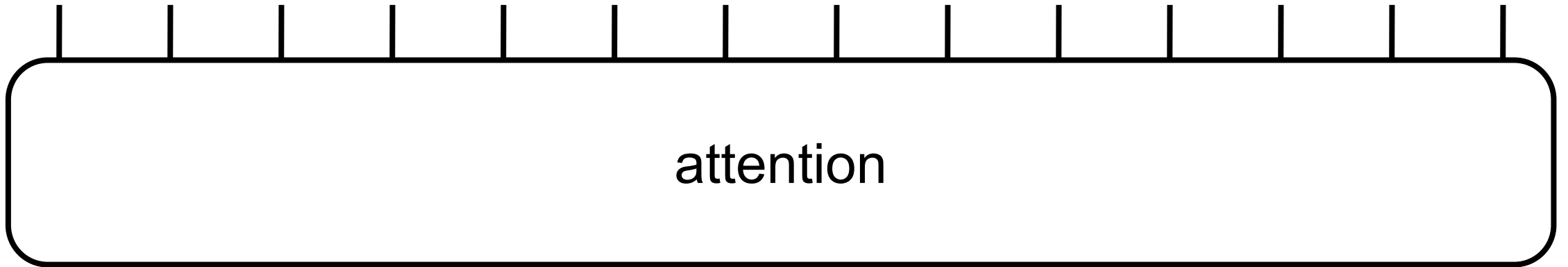
attention

in bed Wondering if she had changed at all if her hair was still



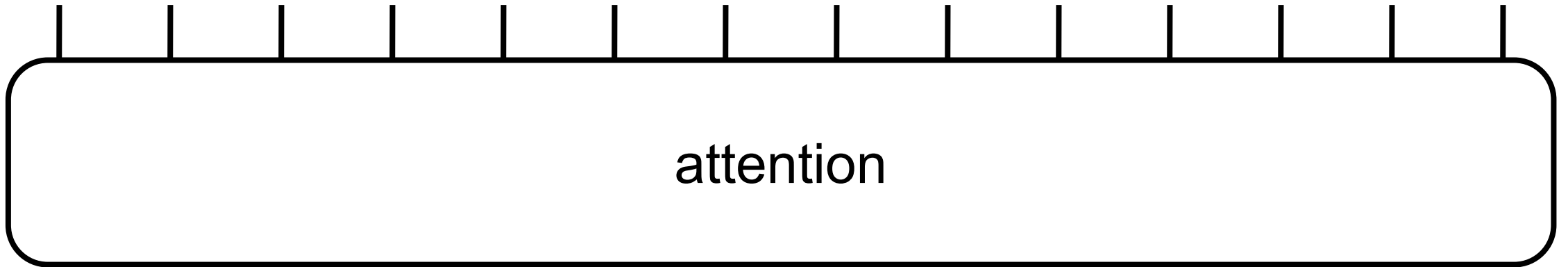


Cstill

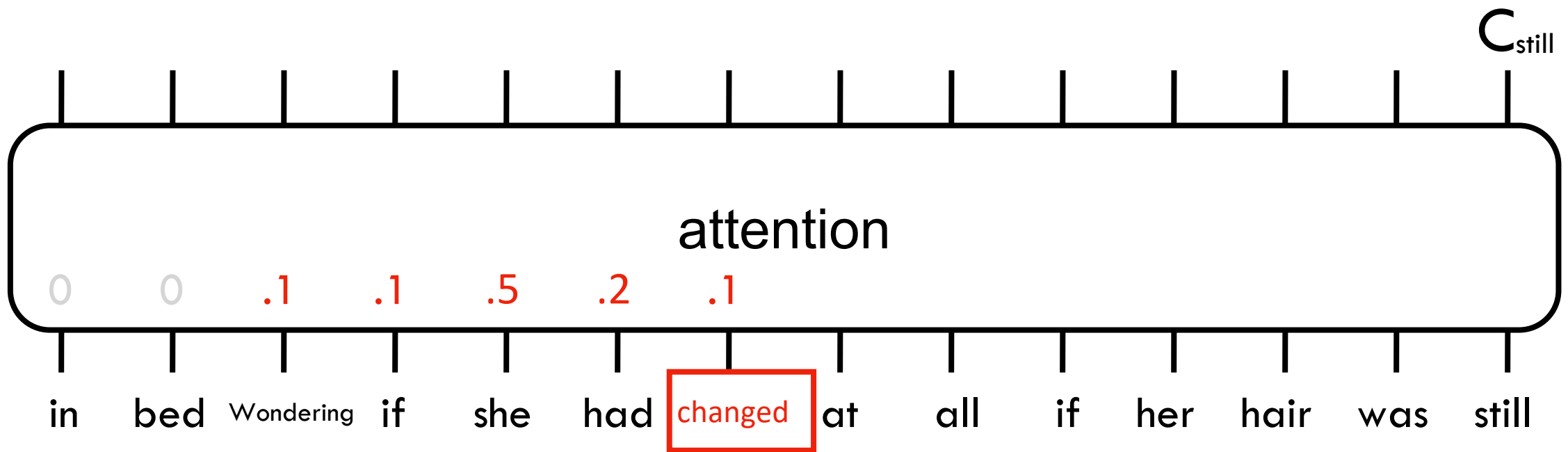


in bed Wondering if she had changed at all if her hair was still

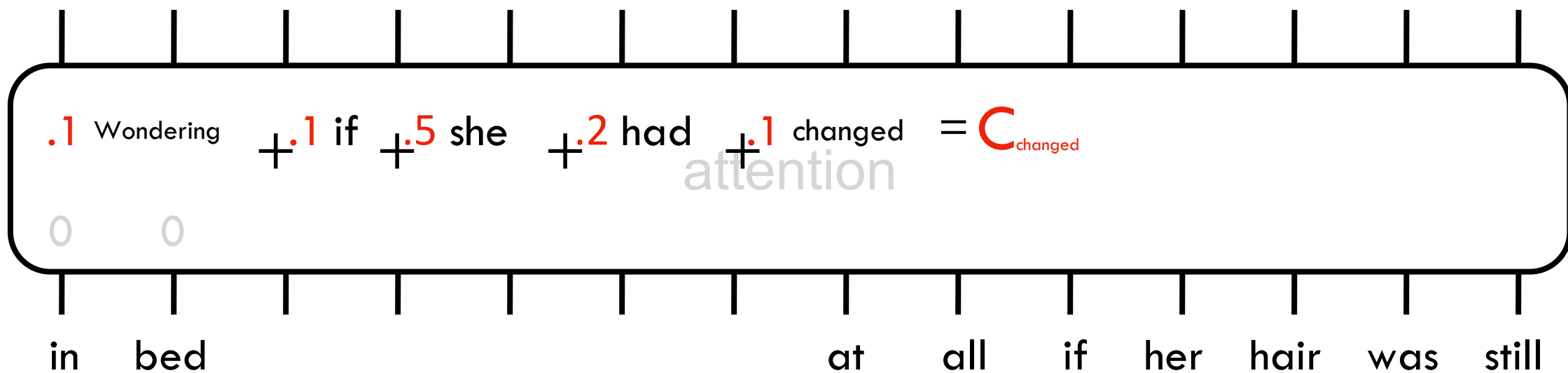
C still

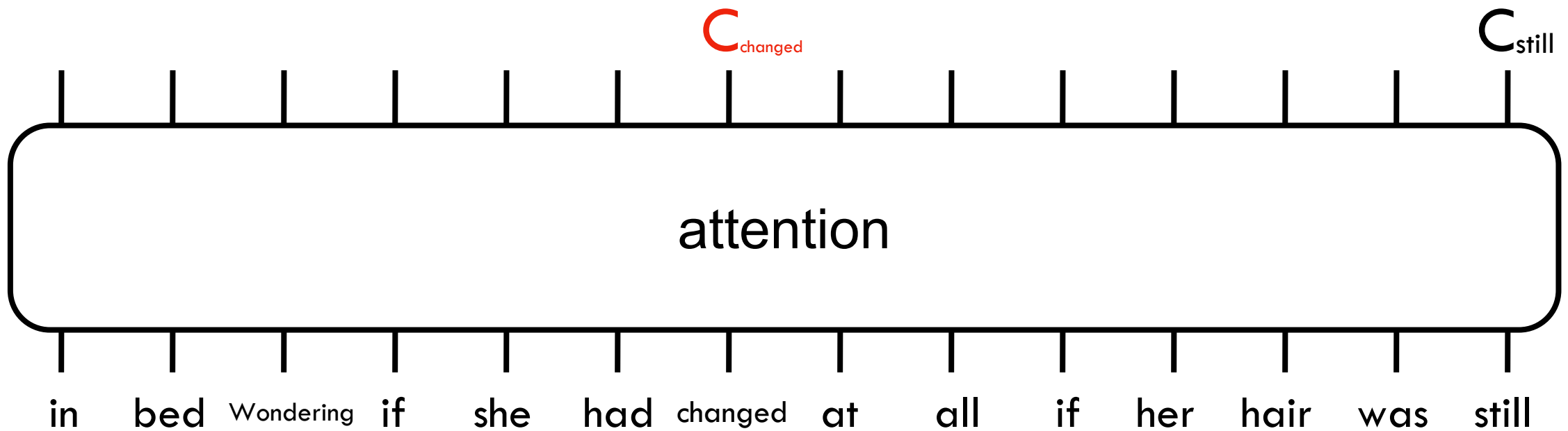


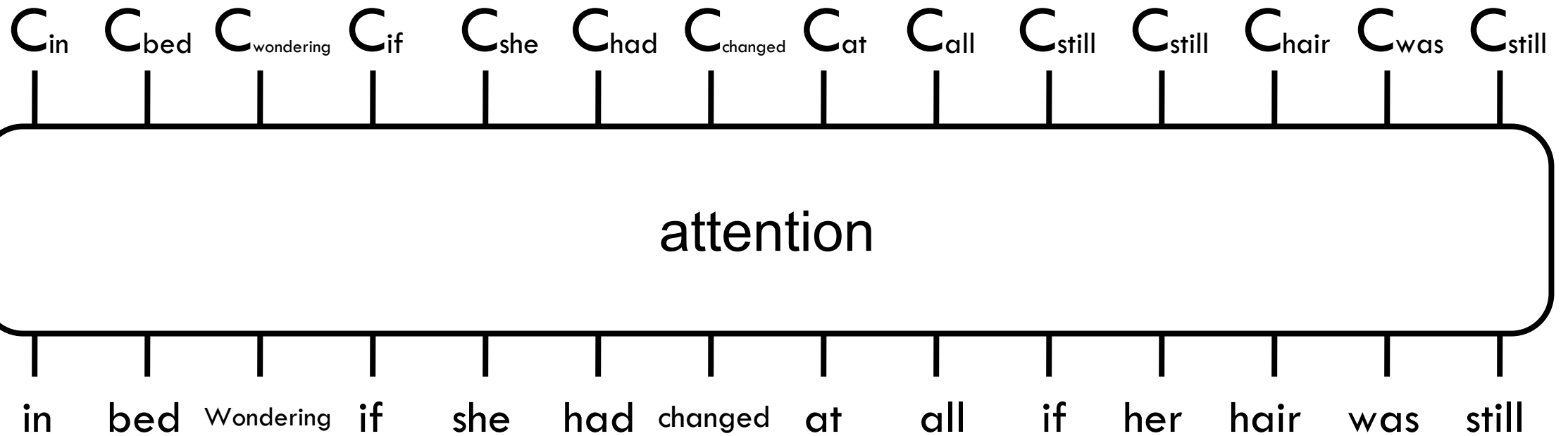
in bed Wondering if she had changed at all if her hair was still



C_{still}







prediction

C_{in} C_{bed} C_{wondering} C_{if} C_{she} C_{had} C_{changed} C_{at} C_{all} C_{still} C_{still} C_{hair} C_{was} C_{still}

attention

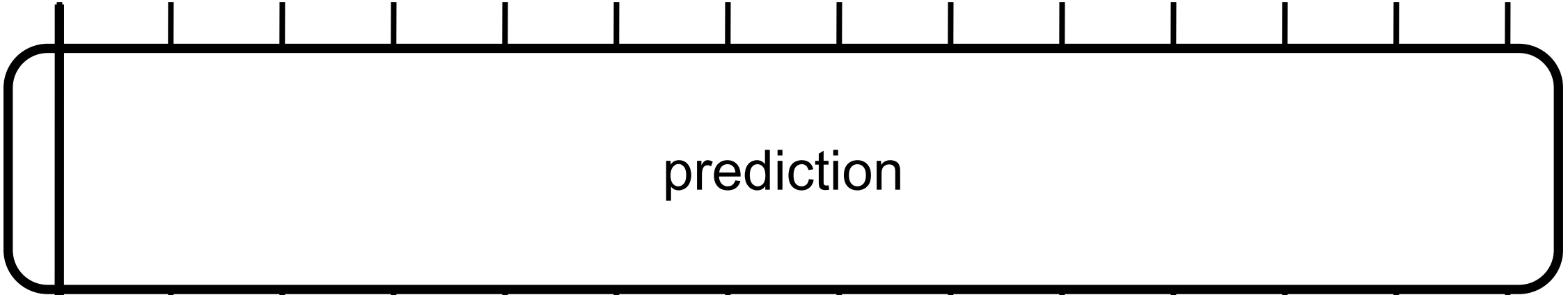
in bed Wondering if she had changed at all if her hair was still

prediction

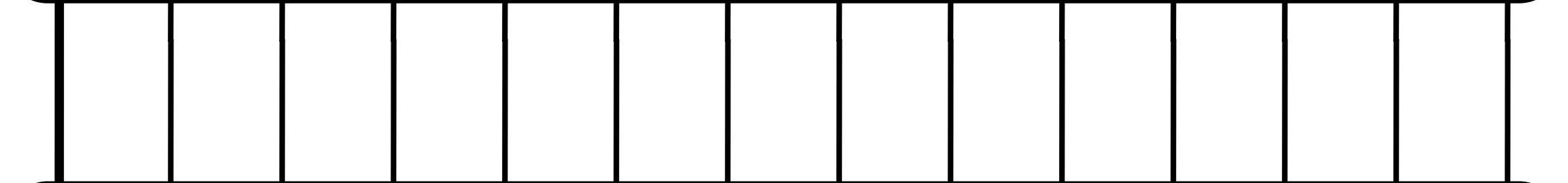
C_{in} C_{bed} C_{wondering} C_{if} C_{she} C_{had} C_{changed} C_{at} C_{all} C_{still} C_{still} C_{hair} C_{was} C_{still}
in bed Wondering if she had changed at all if her hair was still

attention

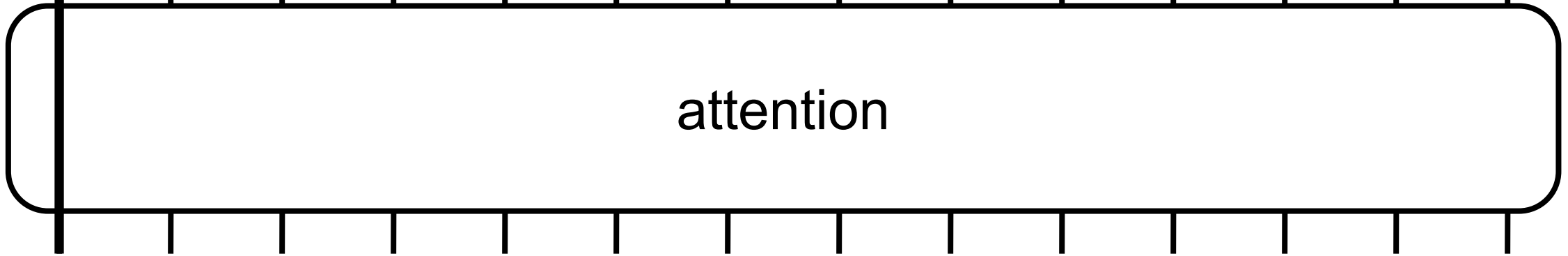
a



prediction



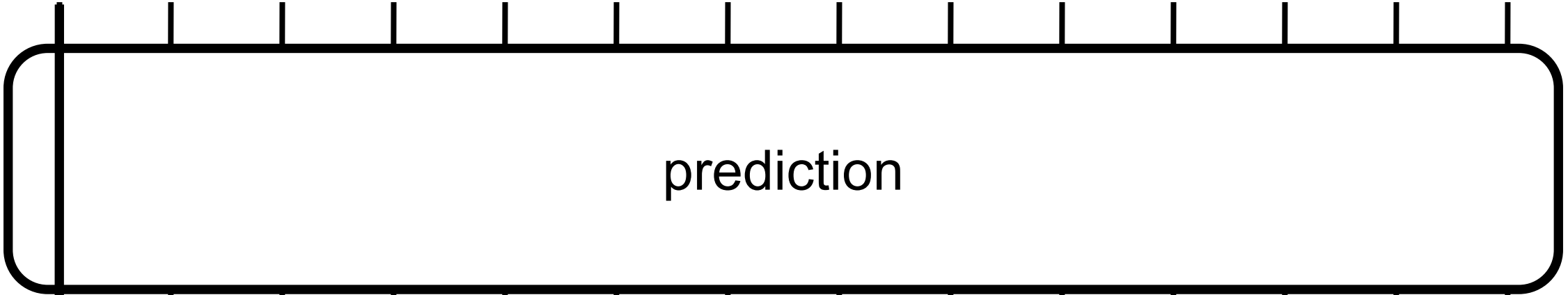
attention



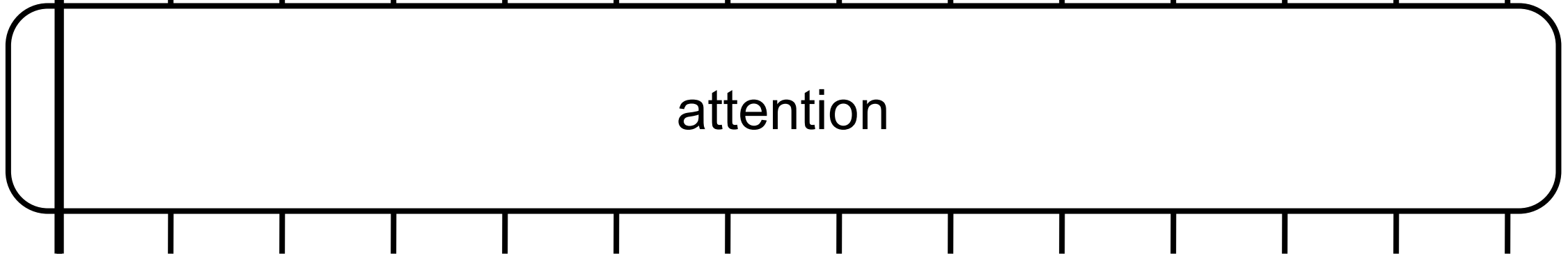
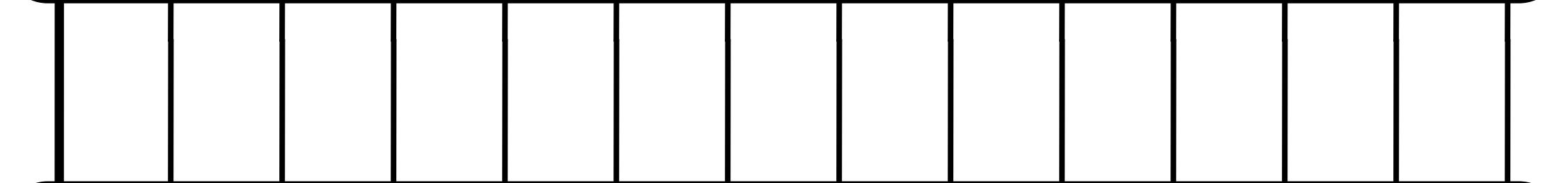
It's

a	the	looking	possible	getting
0.4	0.3	0.1	0.1	0.1

a

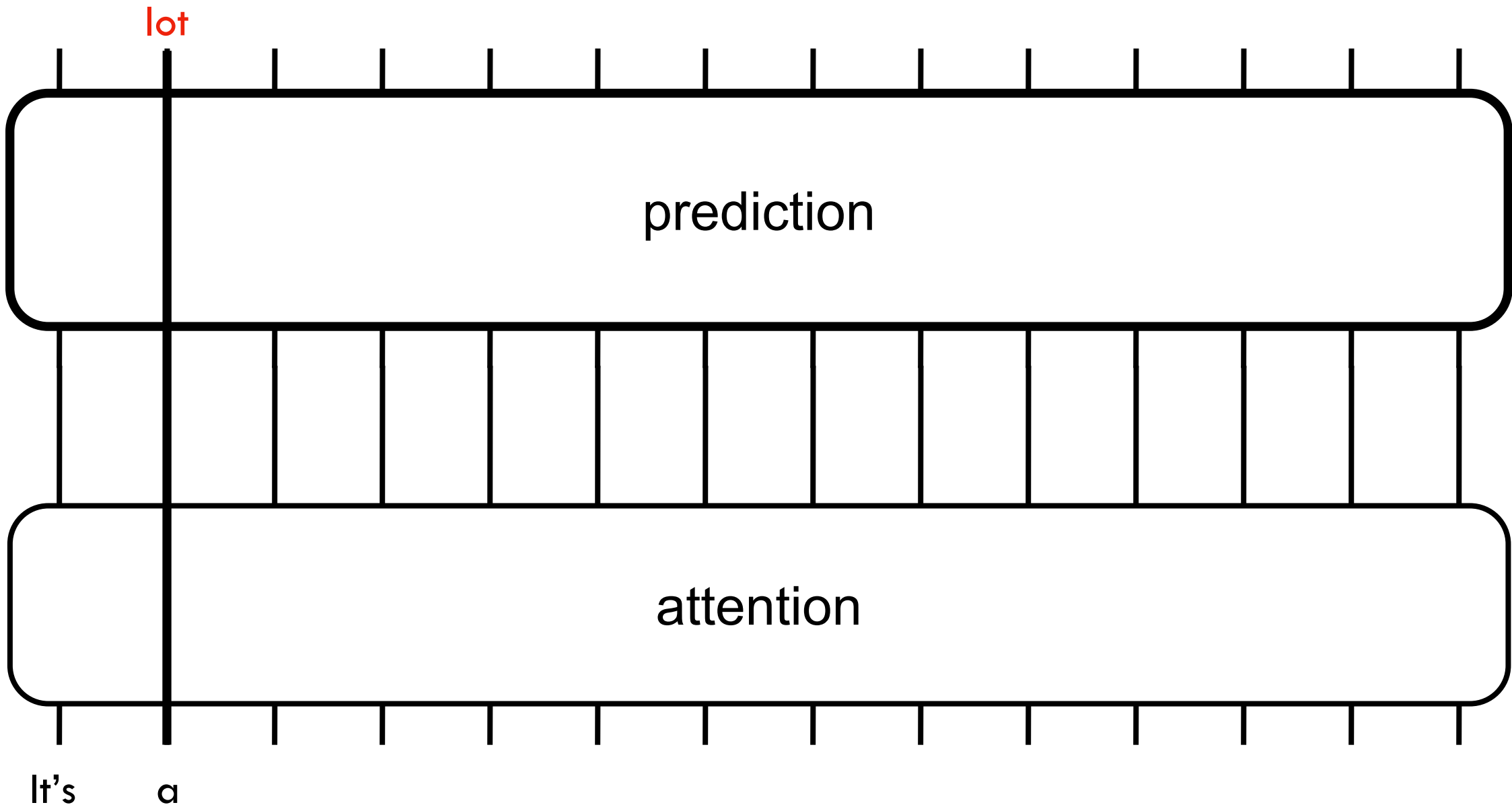


prediction



attention

It's



of

prediction

attention

It's

a

lot

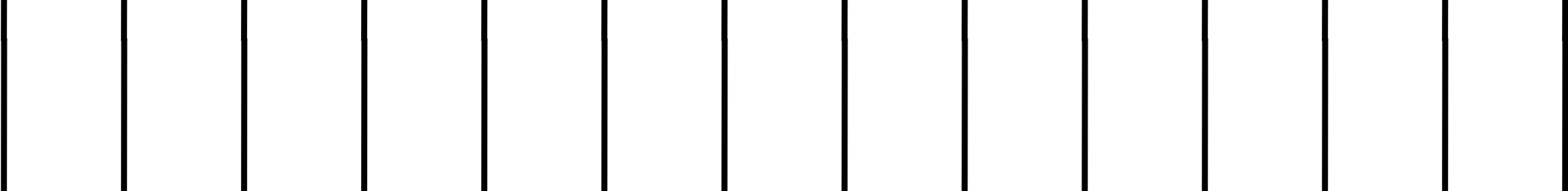
fun

prediction

attention

It's a lot of

prediction



attention

It's a lot of fun

Abraham

prediction

attention

The 16th
president was

The 16th President was ?

The capital of Zimbabwe is ?

Frank Zappa's middle name is ?

Napoleon was born on this date ?

The prime factorization of 19456721434 is ?

Queen Victoria's maiden name was ?

US per-capita income in 1957 was ?

The lat long coordinates of Rome are ?

prediction

attention

⋮

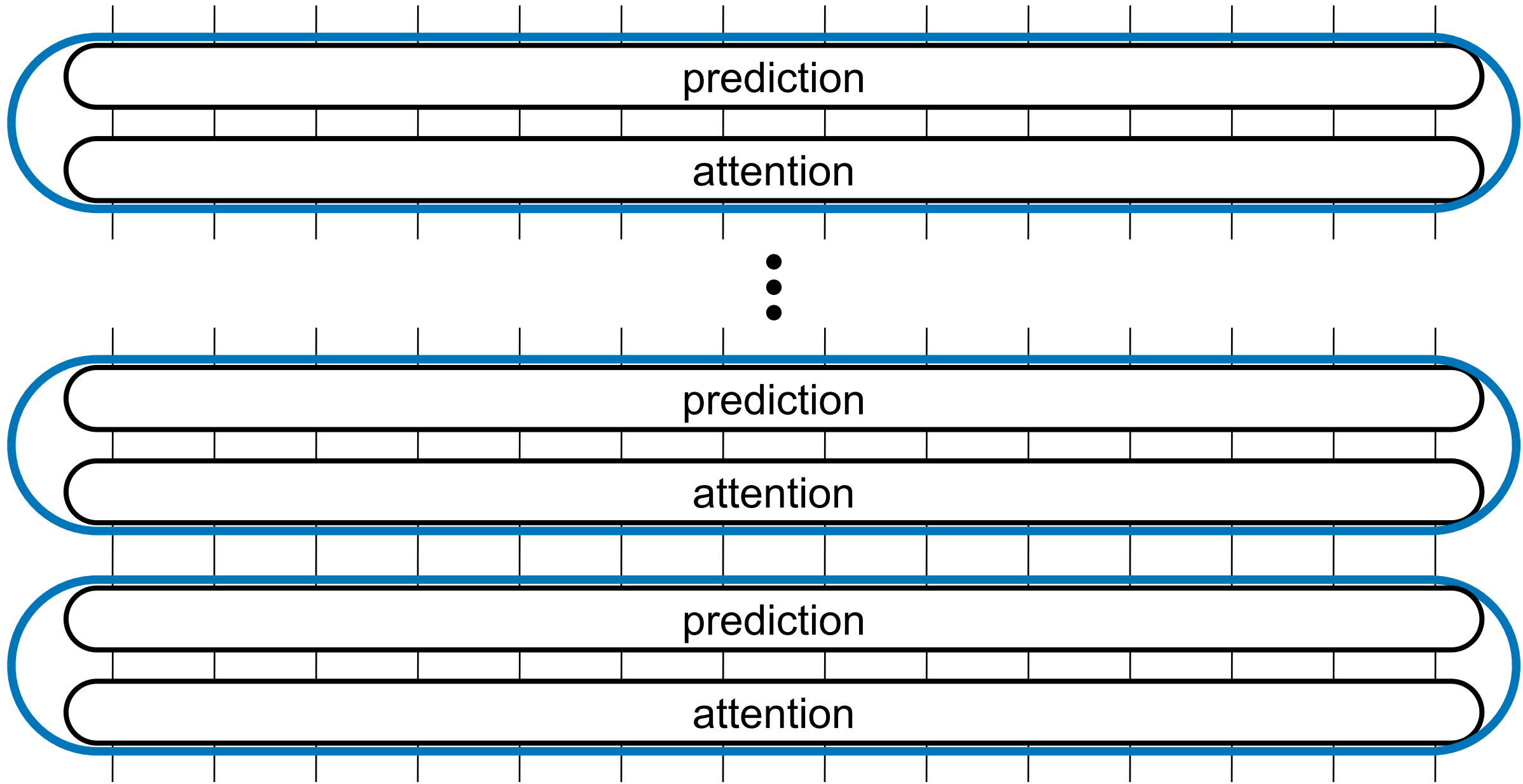
96 (GPT-3) **118** (PaIm)

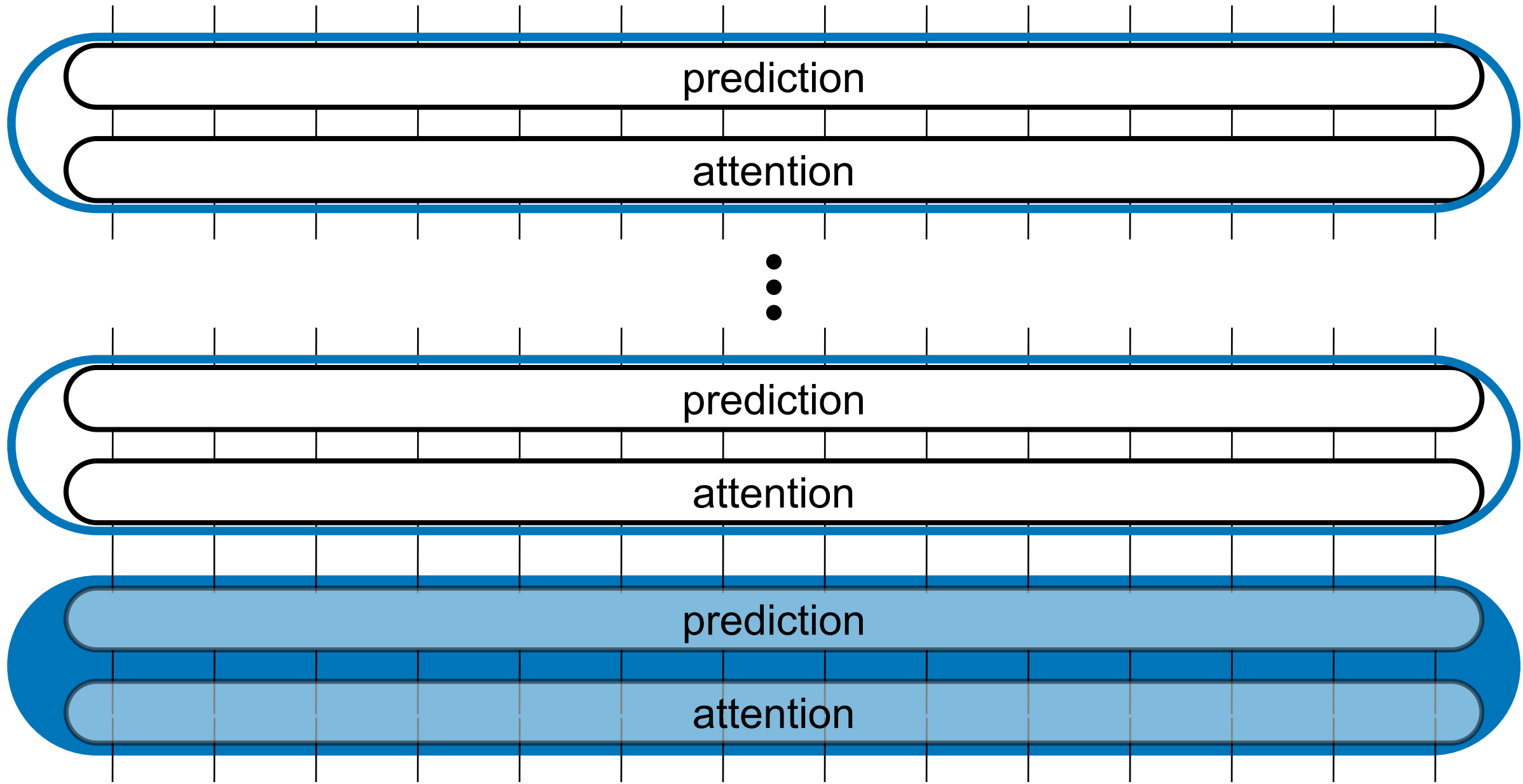
prediction

attention

prediction

attention





Syntax

Semantics

prediction

attention

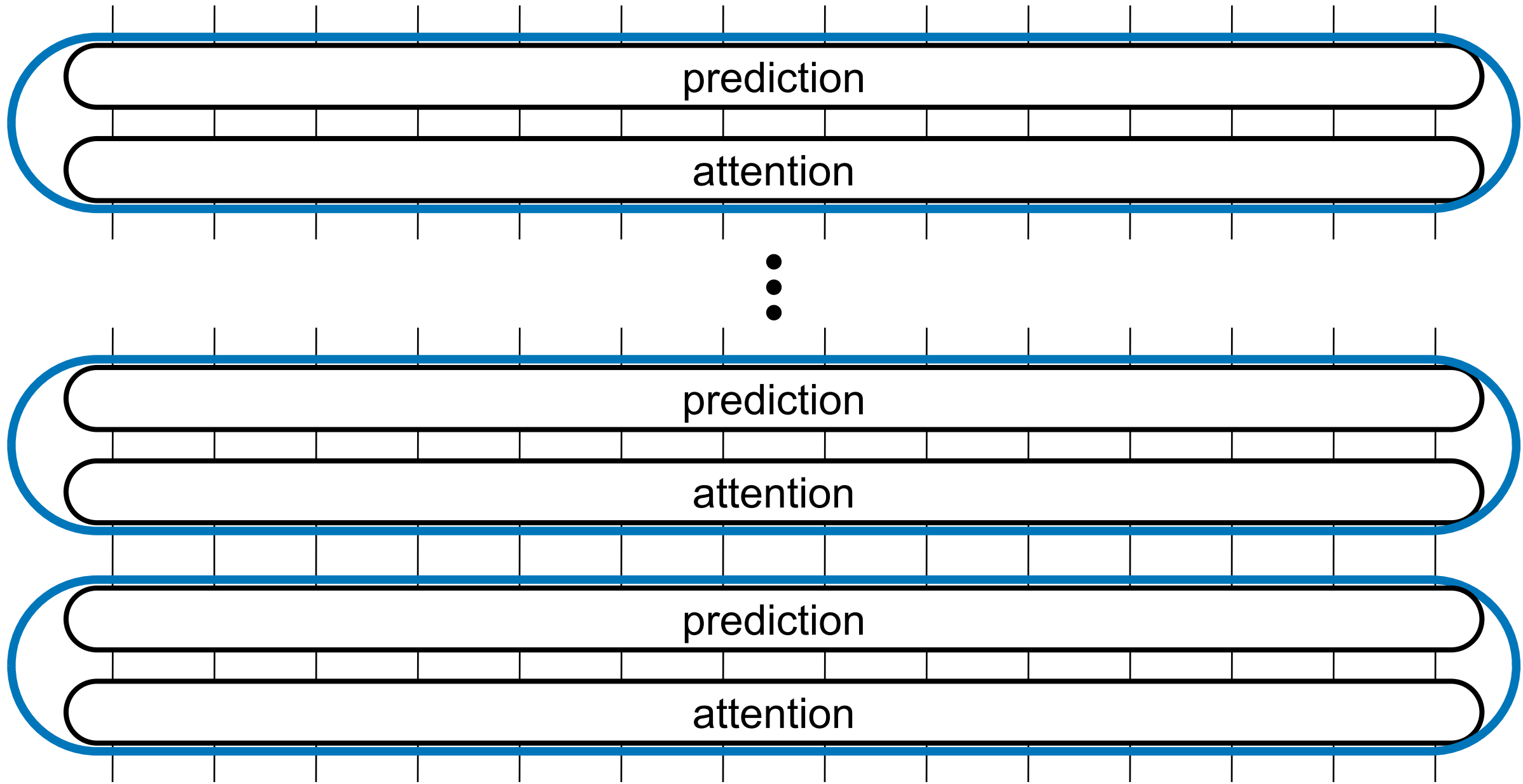
⋮

prediction

attention

prediction

attention



**How much data
to train?**

All of it...

All text on the internet?

Is that legal?

AI & Ethics!

All text on the internet?

Is that legal?

AI & Ethics!



REUTERS®

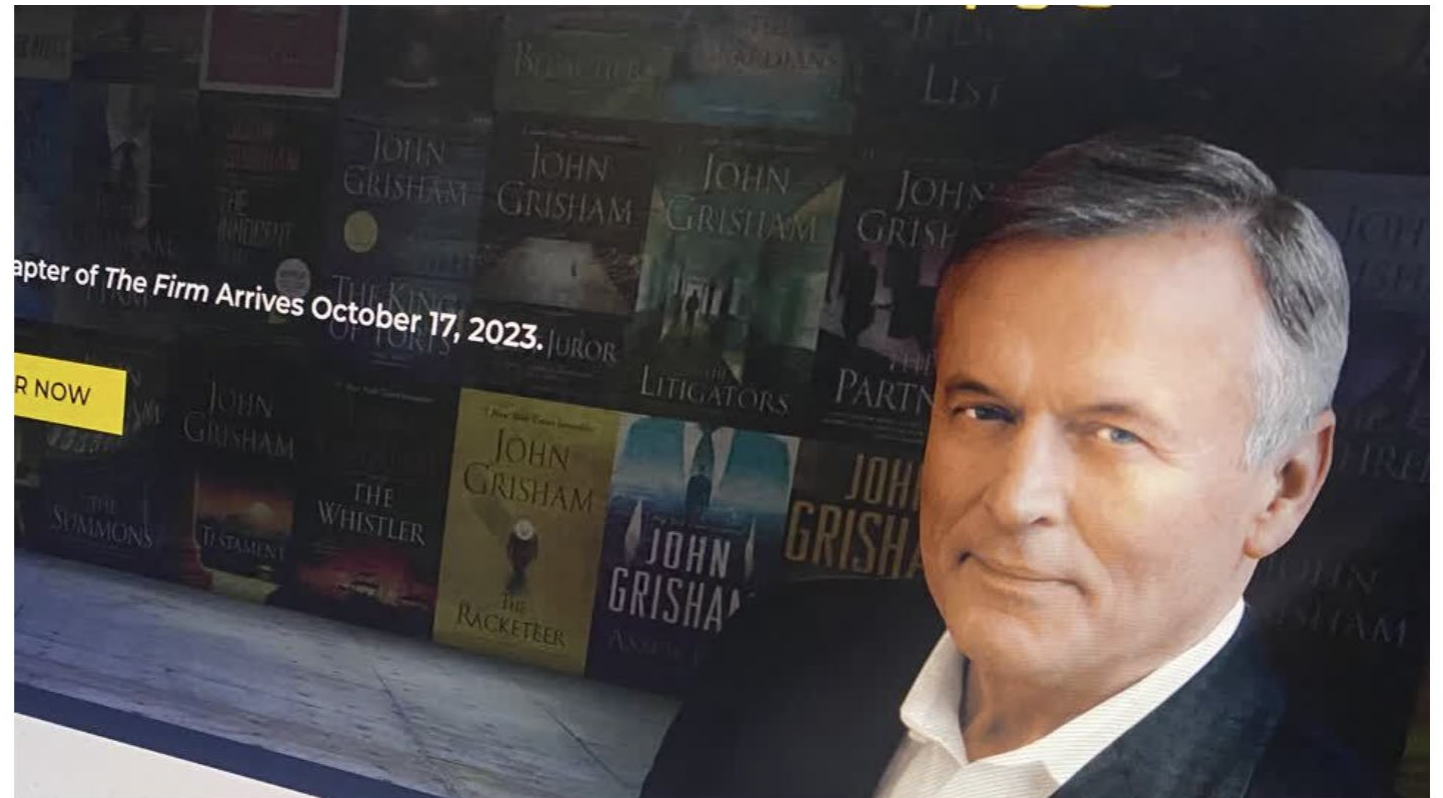
World ▾ Business ▾ Markets ▾ Sustainability ▾ Legal ▾ Breakingviews ▾ Technology ▾ Investic

Litigation | Copyright | Litigation | Technology | Intellectual Property

John Grisham, other top US authors sue OpenAI over copyrights

By Blake Brittain

September 21, 2023 6:34 AM EDT · Updated 7 months ago



All text on the internet?

Is that legal?

AI & Ethics!

December 27, 2023

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

All text on the internet?

Is that legal?

AI & Ethics!



Training the 175 billion parameters
of GPT-3 on
“all text on the internet”
on a single GPU or computer
would take 355 years
and \$4,600,000

How long did it take OpenAI to train GPT-3?

a month

What did OpenAI train on?

GPT-3 training data^{[1]:9}

Dataset	# tokens	Proportion within training
Common Crawl	410 billion	60%
WebText2	19 billion	22%
Books1	12 billion	8%
Books2	55 billion	8%
Wikipedia	3 billion	3%

Source: Wikipedia

What about GPT-4?

- 1 trillion parameters
- Sam Altman stated that the cost of training GPT-4 was more than **\$100 million**.

What about GPT-4?

- 1 trillion parameters
- Sam Altman stated that the cost of training GPT-4 was more than **\$100 million**.

Why are the lawsuits so costly?

- We don't know how to “untrain” neural networks.
- “Unlearning” is an exciting research area!

The 16th President was

The capital of Zimbabwe is

Frank Zappa's middle name is

Napoleon was born on this date

The prime factorization of 19456721434 is

Queen Victoria's maiden name was

US per-capita income in 1957 was

The lat long coordinates of Rome are

The 16th President was Abraham Lincoln

The capital of Zimbabwe is Harare

Frank Zappa's middle name is Vincent

Napoleon was born on this date 1769

The prime factorization of 19456721434 is $2 \times 3 \times 3 \times 17$

Queen Victoria's maiden name was Alexandrina Victoria

US per-capita income in 1957 was \$2,974

The lat long coordinates of Rome are 41.894722, 12.48

a haiku about a turtle and a stop sign

a haiku about a turtle and a stop sign

Turtle at the stop sign
waiting for the perfect moment
to cross the busy road

translate “tangled up in blue” to 5 languages

translate “tangled up in blue” to 5 languages

1. Spanish: Enredado en azul
2. French: Emmêlé dans le bleu
3. Italian: Intrecciato nel blu
4. German: Verwickelt in blau
5. Japanese: 青い糸に絡まって

the number after the number after 37 is 74

I'm heading north. I turn right. Then I turn left.
I'm now heading south

the recipe for chocolate guacamole pancakes

Ingredients:

- 1 ripe avocado, mashed
- 1/4 cup cocoa powder
- 1/4 cup all-purpose flour
- 1 teaspoon baking powder
- 1/4 teaspoon salt
- 1/2 cup milk
- 1 egg
- 1 tablespoon vegetable oil
- 1/4 cup chocolate chips

Directions:

1. In a medium bowl, combine the avocado, cocoa powder, flour, baking powder, and salt.
2. In a separate bowl, whisk together the milk, egg, and vegetable oil.
3. Add the wet ingredients to the dry ingredients and mix until well combined.
4. Fold in the chocolate chips.
5. Heat a large skillet over medium heat.
6. Scoop 1/4 cup batter onto the skillet for each pancake.
7. Cook for 2-3 minutes per side, or until the pancakes are golden brown.
8. Serve with your favorite toppings.



For more details:

<https://jalammar.github.io/illustrated-transformer/>
<https://jalammar.github.io/illustrated-gpt2/>

slide from Steve Seitz's [video](#)

a pattern of characters that looks like a star

• ○ •
○ ○ ○
• ○ •

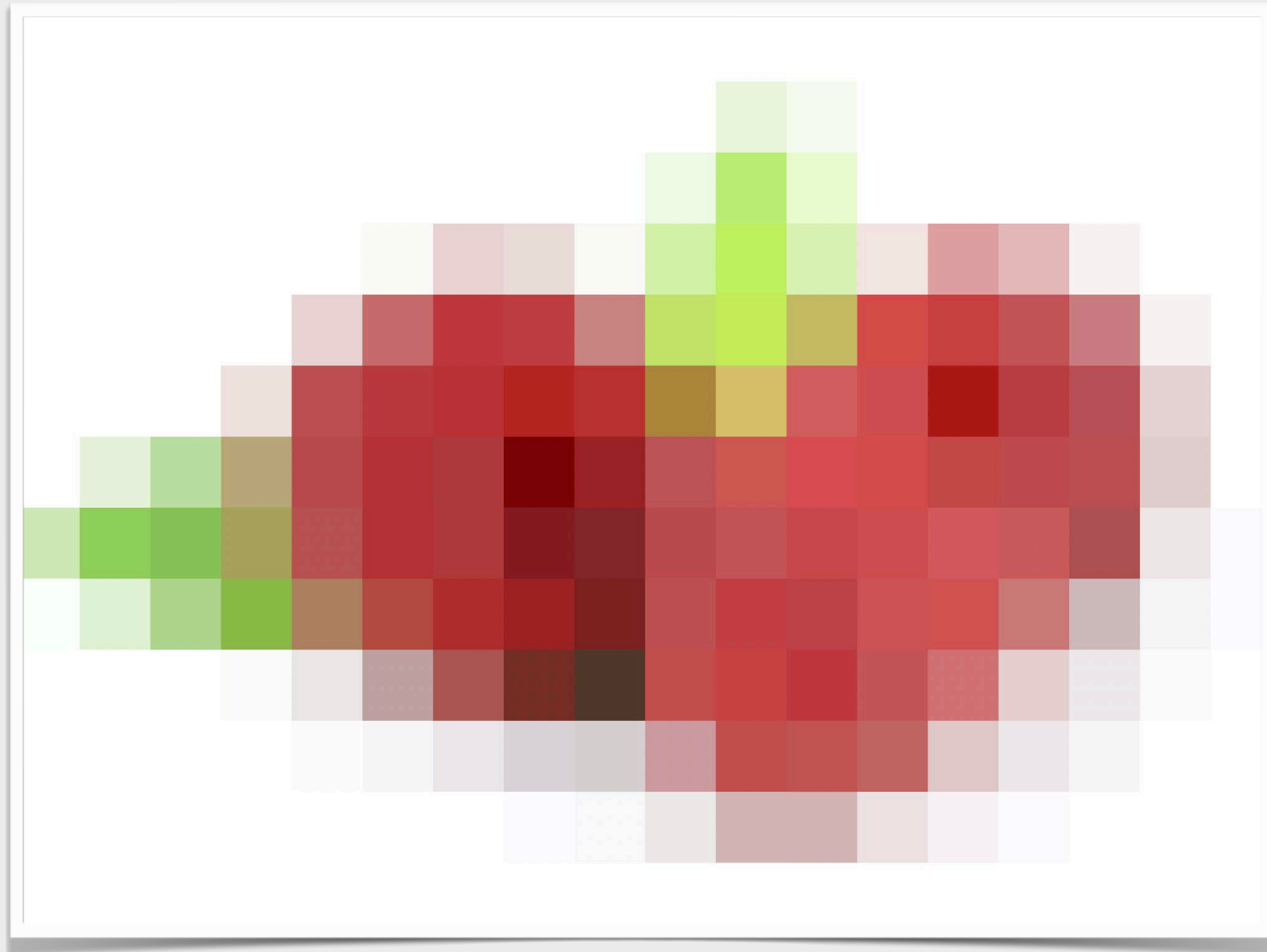
a pattern of characters that looks like a vertical line

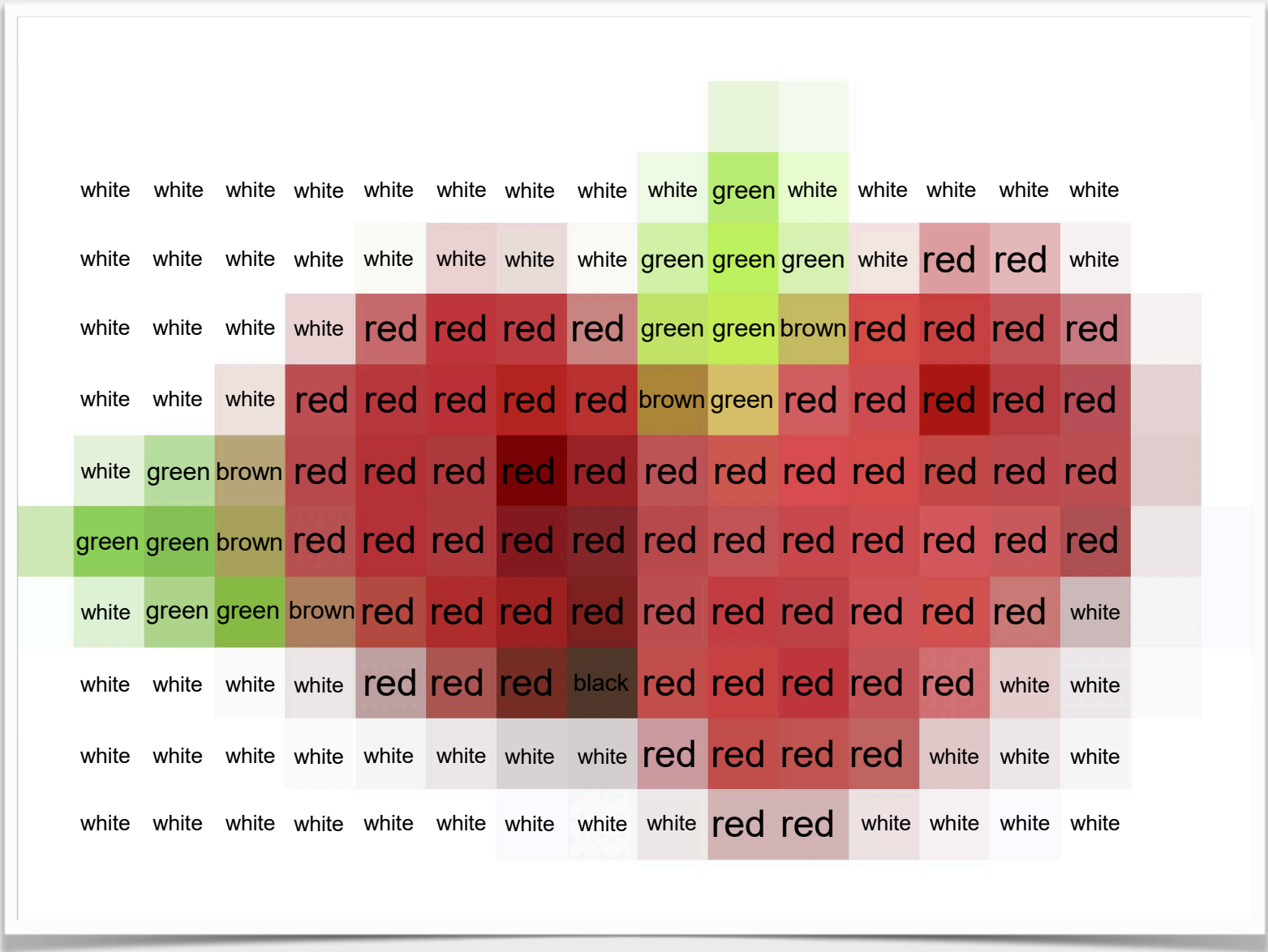
• ○ •
• ○ •
• ○ •
• ○ •
• ○ •

a pattern of characters that looks like a triangle

• ○ •
• ○ ○ •
• ○ ○ ○ •
• ○ ○ ○ ○ •
• ○ ○ ○ ○ ○ •



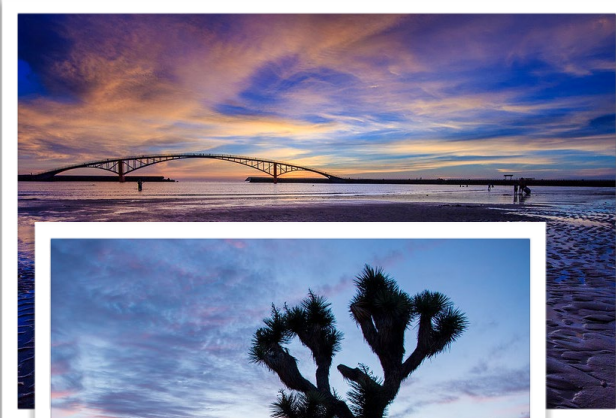
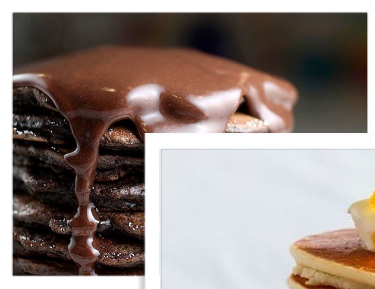




white white white white white white white white white green white white white white white
white white white white white white white white green green green white red red white
white white white white red red red red green green brown red red red red
white white white red red red red red brown green red red red red red red
white green brown red red red red red red red red red red red red red red
green green brown red red red red red red red red red red red red red red
white green green brown red red red red red red red red red red red white
white white white white red red red black red red red red red white white
white white white white white white white white red red red red white white white
white white white white white white white white white white red red white white white white

← (255,0,0)

white white white white white white white white white green white white white white white
white white white white white white white white green green green white red red white
white white white white red red red red green green brown red red red red
white white white red red red red red brown green red red red red red
white green brown red red red red red red red red red red red red red
green green brown red red red red red red red red red red red red red
white green green brown red red red red red red red red red red red white
white white white white red red red black red red red red red white white
white white white white white white white white red red red red white white white
white white white white white white white white white red red white white white white



raspberries

pancakes

sunsets

1 Billion

white

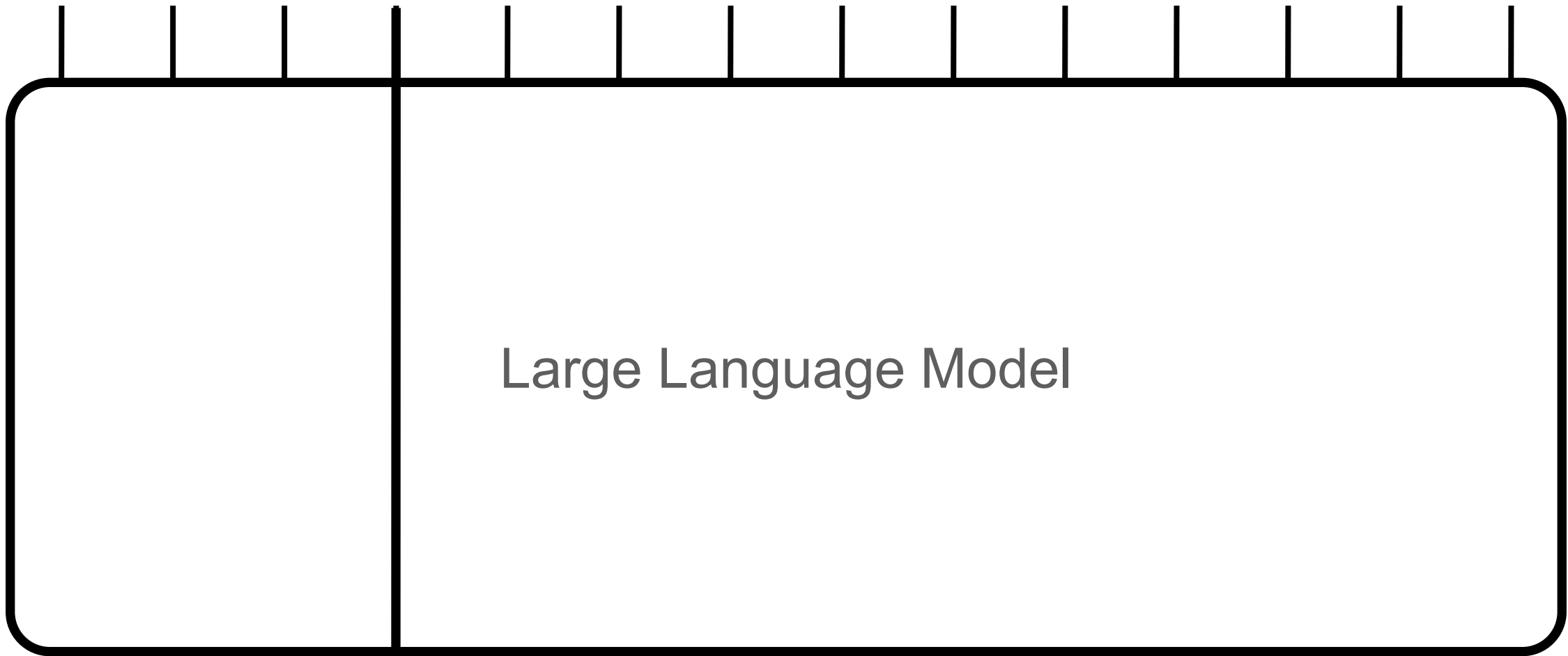
Large Language Model

A

image

raspberry

white



Large Language Model

A

image white

raspberry

red

Large Language Model

A

image white white

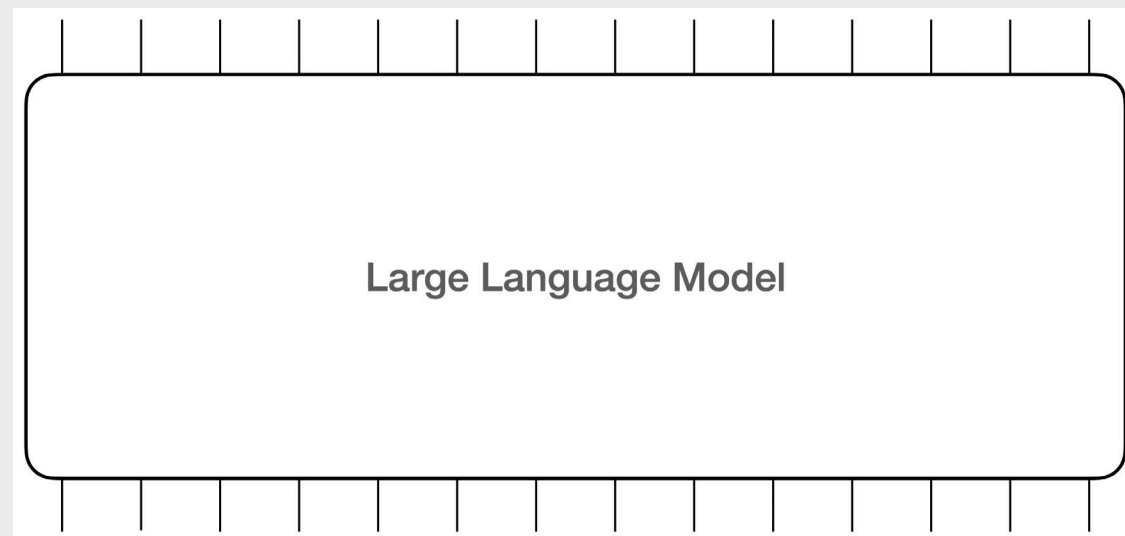
raspberry

Large Language Model

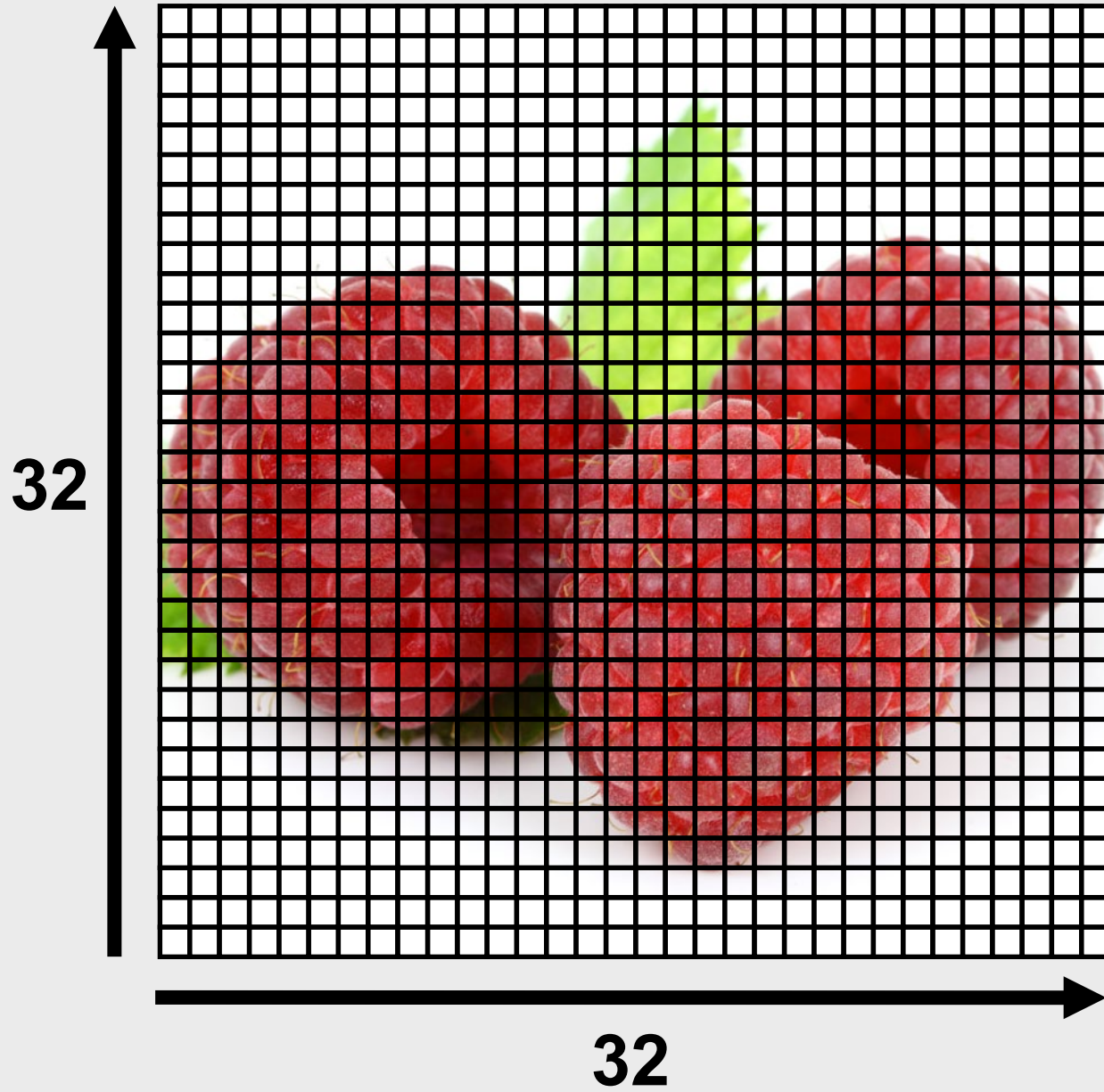
A raspberry image white white red red red white white green green green white



1,000,000s of pixels



1,000s of words



32 × 32
= 1024


0
0
0
0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 2 5 5 2 2 0 0 0 0
0 0 0 0 0 0 2 2 2 2 2 2 2 0 0 1 1 1 1 1 0 0 0 2 5 2 2 5 5 2 0 0 0
0 0 0 2 2 2 2 2 2 2 2 2 5 2 2 2 1 1 1 1 1 2 2 5 5 5 5 2 5 2 5 2 2 0
0 2 2 2 2 2 2 2 2 2 5 5 2 2 1 1 1 1 1 2 2 5 5 5 5 2 5 2 2 2 2 2 2
0 2 2 2 2 5 2 2 5 2 5 5 2 2 2 1 1 1 2 2 2 5 5 3 3 2 2 2 2 2 2 2 6
8 2 2 2 5 5 5 5 5 3 3 3 5 5 2 2 2 5 5 2 2 5 5 2 3 3 3 5 5 2 2 2 6
8 2 2 2 2 2 2 2 2 3 3 3 2 5 2 2 2 2 2 2 5 5 2 2 3 3 2 2 2 2 2 6
8 2 2 2 2 2 2 2 2 3 3 3 2 2 2 2 2 2 2 2 5 2 2 5 5 5 5 2 2 2 5 2 6
8 2 2 2 2 2 2 2 2 2 3 3 2 2 3 2 5 2 2 2 5 2 2 2 2 2 2 2 2 2 2 2 2 0
7 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 5 5 2 5 3 3 3 0 0
7 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 5 5 5 2 5 2 2 2 5 2 2 5 3 0 0 0 0
7 7 2 2 2 2 2 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 5 5 2 2 2 0 0 0 0 0
0 0 0 2 2 2 2 5 5 2 2 2 9 2 5 2 2 5 2 2 2 2 2 2 5 2 2 2 2 0 0 0 0 0 0
0 0 0 0 0 2 2 2 2 2 9 9 9 9 5 5 5 2 2 2 5 2 2 5 2 2 2 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 9 9 9 9 0 2 2 5 5 2 2 2 2 2 5 2 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 5 2 2 5 5 2 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0
0
0
0 0

Visual
words

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1 

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1 1

squirrel reaching for a nut

1	1	1	1	1	1	1	1	1	1	1	1	1	1	6	6	6	1	1	6	6	4	4	4	9	9	9	9	9	9
1	1	1	1	1	7	7	1	1	1	1	1	1	1	1	1	6													

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1 6

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1 6

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1 6

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1 6 6 6 6 6 6 6 6 4 1 4 9 9 9 9 9 9
1 1 7 1 1 1 7 1 1 1 1 1 1 1 1 6 6 6 6 6 6 6 6 6 4 1 4 9 9 9 9 9 9
1 1 7 1 1 1 1 1 1 1 1 1 1 1 2 6 2 6 6 2 2 6 2 9 9 9 9 9 9 9 9 9 9
1 1 1 7 7 1 1 1 1 1 1 1 1 1 6 6 6 6 2 5 2 2 4 9 9 9 9 9 9 9 9 9 9
1 1 1 7 1 1 1 1 1 1 1 1 1 2 6 6 6 2 5 2 2 0 9 9 9 9 9 9 9 9 9 9
1 1 1 1 1 1 2 2 2 2 2 2 1 2 6 6 6 2 6 2 6 2 0 9 9 9 9 9 9 9 9 9 9
2 1 1 1 1 2 2 1 2 2 2 2 1 2 6 6 6 6 6 2 6 4 9 9 9 9 9 9 9 8 8 8
2 2 1 1 1 1 1 1 1 1 1 1 1 6 6 2 6 6 6 4 4 9 9 9 9 9 9 9 8 8 8
1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 0 0 0 0 0 4 9 9 9 9 9 9 9 8 8 8
1 1 1 1 1 1 1 1 1 7 1 5 2 2 2 0 0 0 0 0 4 9 9 9 9 9 9 9 8 8 8
1 1 1 1 1 1 1 1 1 1 5 2 2 2 2 0 0 0 0 4 4 6 9 9 9 9 9 9 9 9 9
1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 5 2 0 5 4 6 9 9 9 9 9 9 9 9 9 9
1 1 1 1 7 1 1 1 1 1 2 2 6 6 5 5 2 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 1 1 1 7 2 2 2 1 1 2 2 6 5 5 0 2 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 1 1 7 2 2 2 2 1 1 2 6 6 5 5 0 0 4 4 4 9 9 0 0 0 0 9 9 9 9 9
1 1 0 1 2 1 2 1 1 1 1 1 1 1 0 0 0 0 4 0 0 0 4 0 0 0 0 9 9 9 9 8
1 1 1 1 1 1 1 1 1 1 1 7 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 9 9 9 9 8
1 1 1 1 1 1 1 1 1 7 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 9 9 9 8 8
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 4 4 9 9 9 8 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 4 4 9 9 5 5 9
1 1 1 1 1 7 7 7 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 4 4 9 9 5 4 3
1 1 1 1 1 7 7 3 3 3 3 3 3 3 3 4 0 4 0 0 4 0 0 0 4 4 9 9 5 4 4
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 5 0 0 0 0 0 0 0 4 9 9 9 4 3 3
3 3 3 3 3 3 3 3 3 3 3 4 4 4 3 3 3 5 3 0 4 4 4 4 4 4 9 9 9 4 3 8
3 8 3 4 4 4 4 9 3 9 4 8 8
3 3 8 3 3 3 3 3 3 3 3 3 3 3 8 3 8 8 3 3 8 3 8 4 4 8 4 8 3 8 3 3 3
8 8 3 8 8 3 8 3 3 3 8 3 8 3 3 3 3 3 8 3 8 4 8 8 8 3 3 3 3 3 3
3 3 8 3 3 3 3 3 3 8 8 8 8 3 3 8 8 3 3 8 8 8 8 8 8 8 8 3 8 3 8
3 3 8 8 3 3 3 8 3 3 3 3 3 3 3 3 8 8 3 3 8 3 3 3 8 3 8 8 8 8 3 8
3 3 8 8 3 3 3 8 8 3 8 3 3 3 3 3 3 3 3 3 3 3 3 3 8 3 8 8 8 8 8 8
3 3 8 3 3 3 3 8 3 8 8 3 3 3 3 3 3 3 3 3 3 3 3 3 3 8 8 8 8 8 8

squirrel reaching for a nut

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 1 1 6 6 4 4 4 9 9 9 9 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 1 1 6 6 6 6 6 6 6 6 4 1 4 9 9 9 9 9 9
1 1 7 1 1 1 7 1 1 1 1 1 1 1 1 1 6 6 6 6 6 6 6 6 6 4 1 4 9 9 9 9 9 9
1 1 7 1 1 1 1 1 1 1 1 1 1 1 1 2 6 2 6 6 2 2 6 2 9 9 9 9 9 9 9 9 9 9
1 1 1 7 7 1 1 1 1 1 1 1 1 1 1 6 6 6 6 2 5 2 2 4 9 9 9 9 9 9 9 9 9 9
1 1 1 7 1 1 1 1 1 1 1 1 1 1 2 6 6 6 2 5 2 2 0 9 9 9 9 9 9 9 9 9 9 9
1 1 1 1 1 1 2 2 2 2 2 2 1 2 6 6 6 2 6 2 6 2 0 9 9 9 9 9 9 9 9 9 9 9
2 1 1 1 1 2 2 1 2 2 2 2 1 2 6 6 6 6 6 2 6 4 9 9 9 9 9 9 9 8 8 8 8
2 2 1 1 1 1 1 1 1 1 1 1 1 1 6 6 2 6 6 6 4 4 9 9 9 9 9 9 9 8 8 8 8
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 0 0 0 0 0 4 9 9 9 9 9 9 9 8 8 8 8
1 1 1 1 1 1 1 1 1 1 7 1 5 2 2 2 0 0 0 0 0 0 4 9 9 9 9 9 9 9 8 8 8 8
1 1 1 1 1 1 1 1 1 1 1 5 2 2 2 2 0 0 0 0 4 4 6 9 9 9 9 9 9 9 9 9 9 9
1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 5 2 0 5 4 6 9 9 9 9 9 9 9 9 9 9 9 9
1 1 1 1 7 1 1 1 1 1 2 2 6 6 5 5 2 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 1 1 1 7 2 2 2 1 1 2 2 6 5 5 0 2 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
1 1 1 7 2 2 2 2 1 1 2 6 6 5 5 0 0 4 4 4 9 9 0 0 0 0 9 9 9 9 9 9 9 9
1 1 0 1 2 1 2 1 1 1 1 1 1 1 0 0 0 0 4 0 0 0 4 0 0 0 0 9 9 9 9 8 8 8 8
1 1 1 1 1 1 1 1 1 1 1 7 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 9 9 9 9 8 8 8 8
1 1 1 1 1 1 1 1 1 7 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 9 9 9 8 8 8 8
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 4 4 9 9 9 8 9 9
1 1 1 1 1 7 7 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 4 4 9 9 5 5 9 9
1 1 1 1 1 7 7 7 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 4 4 9 9 5 4 3 3
1 1 1 1 1 7 7 3 3 3 3 3 3 3 3 3 4 0 4 0 0 4 0 0 0 0 4 4 9 9 5 4 4 4
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 5 0 0 0 0 0 0 0 0 4 9 9 9 4 3 3 3
3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 3 3 3 5 3 0 4 4 4 4 4 4 4 9 9 9 4 3 8 8
3 8 3 4 4 4 4 4 9 3 9 4 8 8 8
3 3 8 3 3 3 3 3 3 3 3 3 3 3 3 3 8 3 8 8 3 3 8 3 8 4 4 8 4 8 3 8 3 3 3
8 8 3 8 8 3 8 3 3 3 8 3 8 3 3 3 3 3 3 3 8 3 8 4 8 8 8 3 3 3 3 3 3
3 3 8 3 3 3 3 3 3 3 8 8 8 8 3 3 8 8 3 3 8 8 8 8 8 8 8 8 3 8 3 8
3 3 8 8 3 3 3 8 3 3 3 3 3 3 3 3 8 8 3 3 8 3 3 3 8 3 8 8 8 8 3 8
3 3 8 8 3 3 3 8 8 3 8 3 3 3 3 3 3 3 3 3 3 3 3 3 8 3 8 8 8 8 8 8
3 3 8 3 3 3 3 8 3 8 8 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 8 8 8 8 8 8



squirrel reaching for a nut



squirrel reaching for a nut



**Up-sampled
4x**

squirrel reaching for a nut

slide from Steve Seitz's [video](#)



squirrel reaching for a nut

Parti, <https://parti.research.google/>



squirrel reaching for a nut underwater

slide from Steve Seitz's [video](#)



fossil of a squirrel reaching for a nut

slide from Steve Seitz's [video](#)



squirrel made of toothpicks wearing sunglasses reaching for a nut

slide from Steve Seitz's [video](#)



DLSR photograph of a whimsical fantasy house shaped like a squirrel
with windows and a door, in the forest

slide from Steve Seitz's [video](#)



Squirrel reaching for a nut. by Leonardo da Vinci

slide from Steve Seitz's [video](#)



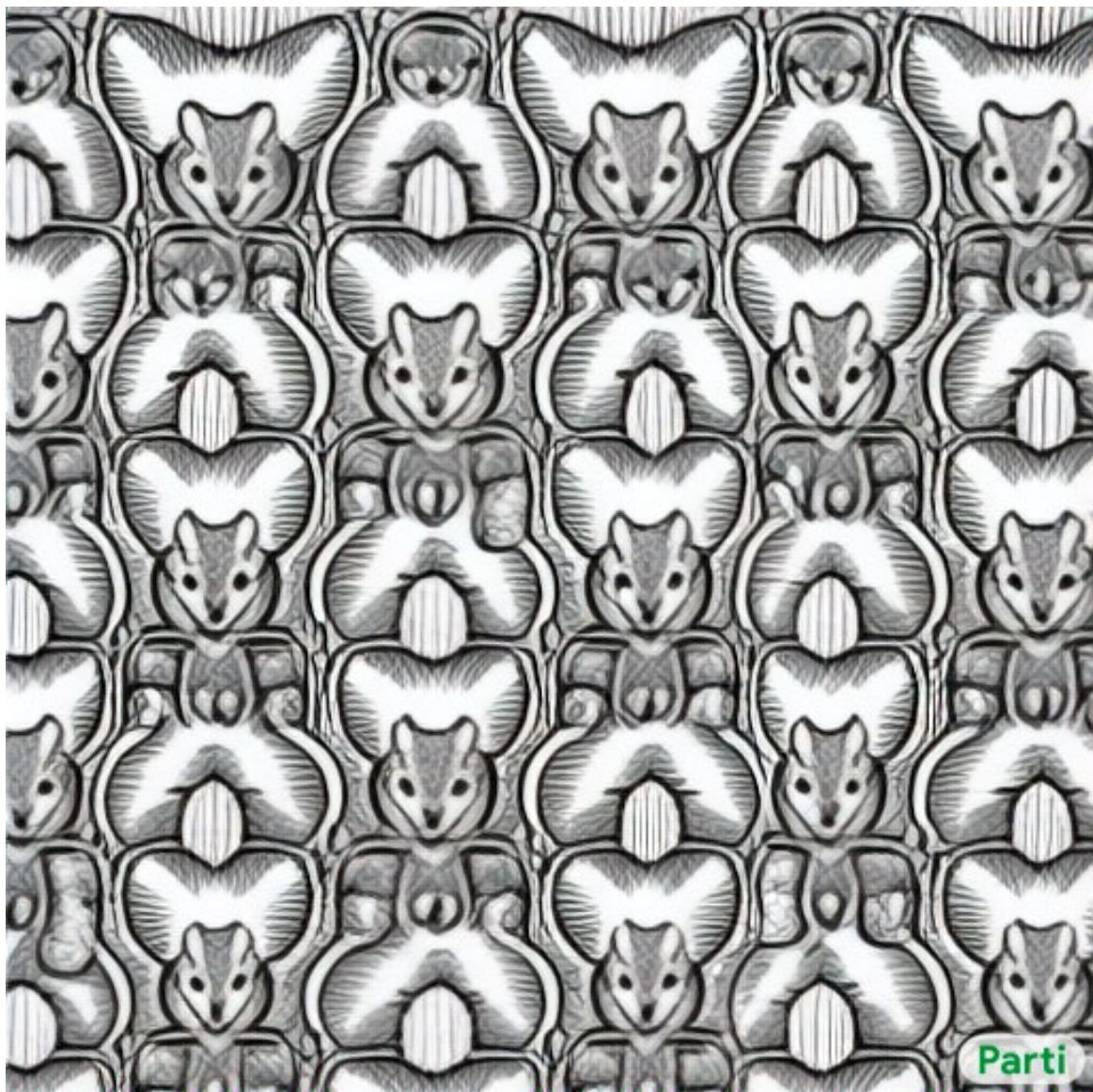
Squirrel reaching for a nut. Van Gogh painting

slide from Steve Seitz's [video](#)



Intricately carved cathedral door of a squirrel reaching for a nut

slide from Steve Seitz's [video](#)



Squirrel reaching for a nut. Woodcut tessellation pattern by M.C. Escher

slide from Steve Seitz's [video](#)



Squirrel reaching for a nut. Latte art

slide from Steve Seitz's [video](#)

Vaswani et al.,
2017

arXiv:1706.03762v7 [cs.CL] 2 Aug 2023

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Vaswani et al.,
2017

arXiv:1706.03762v7 [cs.CL] 2 Aug 2023

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaizer@google.com

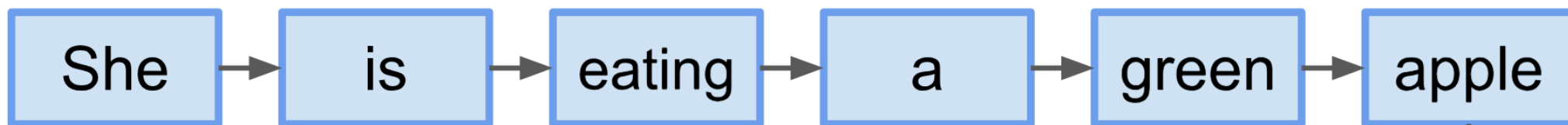
Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Sequence 2 Sequence models in language

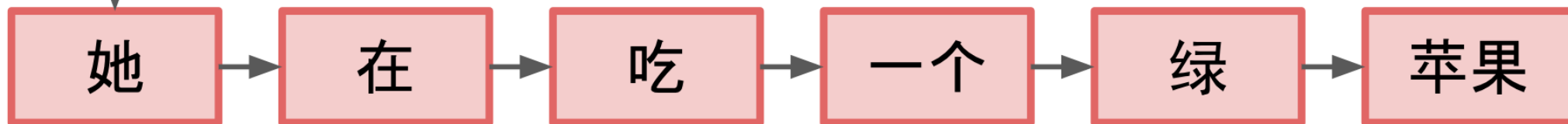
Encoder



Context vector (length: 5)

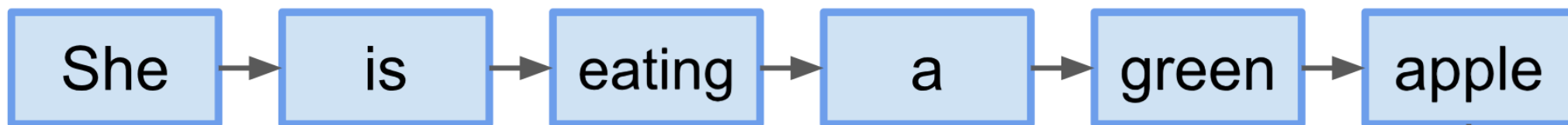
[0.1, -0.2, 0.8, 1.5, -0.3]

Decoder



Attention and Context in language

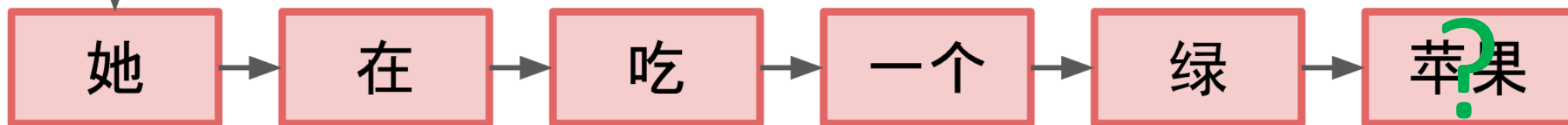
Encoder



Context vector (length: 5)

[0.1, -0.2, 0.8, 1.5, -0.3]

Decoder



attention

Self-Attention

- Content-based querying
- Retrieves similar items
- Weighted sum of similarities
- Constant path length between any two positions
- Variable-sized perceptive field
- Gating/multiplication enables crisp error propagation
- Trivial to parallelize (per layer)
- Can replace sequence-aligned recurrence entirely

Self-Attention Order in Machine Translation

- Encoder-Decoder Attention:
 - from output attending to words in input sequence
- Encoder Self-Attention:
 - attention to words in input sequence (all directions)
- Masked Decoder Self-Attention
 - in output attending only to words that come before

Self-Attention Order in Machine Translation

- Encoder-Decoder Attention:
 - from output attending to words in input sequence
- Encoder Self-Attention:
 - attention to words in input sequence (all directions) You cannot use this if you are predicting the output
- Masked Decoder Self-Attention
 - in output attending only to words that come before Use this instead!

Self-Attention Order in Machine Translation

- Encoder-Decoder Attention:
 - from output attending to words in input sequence
- Encoder Self-Attention:
 - attention to words in input sequence (all directions)
- Masked Decoder Self-Attention
 - in output attending only to words that come before

You cannot use this if you are predicting the output

Use this instead!

BUT with word-by-word processing this would take a very long time to train!

Transformer Architecture

Vaswani et al., 2017

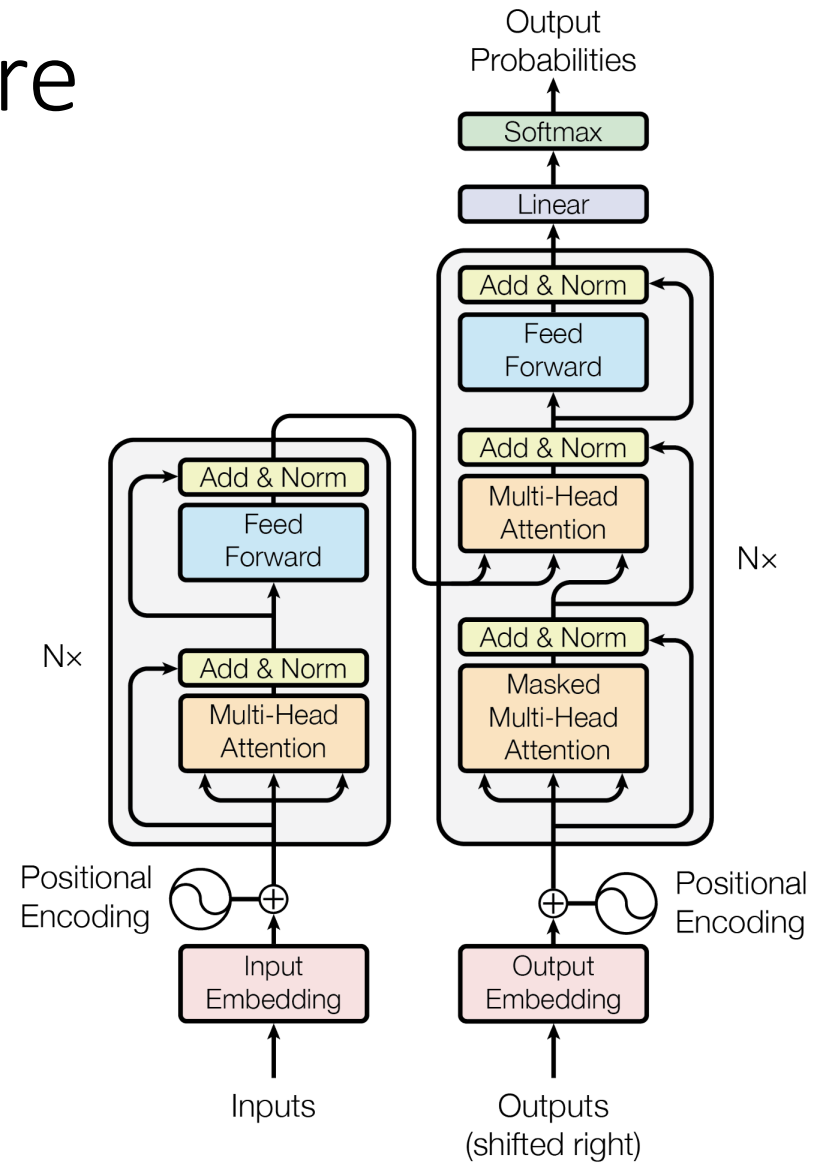


Figure 1: The Transformer - model architecture.

Transformer Architecture

Feed forward network processes every English word

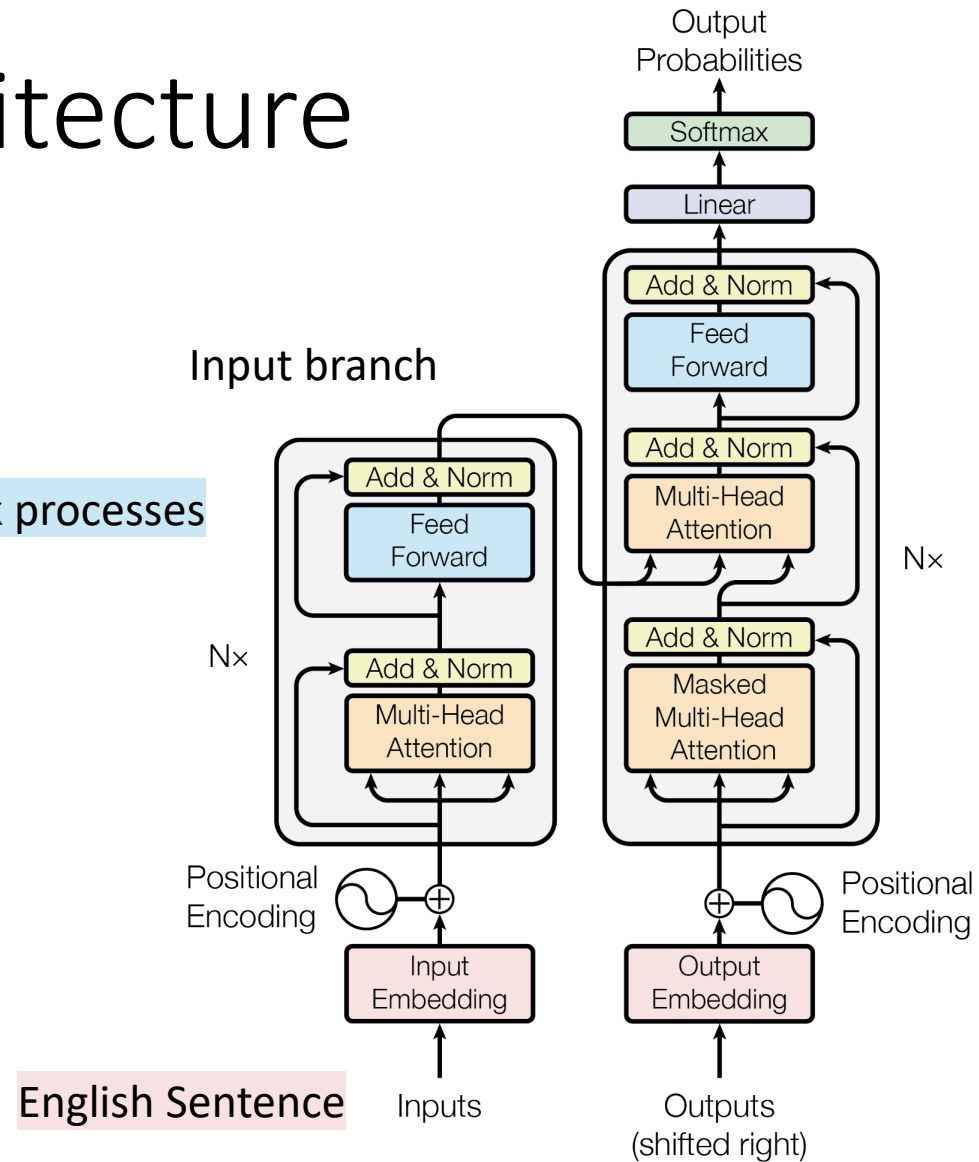


Figure 1: The Transformer - model architecture.

Transformer Architecture

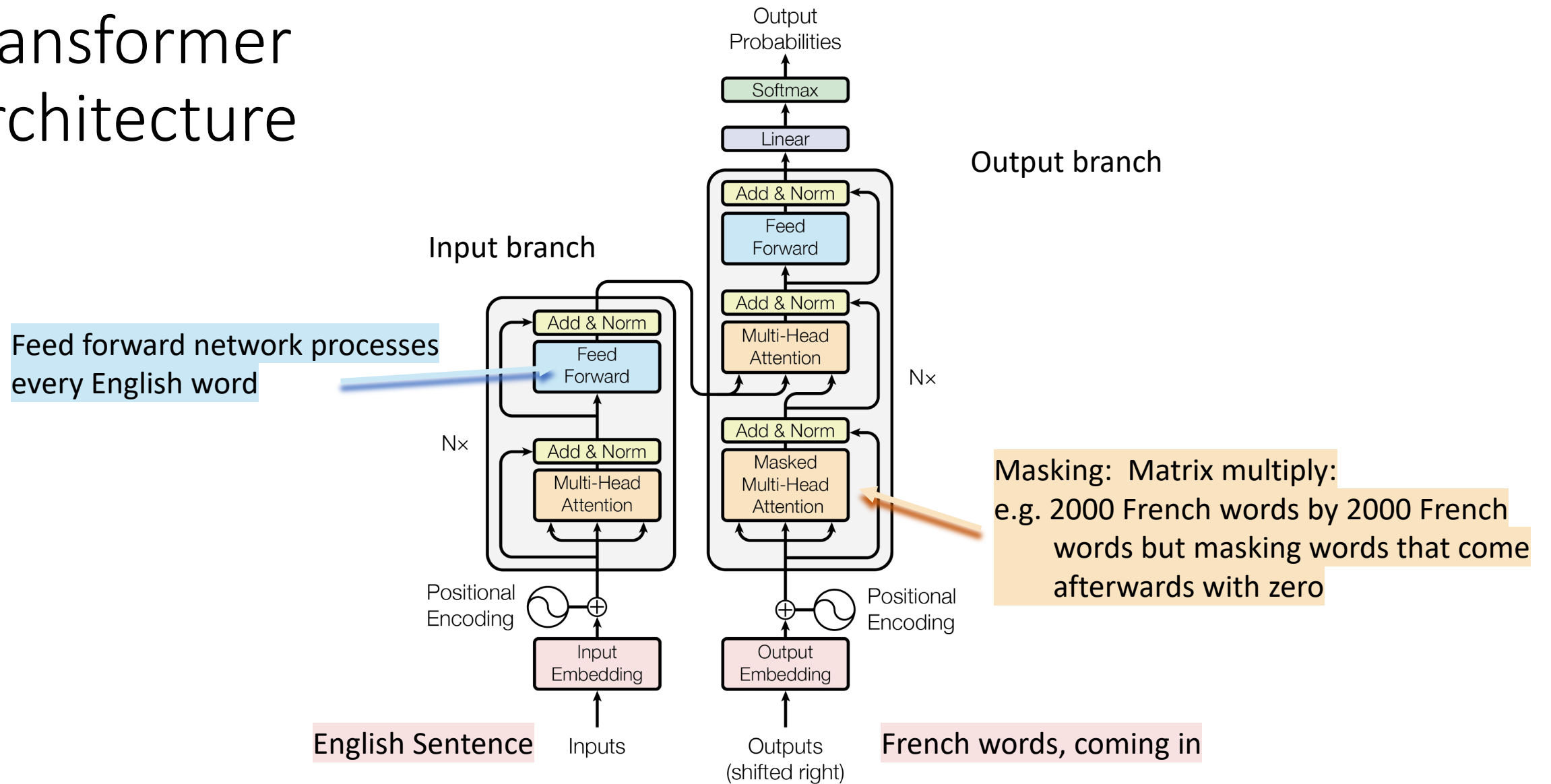


Figure 1: The Transformer - model architecture.

Transformer Architecture

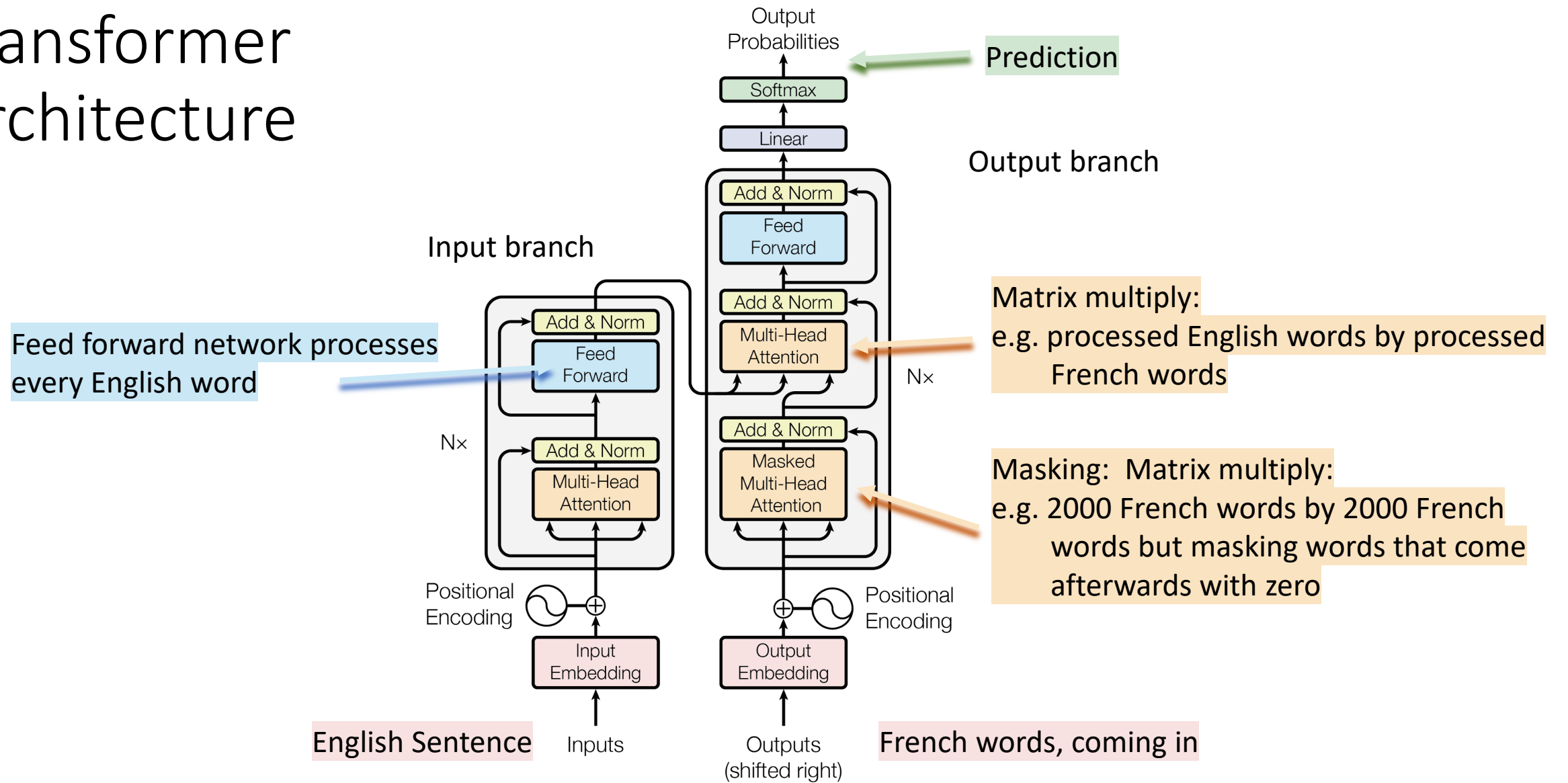


Figure 1: The Transformer - model architecture.

Masking Attention

$$\text{Attention}(Q,K,V) = \text{softmax}(Q K^T) V$$

Q = query vector = current English (or French) word

K key and V value = memory of words seen before

Goal: Find key(s) most similar to query and retrieve value(s) that correspond to this/these key(s)

Softmax = $\frac{\sum_i e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}}$ v_i produces probability distribution over keys with peaks for keys similar to query

Masking Attention

$$\text{Attention}(Q,K,V) = \text{softmax}(Q K^T) V$$

Acts as a weight mask over V

Q = query vector = current English (or French) word

K key and V value = memory of words seen before

Goal: Find key(s) most similar to query and retrieve value(s) that correspond to this/these key(s)

Softmax = $\frac{\sum_i e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}}$ v_i produces probability distribution over keys with peaks for keys similar to query

Masking Attention

Very fast:

2 matrix multiplications & 1 softmax operation

$$\text{Attention}(Q,K,V) = \text{softmax}(Q K^T) V$$

Acts as a weight mask over V

Q = query vector = current English (or French) word

K key and V value = memory of words seen before

Goal: Find key(s) most similar to query and retrieve value(s) that correspond to this/these key(s)

Softmax = $\frac{\sum_i e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}}$ v_i produces probability distribution over keys with peaks for keys similar to query

Masking Attention

Very fast:

2 matrix multiplications & 1 softmax operation

$$\text{Attention}(Q,K,V) = \text{softmax}(Q K^T / \text{sqrt}(d_k)) V$$

Acts as a weight mask over V

Technical detail:

$\text{sqrt}(d_k)$ normalization needed for training

Q = query vector = current English (or French) word

K key and V value = memory of words seen before

Goal: Find key(s) most similar to query and retrieve value(s) that correspond to this/these key(s)

Softmax = $\frac{\sum_i e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}}$ v_i produces probability distribution over keys with peaks for keys similar to query

Why Multi-Head Attention?

Feed forward network processes every English word

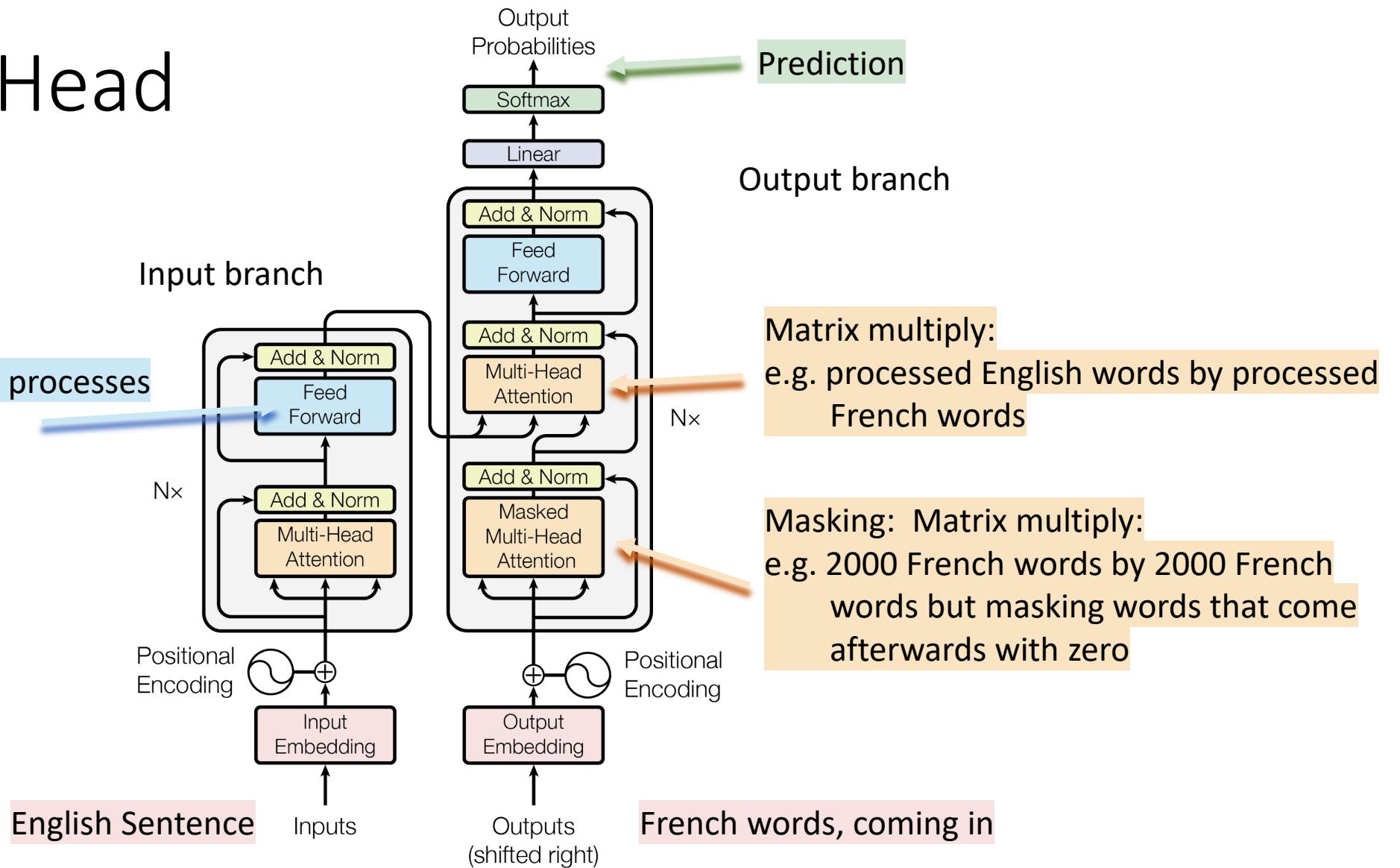


Figure 1: The Transformer - model architecture.

Why Multi-Head Attention?

- Multiple attention layers (heads) in parallel
- Each head uses different linear transformation
- Different heads can learn different relationships

Attention Visualizations

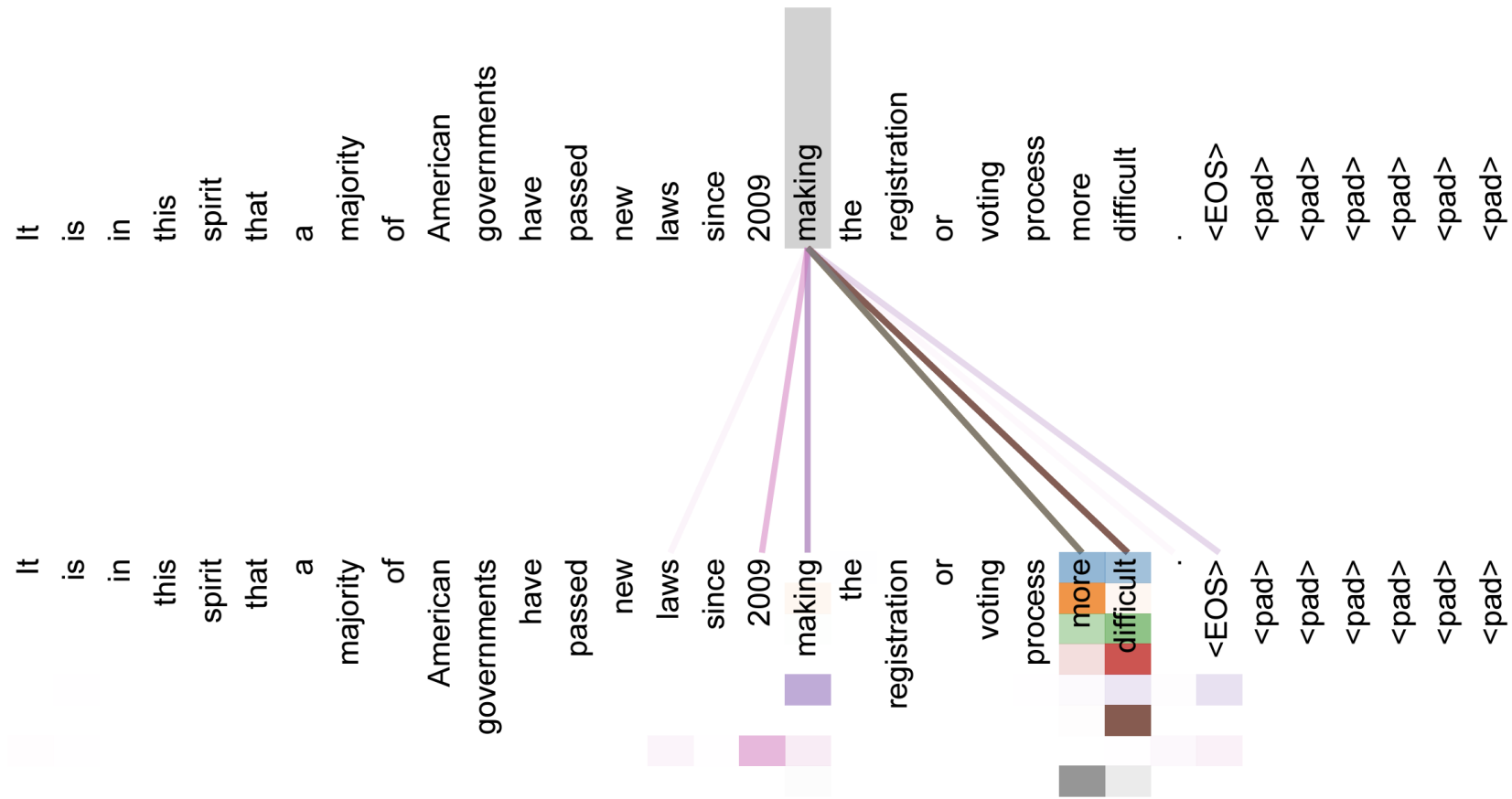


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder **self-attention** in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colors represent different heads. Best viewed in color.

Training a Transformer

- ADAM optimizer
- Dropout during training at every layer
- Label smoothing
- Auto-regressive decoding with beam-search
- Checkpoint-averaging
- Library available: <https://github.com/tensorflow/tensor2tensor>

Transformer Architecture Complexity

- n = number of words in sequence
- d = network depth

Number of operations: $n^2 d$

Number of activations: $n^2 + n d$

Much better than CNNs or RNNs with number of operations $n d^2$

Transformer Architecture Complexity

- n = number of words in sequence (<70 words per sentence)
- d = network depth (maybe 1000)

Every word attends to every word

Number of operations: $n^2 d$ e.g., $70 \times 70 \times 1000 = 4.9$ mill

Number of activations: $n^2 + n d$

Much better than CNNs or RNNs with number of operations $n d^2$

e.g., $70 \times 1000 \times 1000 = 70$ mill

Vaswani et al.,
2017

arXiv:1706.03762v7 [cs.CL] 2 Aug 2023

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Dosovitskiy
et al.,
2020

Vision
Transformer
ViT

2 [cs.CV] 3 Jun 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

**Alexey Dosovitskiy^{*,†}, Lucas Beyer^{*}, Alexander Kolesnikov^{*}, Dirk Weissenborn^{*},
Xiaohua Zhai^{*}, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby^{*,†}**

^{*}equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.¹

ViT Architecture

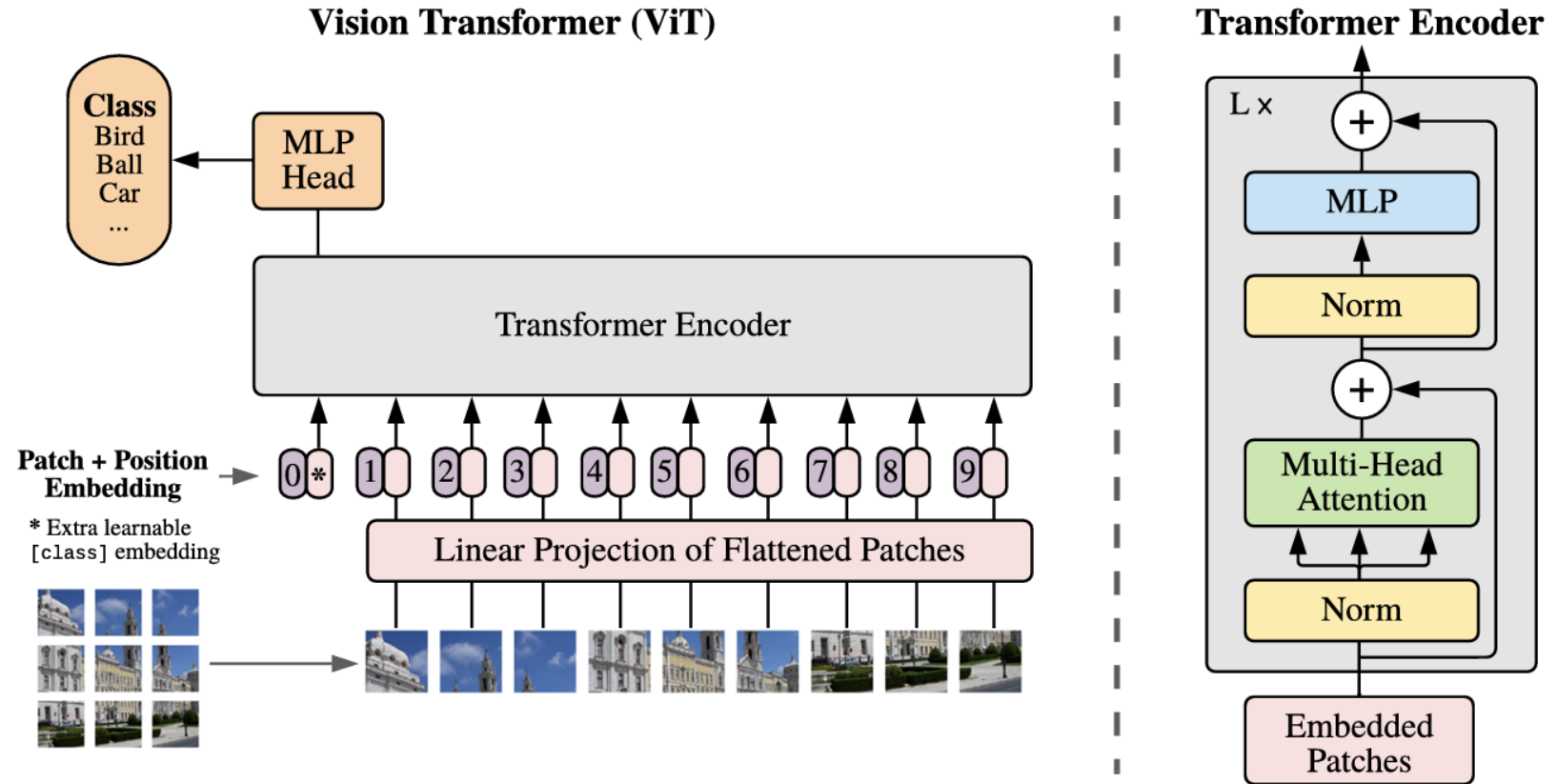
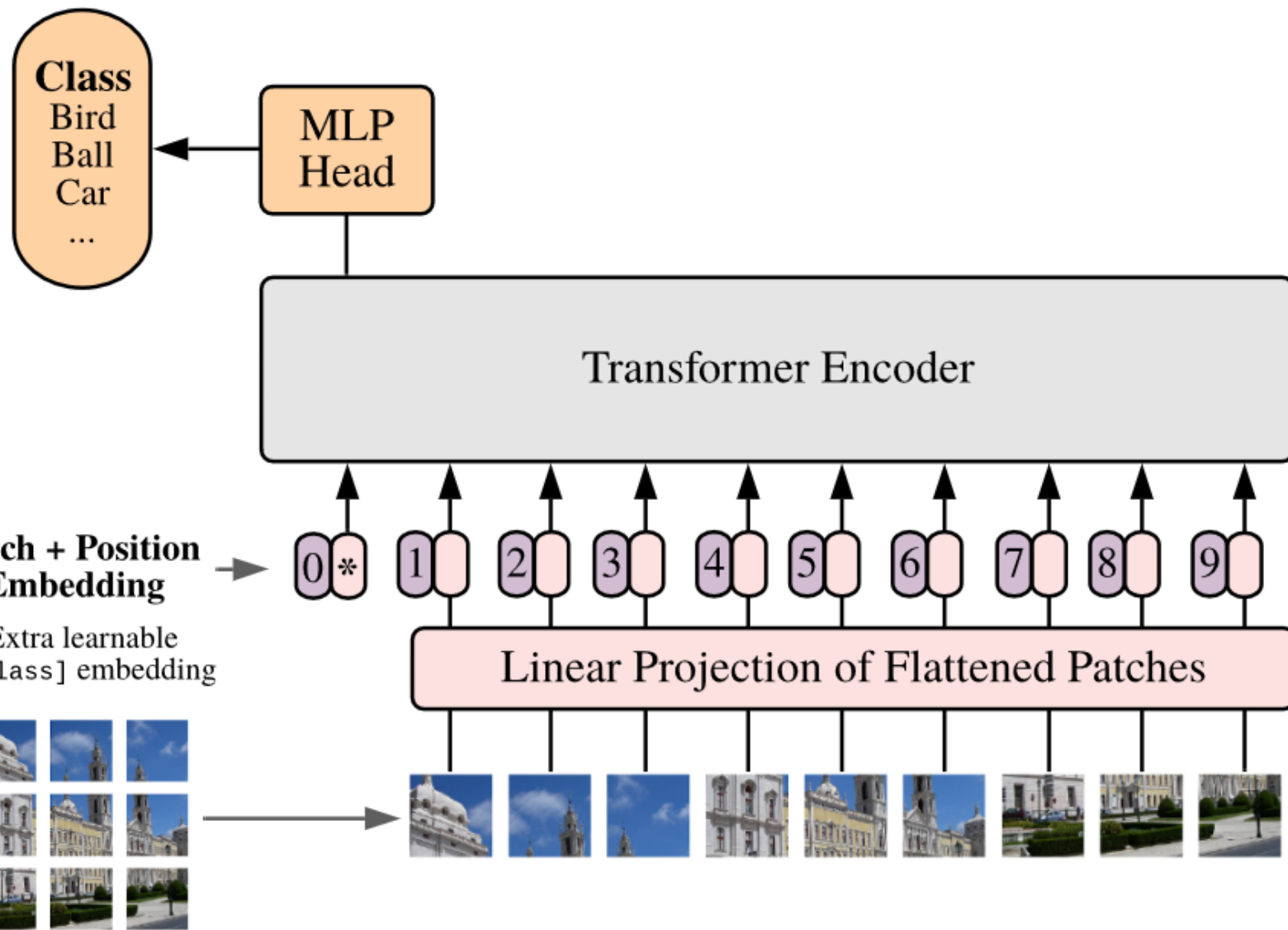
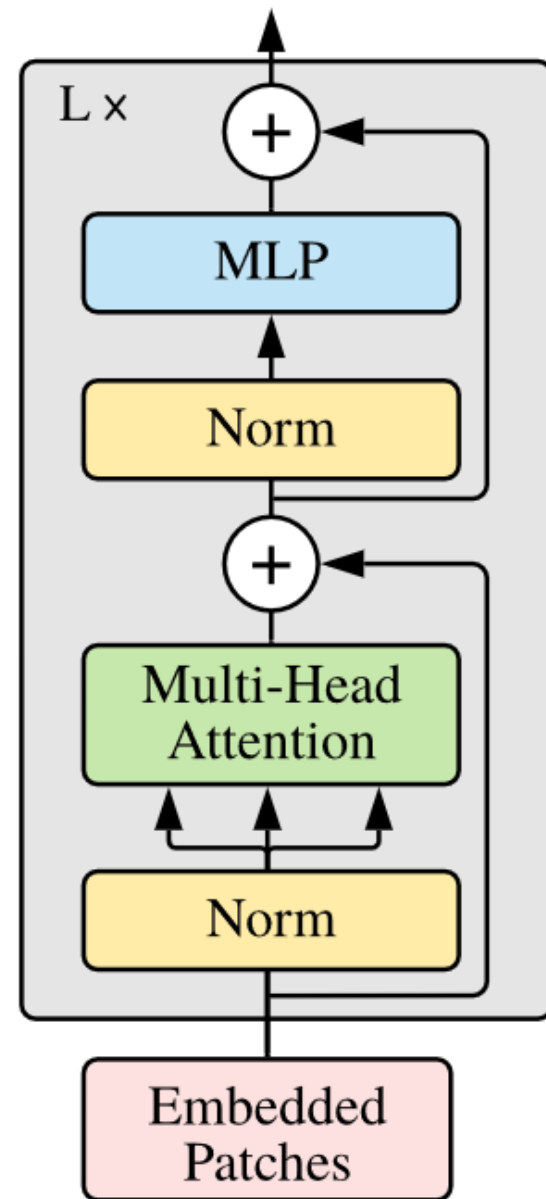


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable “classification token” to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

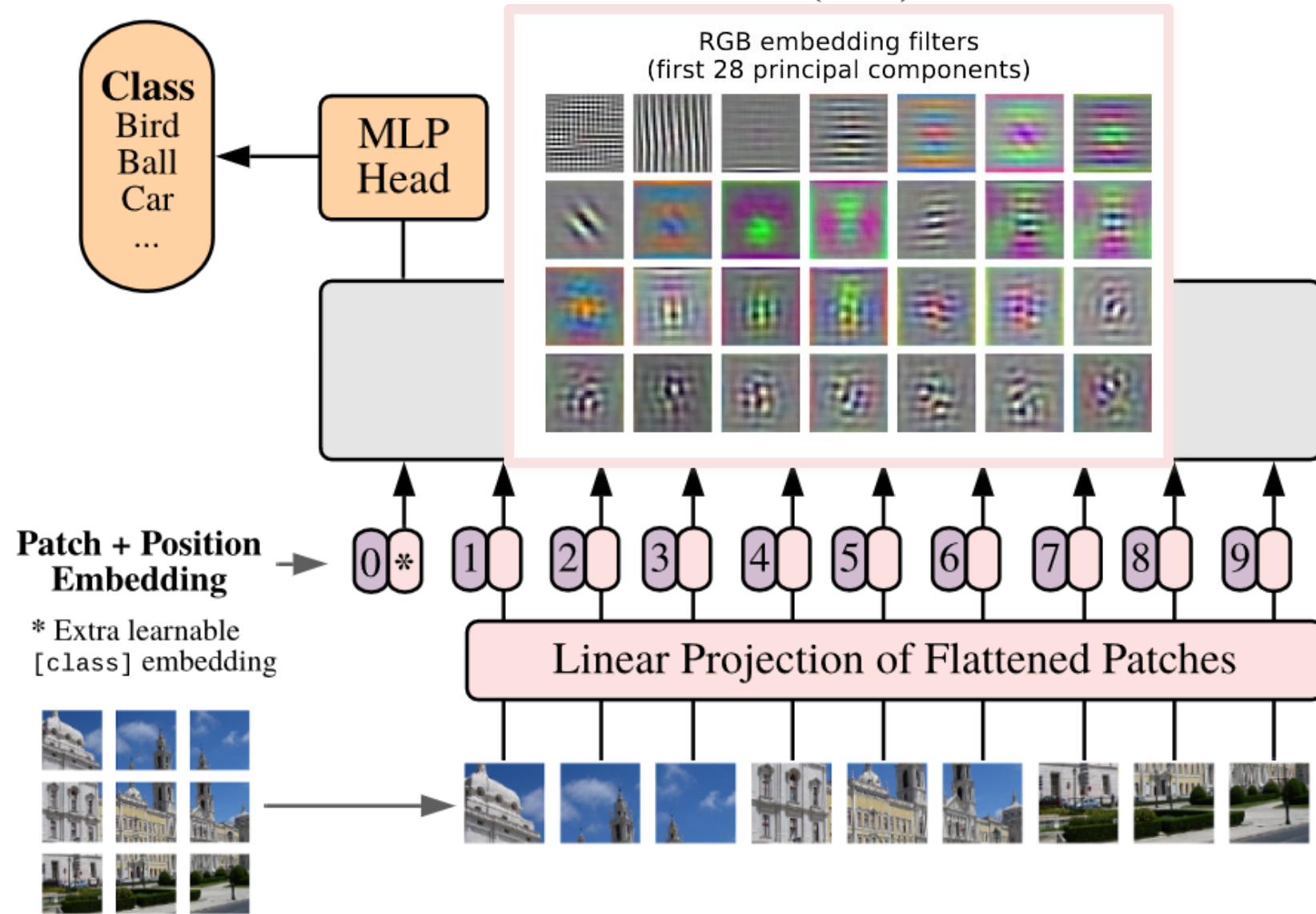
Vision Transformer (ViT)



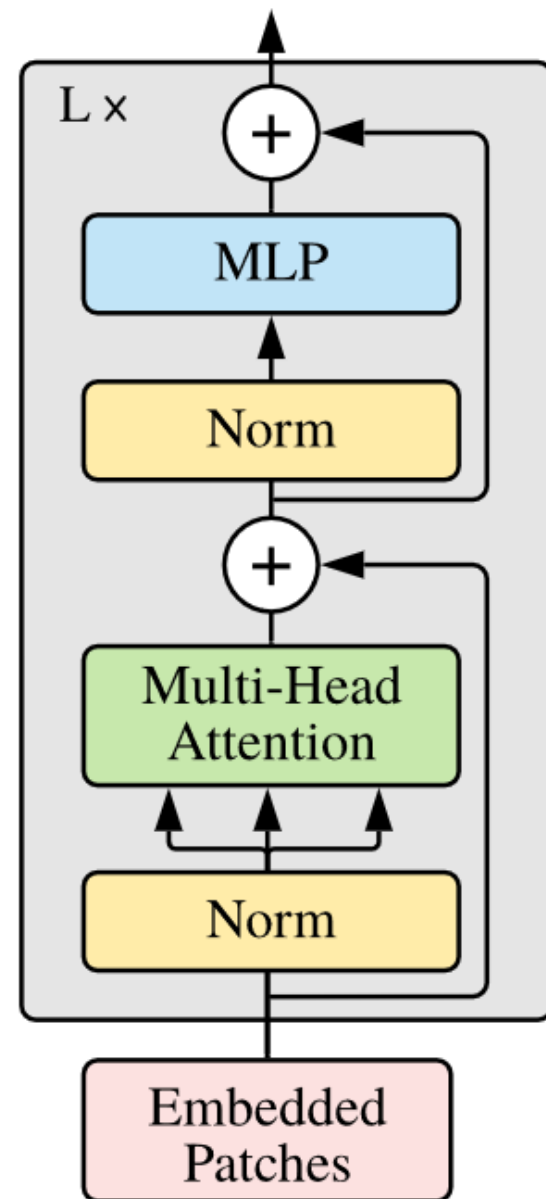
Transformer Encoder

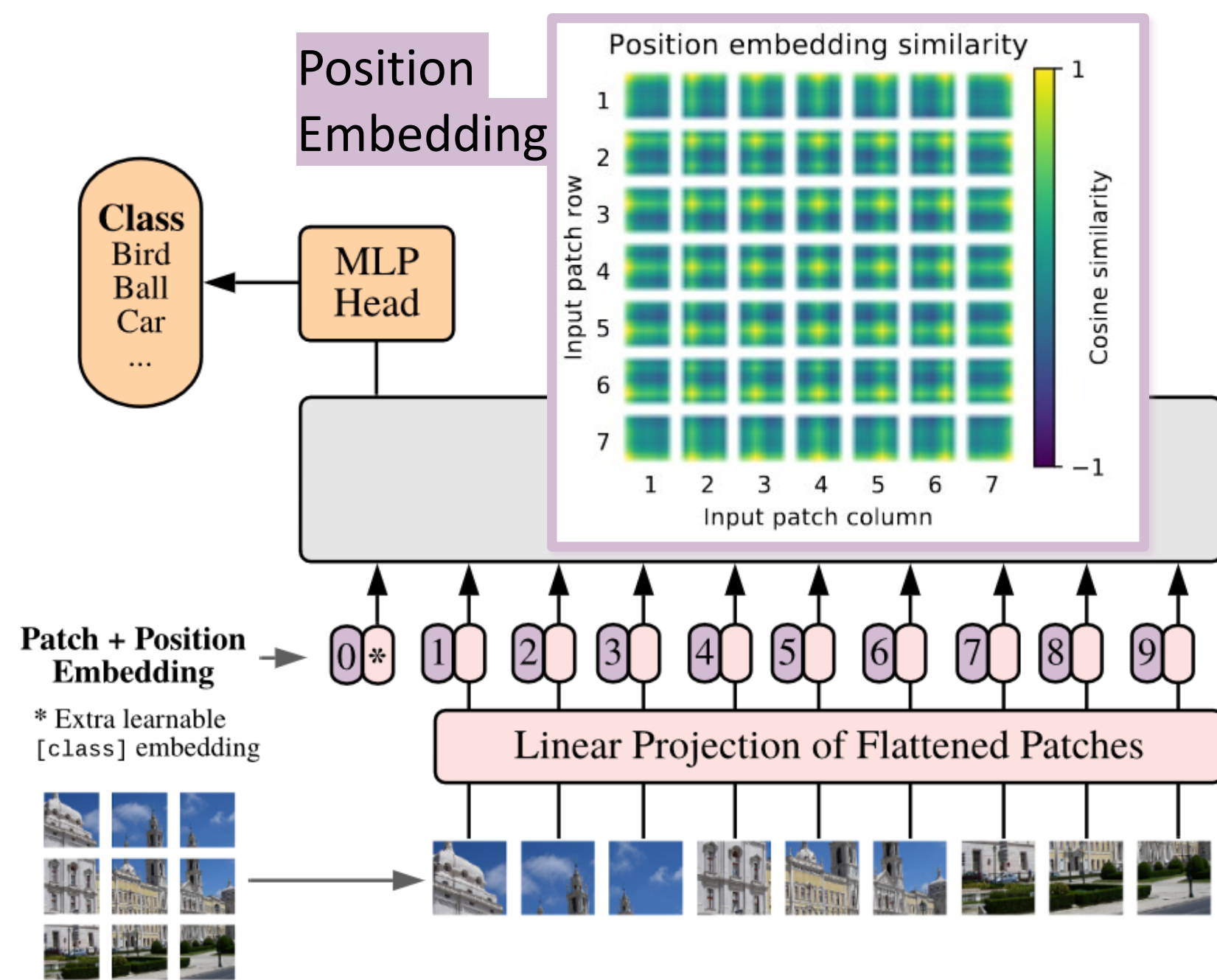


Embedding Vision Transformer (ViT)

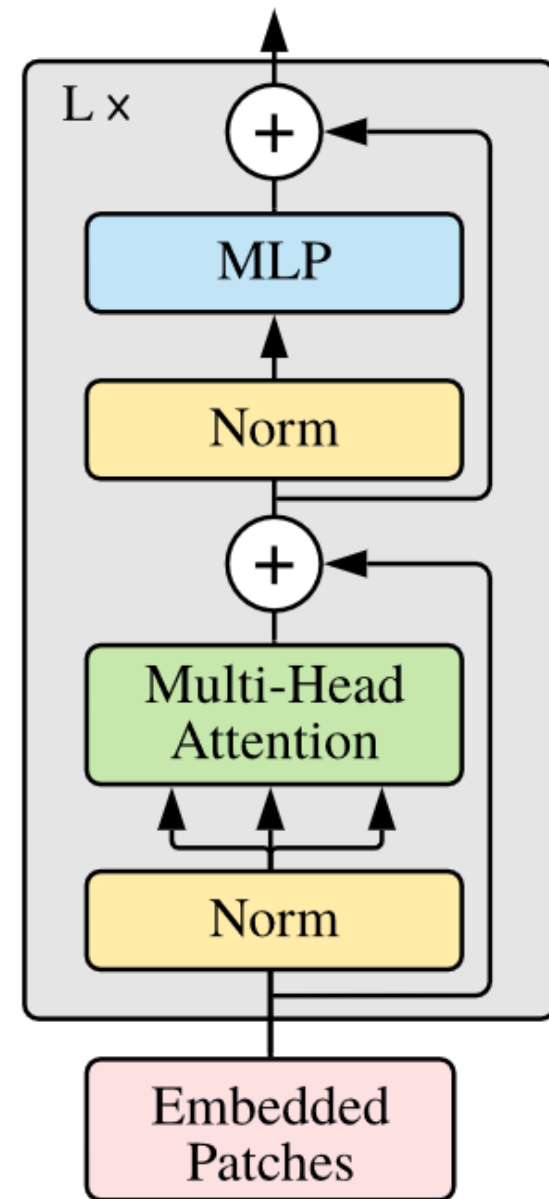


Transformer Encoder





Transformer Encoder



Vision Transformer

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

Vision Transformer Results

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.55 ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	90.72 ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	99.50 ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	—
CIFAR-100	94.55 ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	—
Oxford-IIIT Pets	97.56 ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	—
Oxford Flowers-102	99.68 ± 0.02	99.74 ± 0.00	99.61 ± 0.02	99.63 ± 0.03	—
VTAB (19 tasks)	77.63 ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

Vision Transformer Results

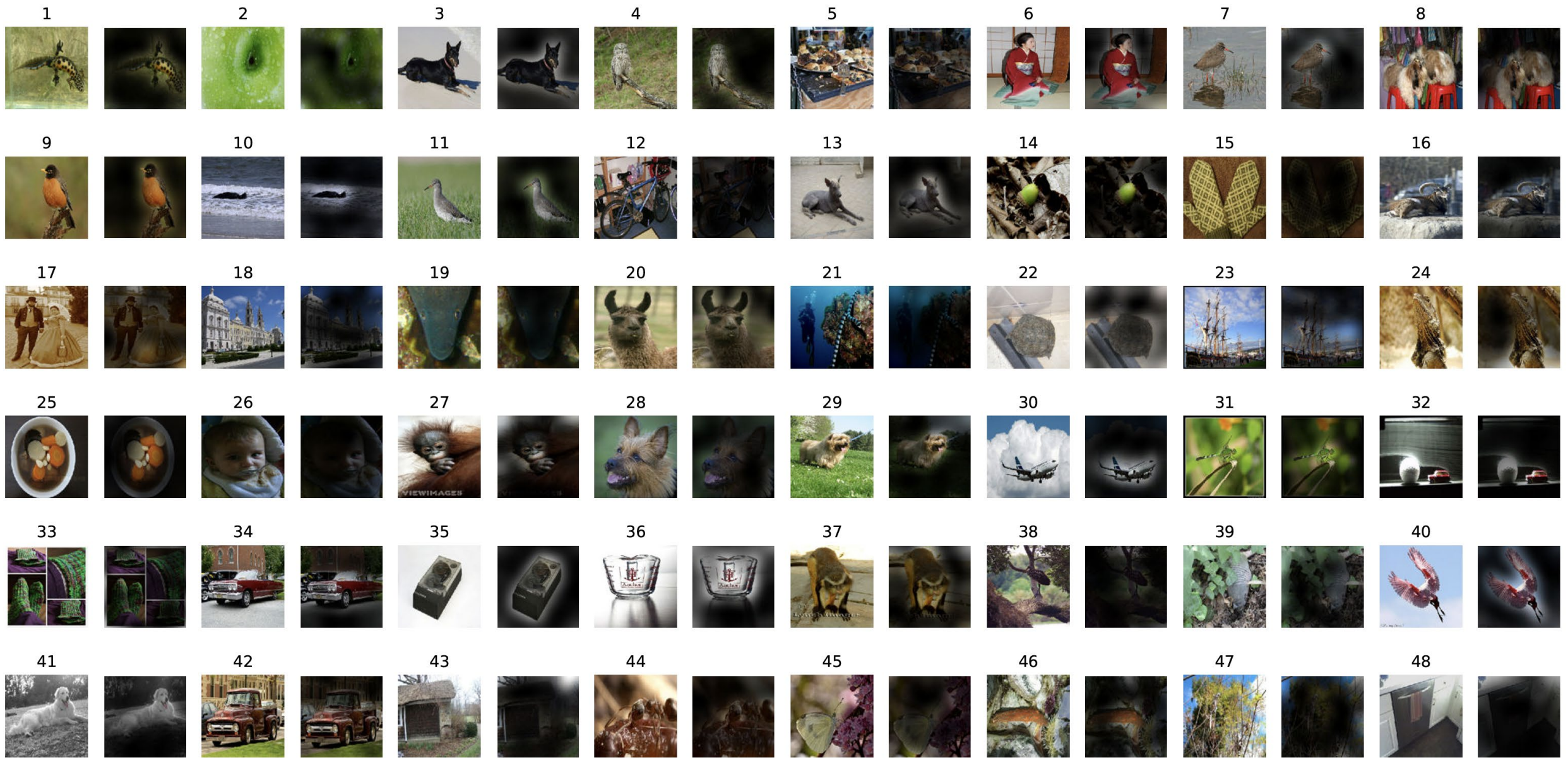
	● Caltech101	● CIFAR-100	● DTD	● Flowers102	● Pets	● Sun397	● SVHN
ViT-H/14 (JFT)	95.3	85.5	75.2	99.7	97.2	65.0	88.9
ViT-L/16 (JFT)	95.4	81.9	74.3	99.7	96.7	63.5	87.4
ViT-L/16 (I21k)	90.8	84.1	74.1	99.3	92.7	61.0	80.9

Vision Transformer

Input

Attention





Dosovitskiy
et al.,
2020

Vision
Transformer
ViT

2 [cs.CV] 3 Jun 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

**Alexey Dosovitskiy^{*,†}, Lucas Beyer^{*}, Alexander Kolesnikov^{*}, Dirk Weissenborn^{*},
Xiaohua Zhai^{*}, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby^{*,†}**

^{*}equal technical contribution, [†]equal advising

Google Research, Brain Team

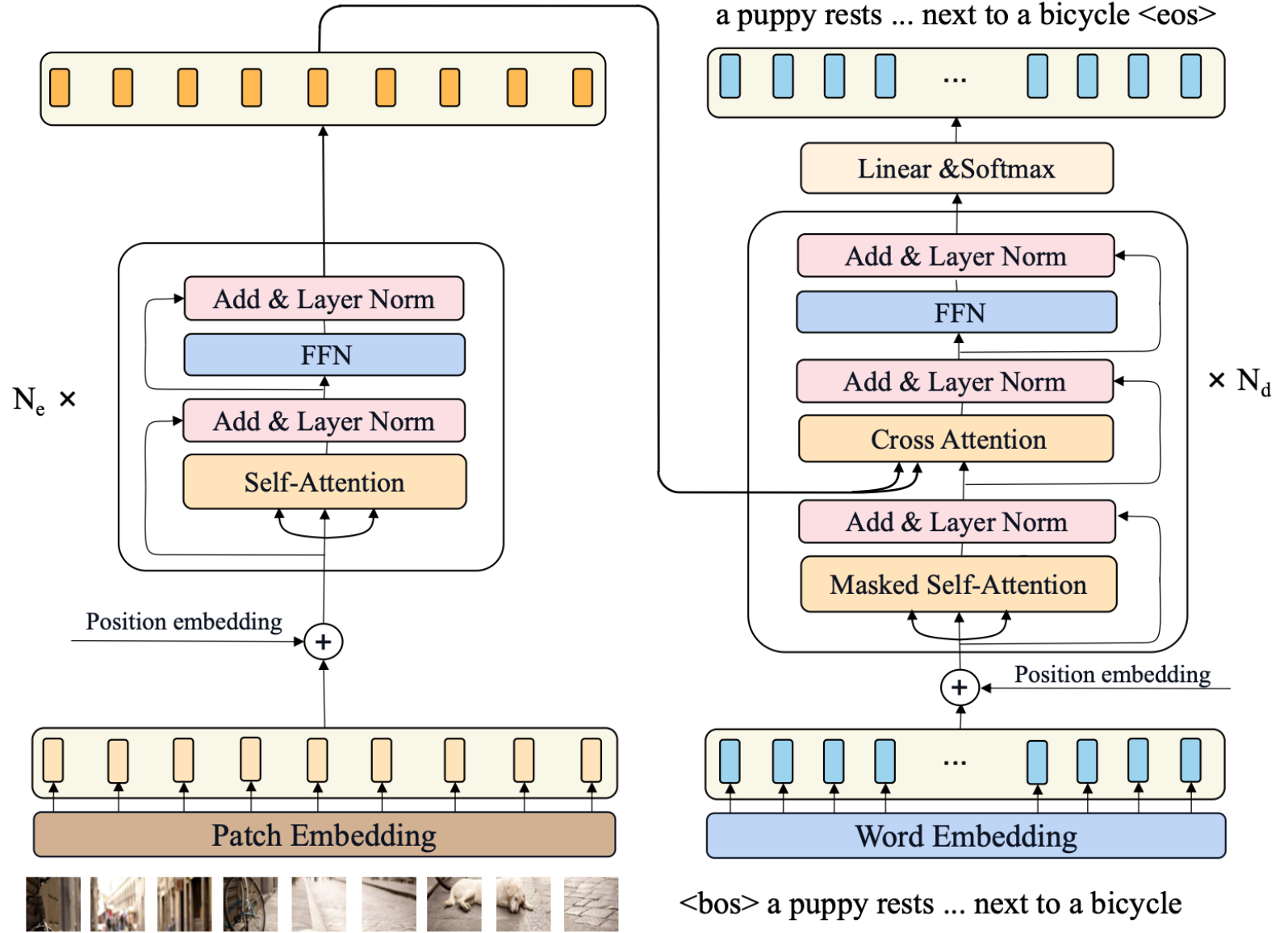
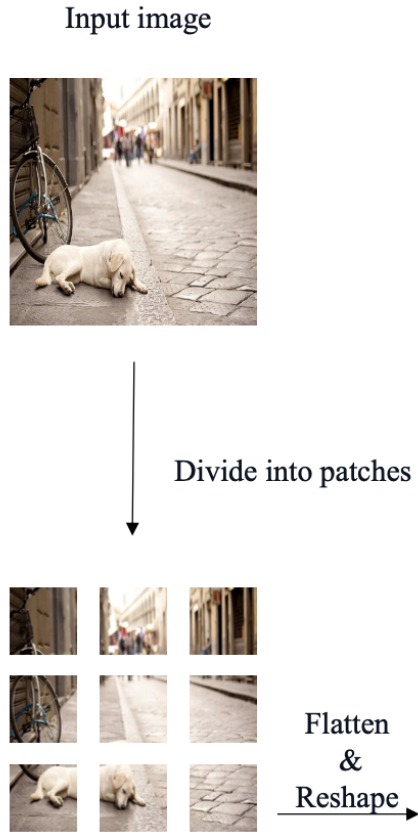
{adosovitskiy, neilhoulby}@google.com

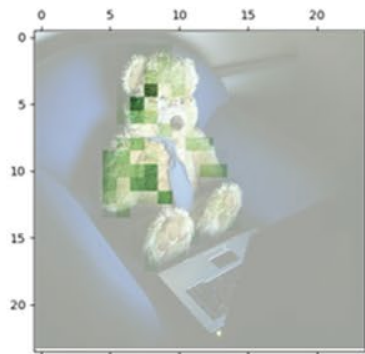
ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.¹

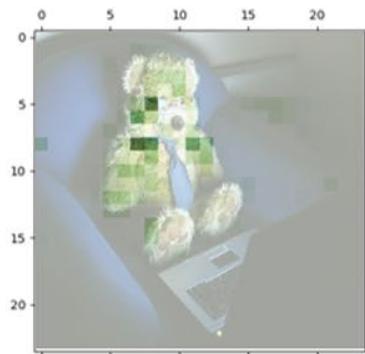
Task: Image Captioning
 = Creating text that describes the image

Liu et al., 2021

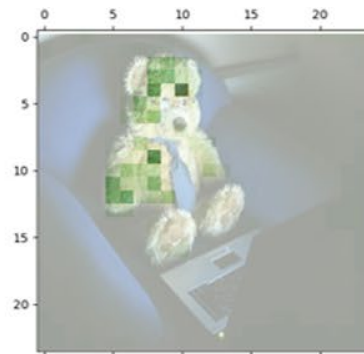




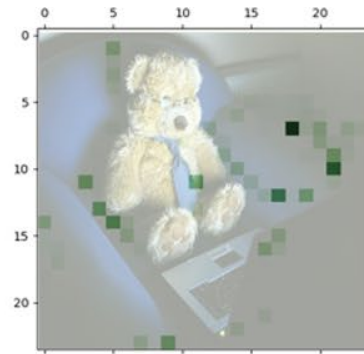
[a]



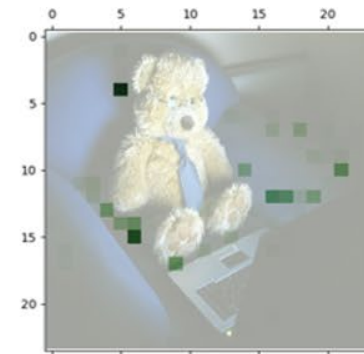
[teddy]



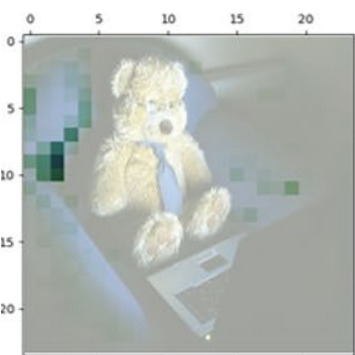
[bear]



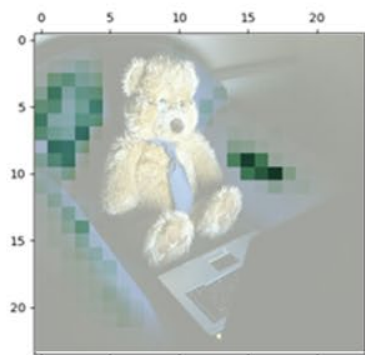
[sitting]



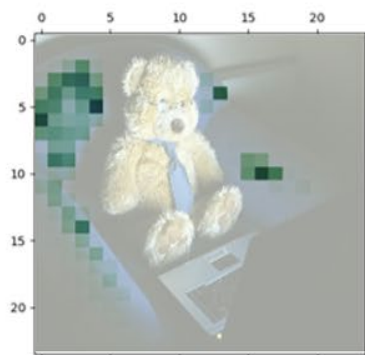
[in]



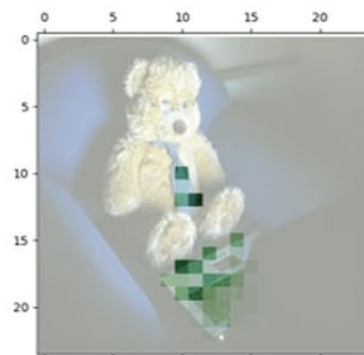
[a]



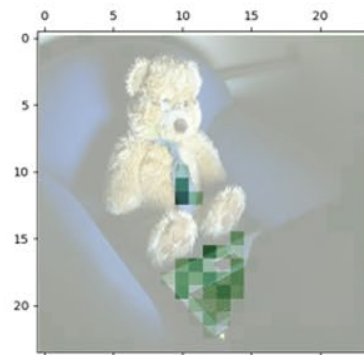
[blue]



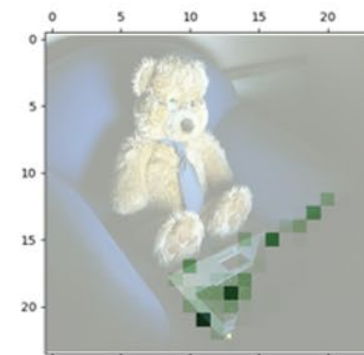
[chair]



[with]



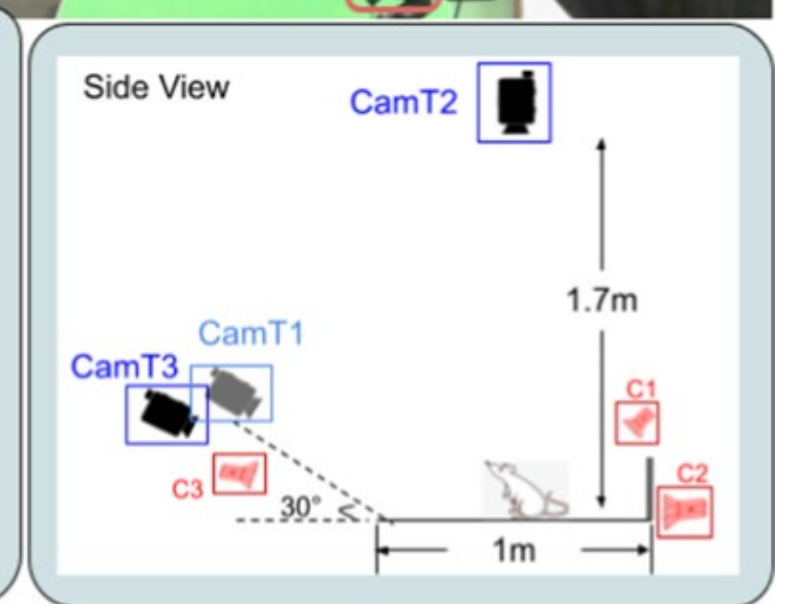
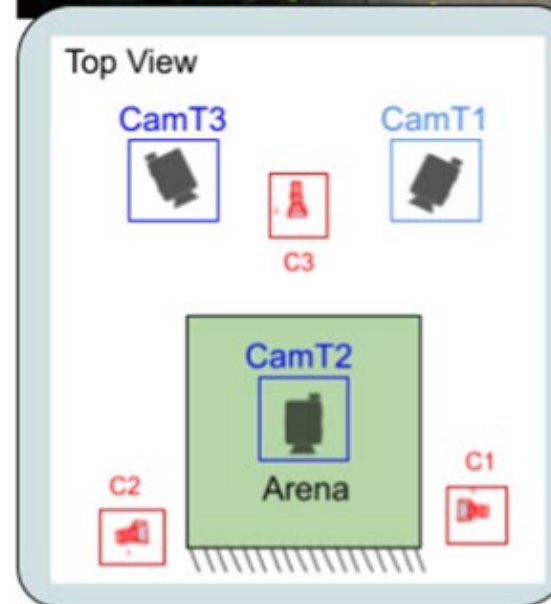
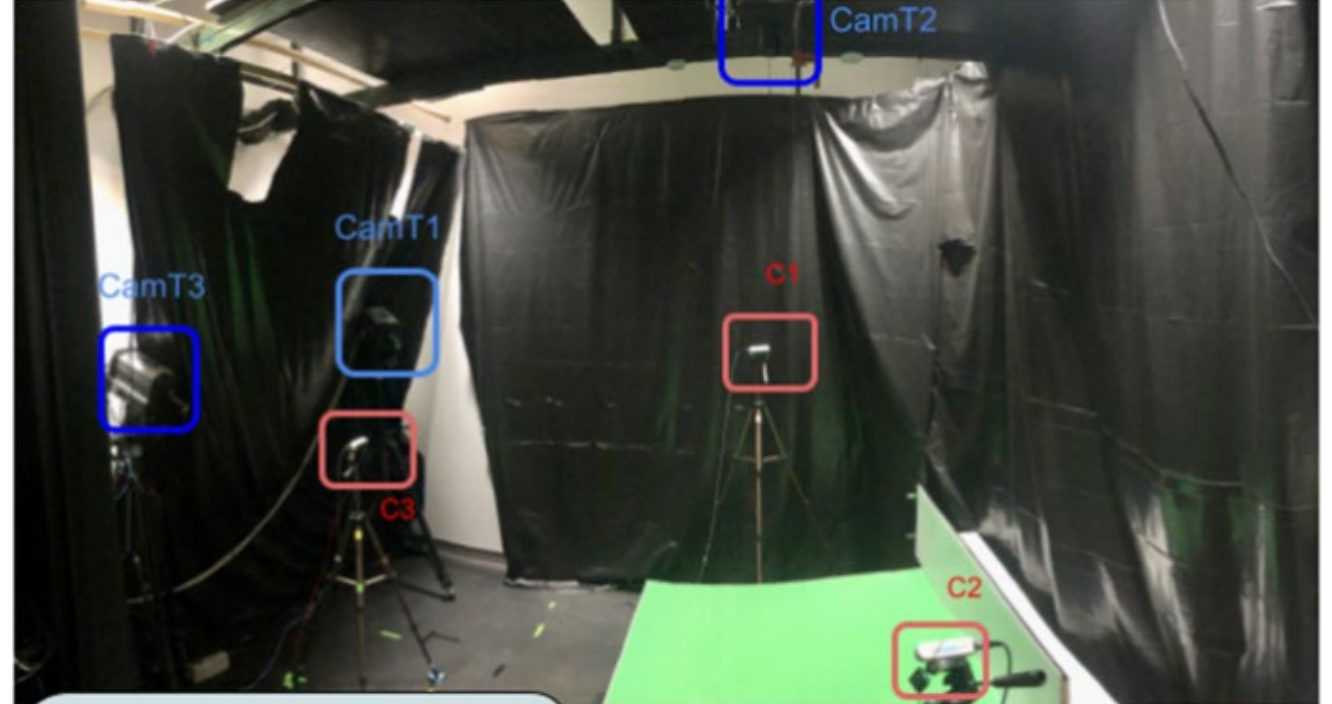
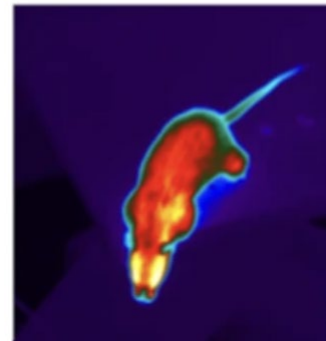
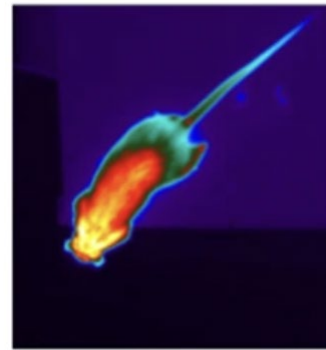
[a]



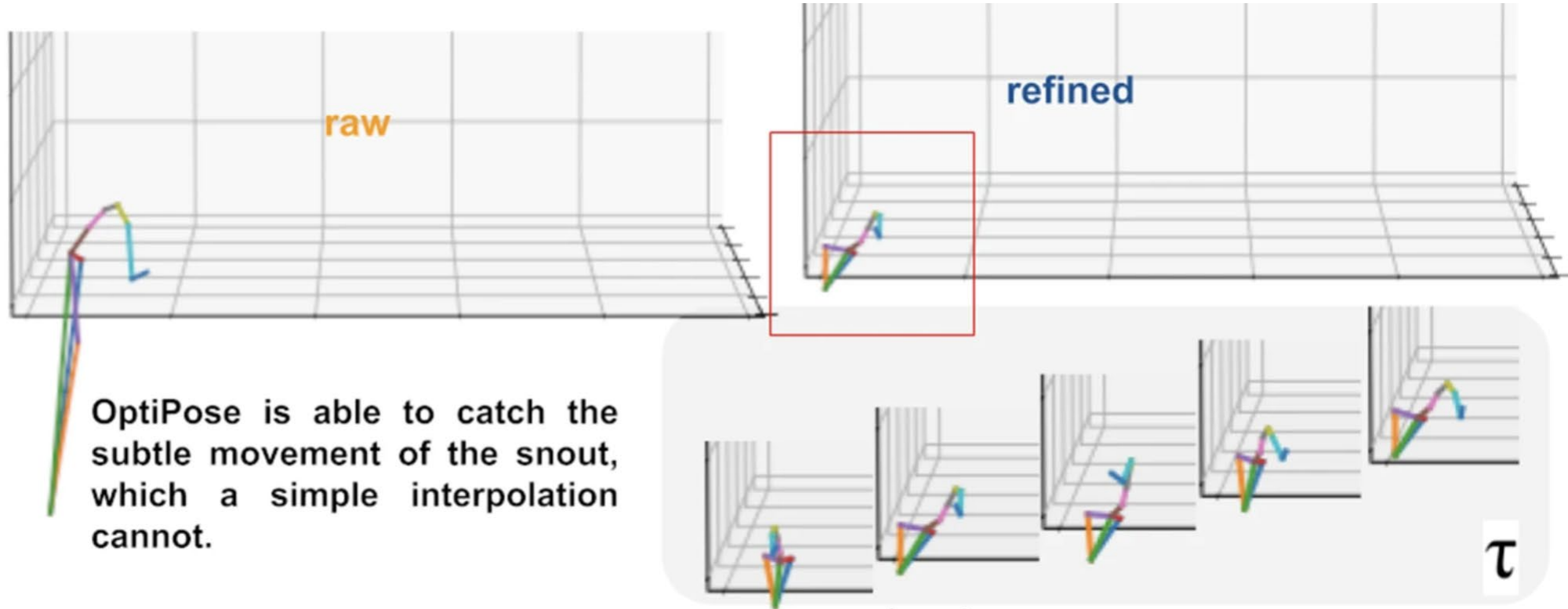
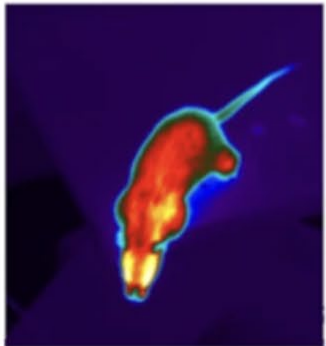
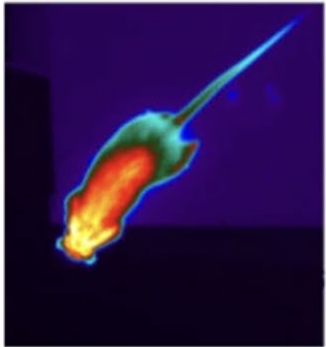
[laptop]

Animal Pose Tracking: 3D Multimodal Dataset and Token-based Pose Optimization

[Patel et al., 2022](#)



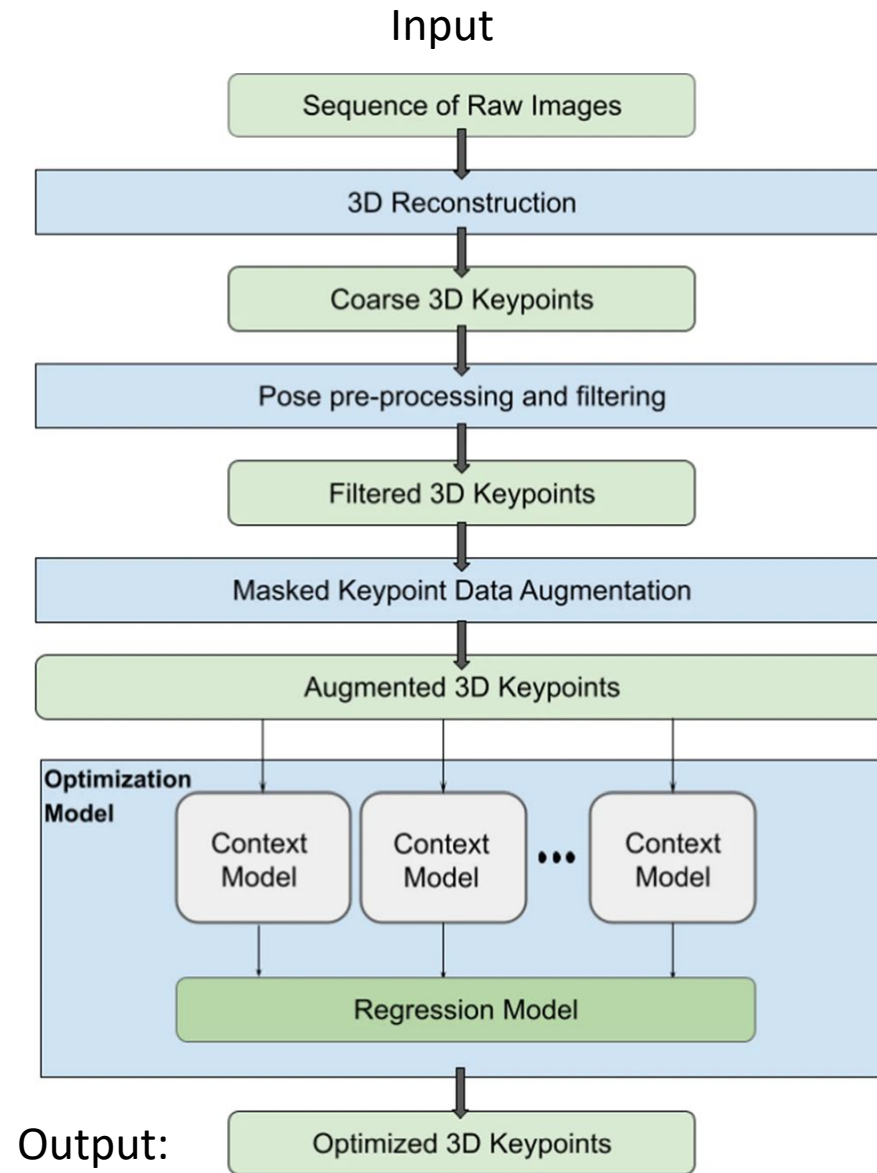
Animal Pose Tracking: 3D Multimodal Dataset and Token- based Pose Optimization



Animal Pose Tracking: 3D Multimodal Dataset and Token-based Pose Optimization

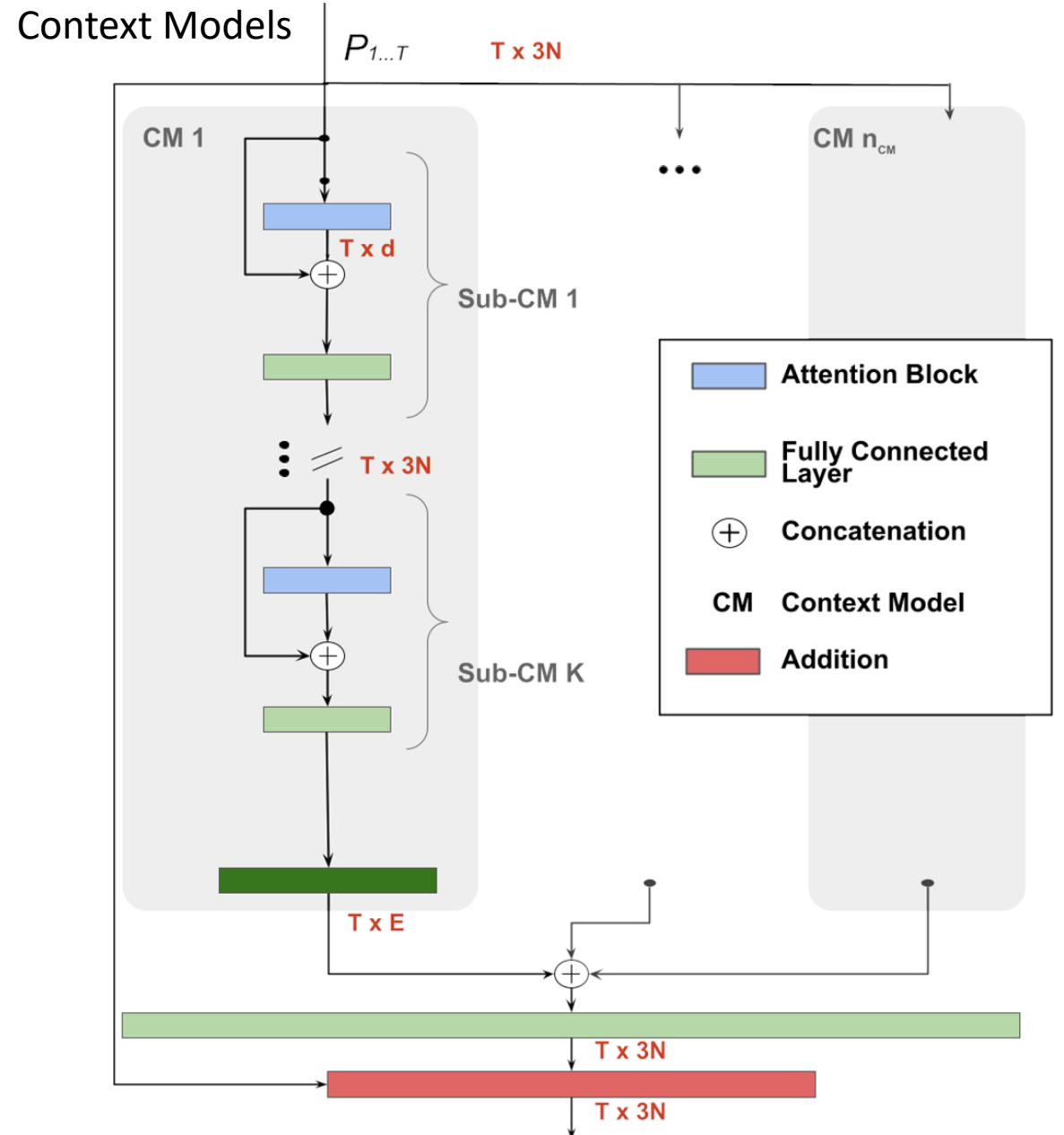
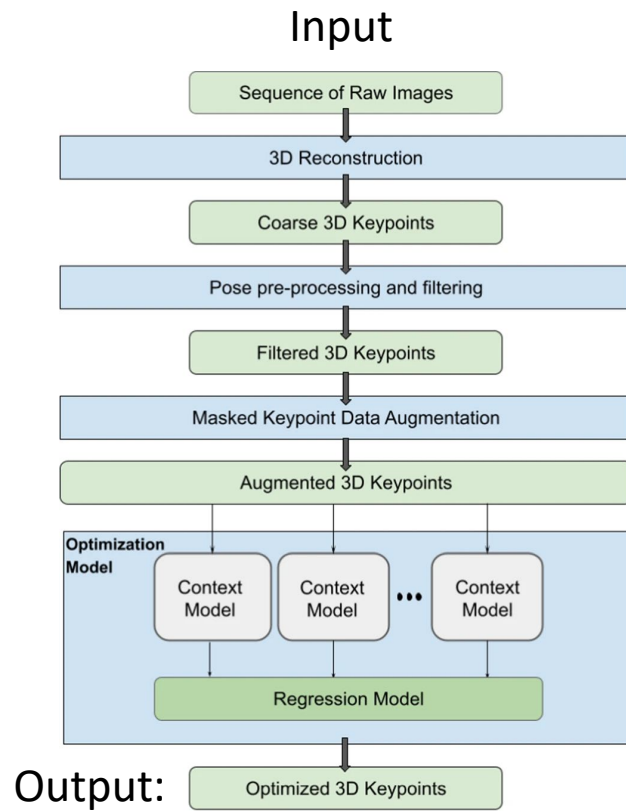
[Patel et al., 2022](#)

Using a Transformer:



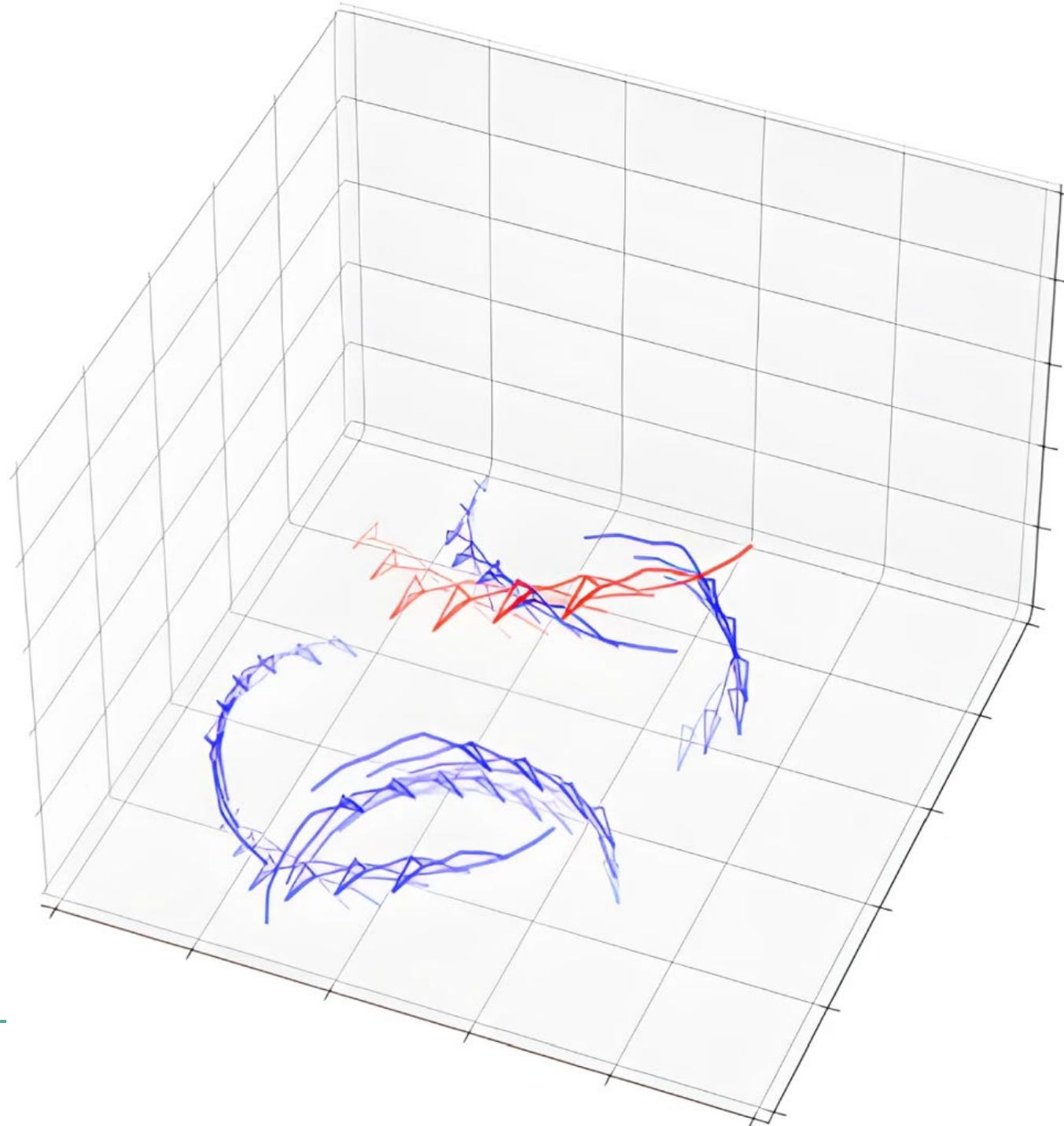
Animal Pose Tracking: 3D Multimodal Dataset and Token-based Pose Optimization

[Patel et al., 2022](#)



Animal Pose
Tracking: 3D
Multimodal
Dataset and
Token-based
Pose
Optimization

[Patel et al.,
2022](#)



A ConvNet for the 2020s

Zhuang Liu^{1,2*} Hanzi Mao¹ Chao-Yuan Wu¹ Christoph Feichtenhofer¹ Trevor Darrell² Saining Xie^{1†}

¹Facebook AI Research (FAIR) ²UC Berkeley

Code: <https://github.com/facebookresearch/ConvNeXt>

Abstract

The “Roaring 20s” of visual recognition began with the introduction of Vision Transformers (ViTs), which quickly superseded ConvNets as the state-of-the-art image classification model. A vanilla ViT, on the other hand, faces difficulties when applied to general computer vision tasks such as object detection and semantic segmentation. It is the hierarchical Transformers (e.g., Swin Transformers) that reintroduced several ConvNet priors, making Transformers practically viable as a generic vision backbone and demonstrating remarkable performance on a wide variety of vision tasks. However, the effectiveness of such hybrid approaches is still largely credited to the intrinsic superiority of Transformers, rather than the inherent inductive biases of convolutions. In this work, we reexamine the design spaces and test the limits of what a pure ConvNet can achieve. We gradually “modernize” a standard ResNet toward the design of a vision Transformer, and discover several key components that contribute to the performance difference along the way. The outcome of this exploration is a family of pure ConvNet models dubbed ConvNeXt. Constructed entirely from standard ConvNet modules, ConvNeXts compete favorably with Transformers in terms of accuracy and scalability, achieving 87.8% ImageNet top-1 accuracy and outperforming Swin Transformers on COCO detection and ADE20K segmentation, while maintaining the simplicity and efficiency of standard ConvNets.

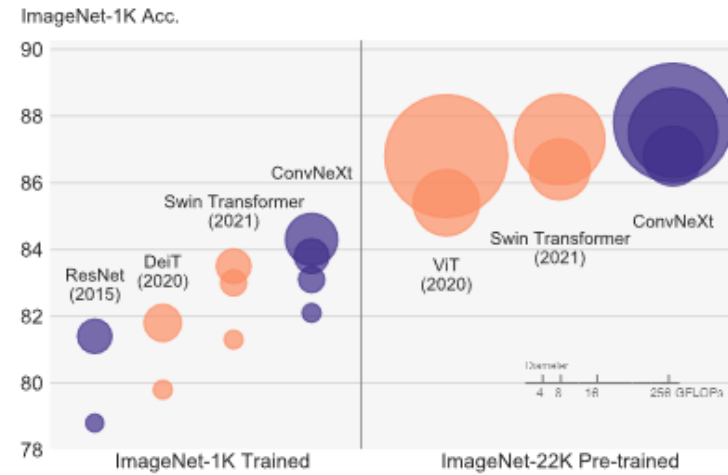


Figure 1. **ImageNet-1K classification** results for • ConvNets and ○ vision Transformers. Each bubble’s area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take $224^2/384^2$ images respectively. ResNet and ViT results were obtained with improved training procedures over the original papers. We demonstrate that a standard ConvNet model can achieve the same level of scalability as hierarchical vision Transformers while being much simpler in design.

visual feature learning. The introduction of AlexNet [40] precipitated the “ImageNet moment” [59], ushering in a new era of computer vision. The field has since evolved at a rapid speed. Representative ConvNets like VGGNet [64], Inceptions [68], ResNe(X)t [28, 87], DenseNet [36], MobileNet [34], EfficientNet [71] and RegNet [54] focused on different aspects of accuracy, efficiency and scalability, and popularized many useful design principles.

arXiv:2201.03545v2 [cs.CV] 2 Mar 2022

A ConvNet for the 2020s

Zhuang Liu^{1,2*} Hanzi Mao¹ Chao-Yuan Wu¹ Christoph Feichtenhofer¹ Trevor Darrell² Saining Xie^{1†}

¹Facebook AI Research (FAIR) ²UC Berkeley

Code: <https://github.com/facebookresearch/ConvNeXt>

Abstract

The “Roaring 20s” of visual recognition began with the introduction of Vision Transformers (ViTs), which quickly superseded ConvNets as the state-of-the-art image classification model. A vanilla ViT, on the other hand, faces difficulties when applied to general computer vision tasks such as object detection and semantic segmentation. It is the hierarchical Transformers (e.g., Swin Transformers) that reintroduced several ConvNet priors, making Transformers practically viable as a generic vision backbone and demonstrating remarkable performance on a wide variety of vision tasks. However, the effectiveness of such hybrid approaches is still largely credited to the intrinsic superiority of Transformers, rather than the inherent inductive biases of convolutions. In this work, we reexamine the design spaces and test the limits of what a pure ConvNet can achieve. We gradually “modernize” a standard ResNet toward the design of a vision Transformer, and discover several key components that contribute to the performance difference along the way. The outcome of this exploration is a family of pure ConvNet models dubbed ConvNeXt. Constructed entirely from standard ConvNet modules, ConvNeXts compete favorably with Transformers in terms of accuracy and scalability, achieving 87.8% ImageNet top-1 accuracy and outperforming Swin Transformers on COCO detection and ADE20K segmentation, while maintaining the simplicity and efficiency of standard ConvNets.

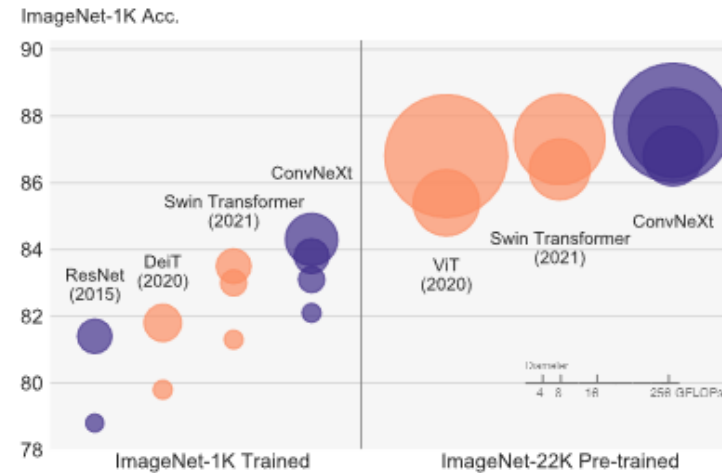


Figure 1. **ImageNet-1K classification** results for • ConvNets and ○ vision Transformers. Each bubble’s area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take $224^2/384^2$ images respectively. ResNet and ViT results were obtained with improved training procedures over the original papers. We demonstrate that a standard ConvNet model can achieve the same level of scalability as hierarchical vision Transformers while being much simpler in design.

visual feature learning. The introduction of AlexNet [40] precipitated the “ImageNet moment” [59], ushering in a new era of computer vision. The field has since evolved at a rapid speed. Representative ConvNets like VGGNet [64], Inceptions [68], ResNe(X)t [28, 87], DenseNet [36], MobileNet [34], EfficientNet [71] and RegNet [54] focused on different aspects of accuracy, efficiency and scalability, and popularized many useful design principles.

arXiv:2201.03545v2 [cs.CV] 2 Mar 2022

Learning Outcomes

Understand

- Concept of attention
- Transformers for NLP (“Attention is All you Need”)
- Vision transformers for object recognition (“An image is worth 16x16 Words”)
- Vision transformers for image captioning
- Vision transformers for 3D pose optimization and tracking