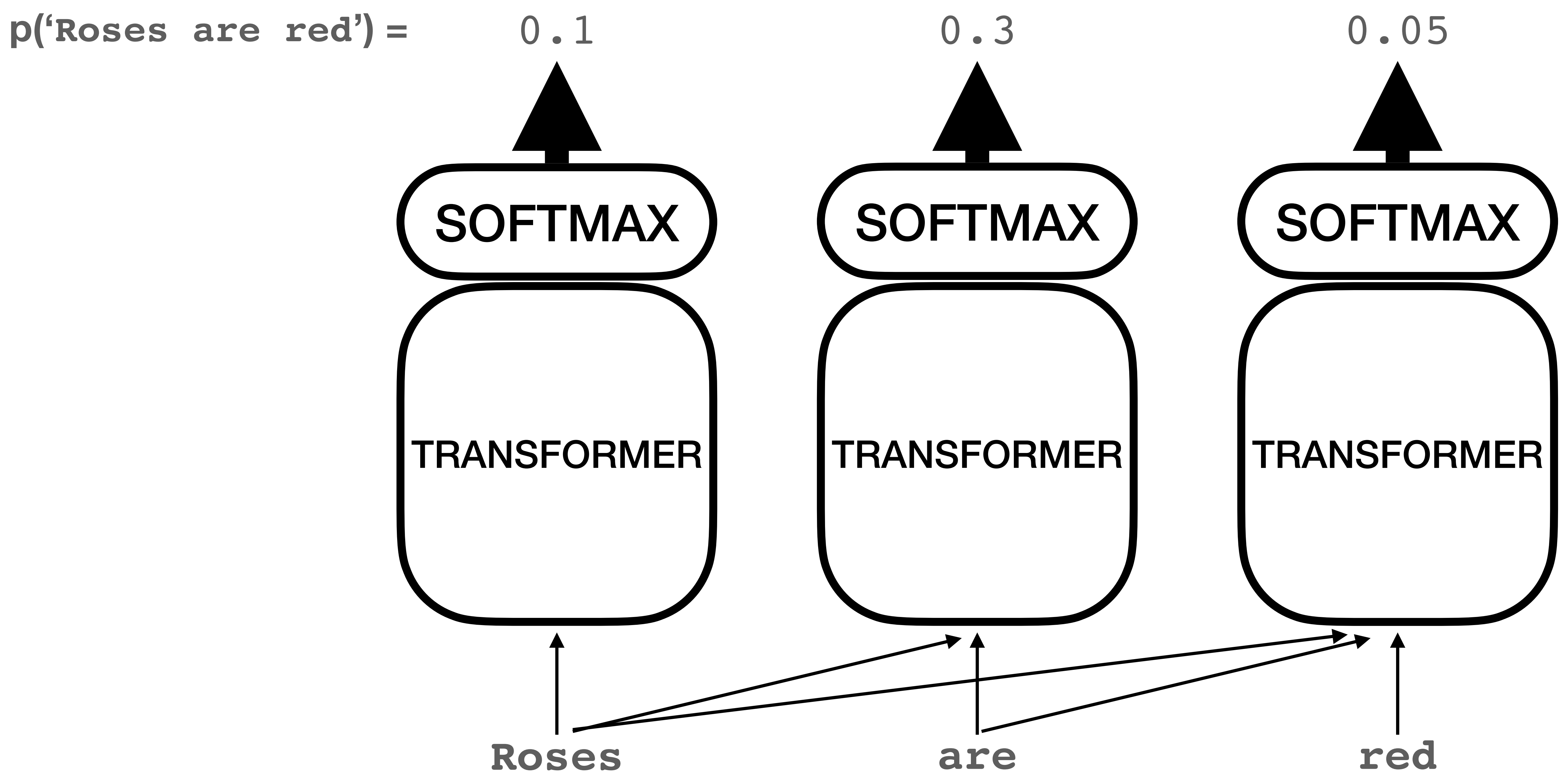
limitations of autoregressive models and their alternatives

Chu-Cheng Lin[♯], Aaron Jaech[♭], Xin Li[♯], Matt Gormley[♮], Jason Eisner[♯]

[♯]Johns Hopkins University
[♭]Facebook AI
[♮]Carnegie Mellon University

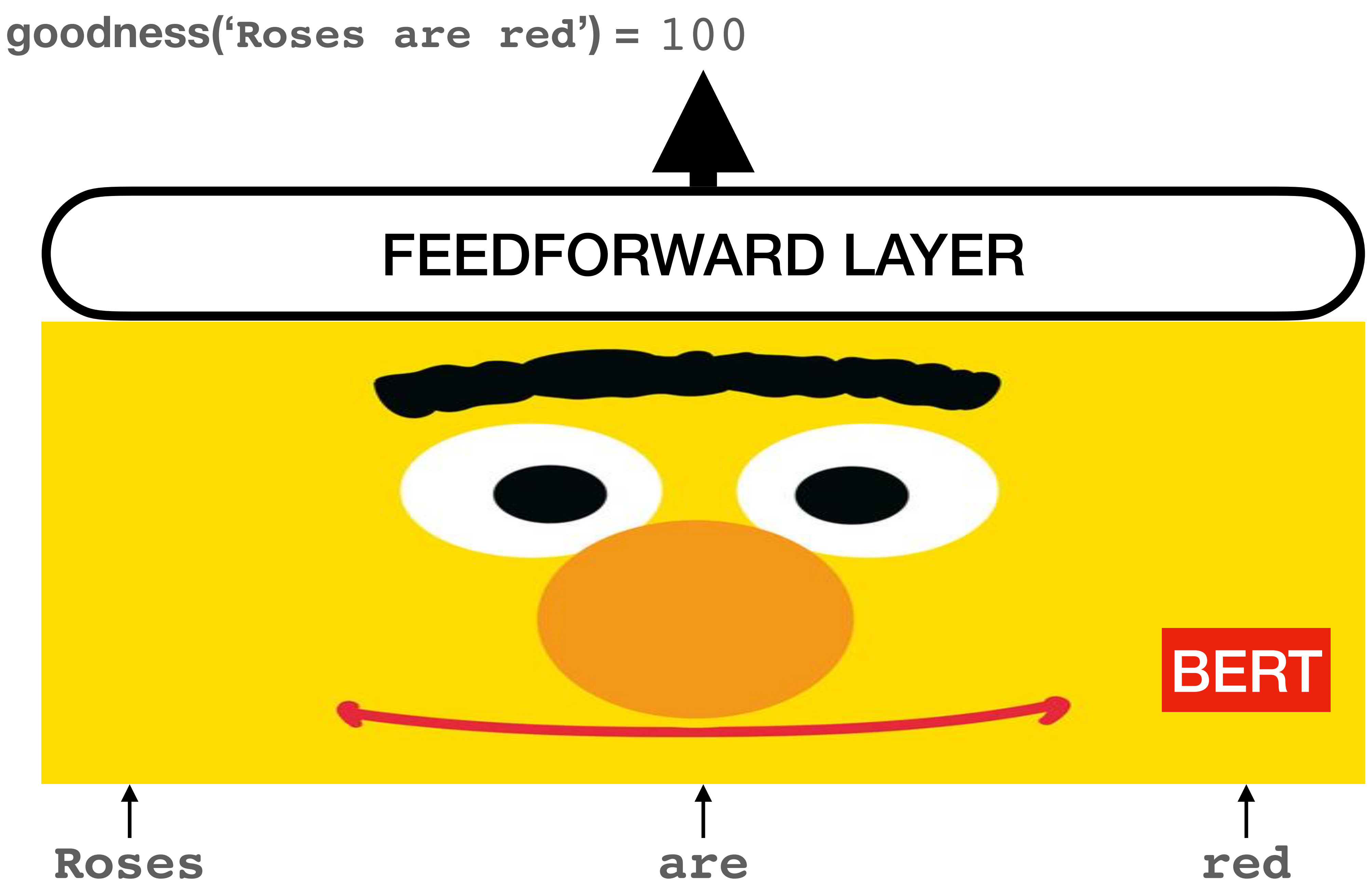
This talk

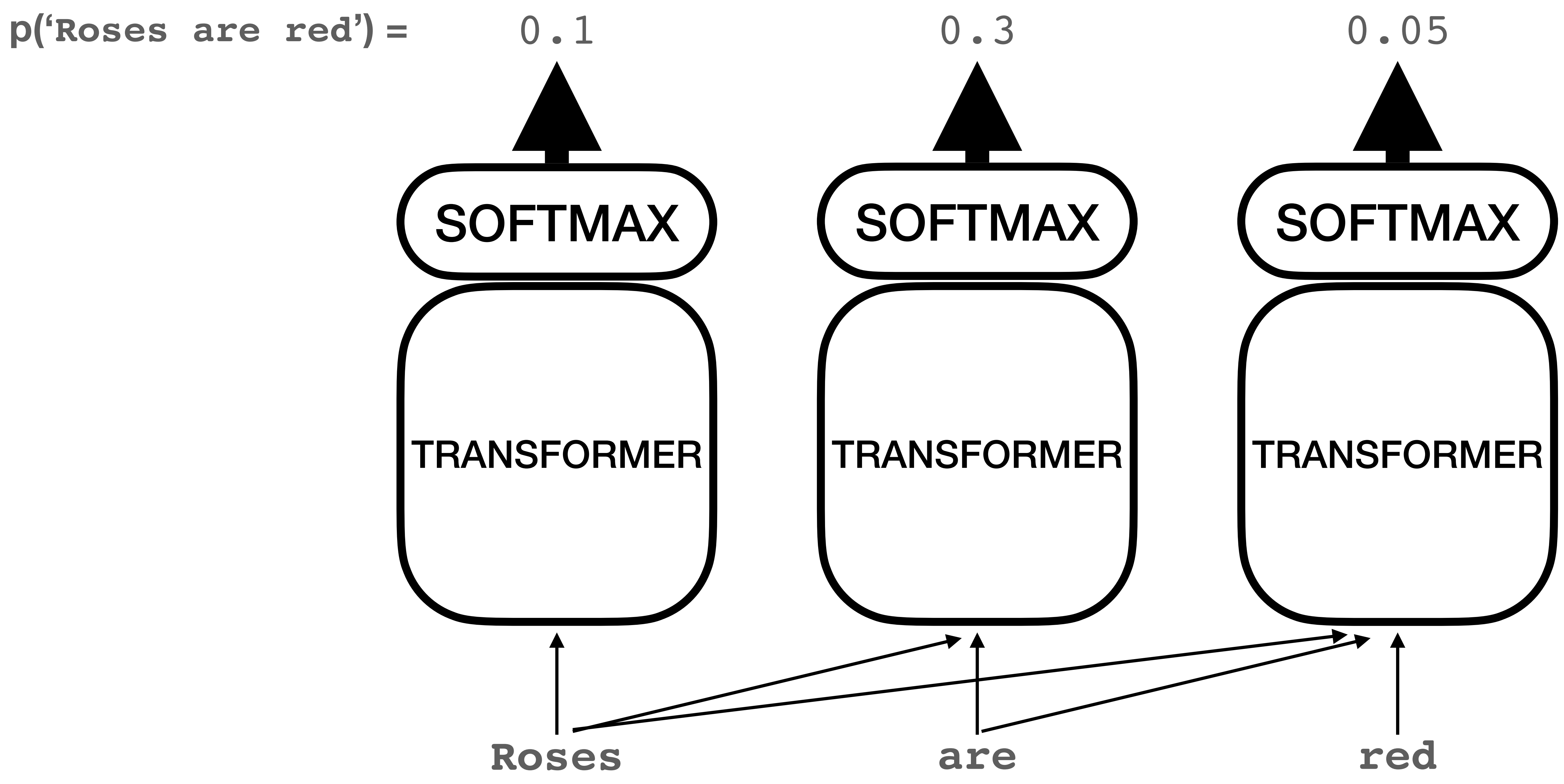
This talk



autoregressive models

This talk

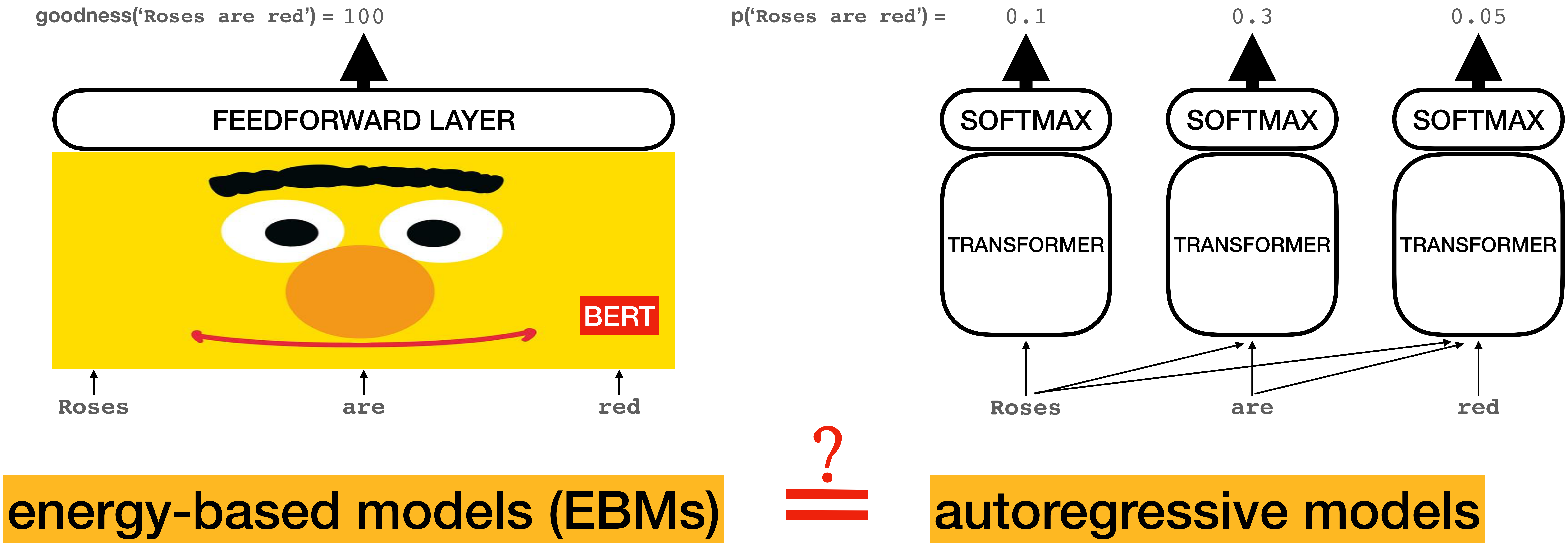




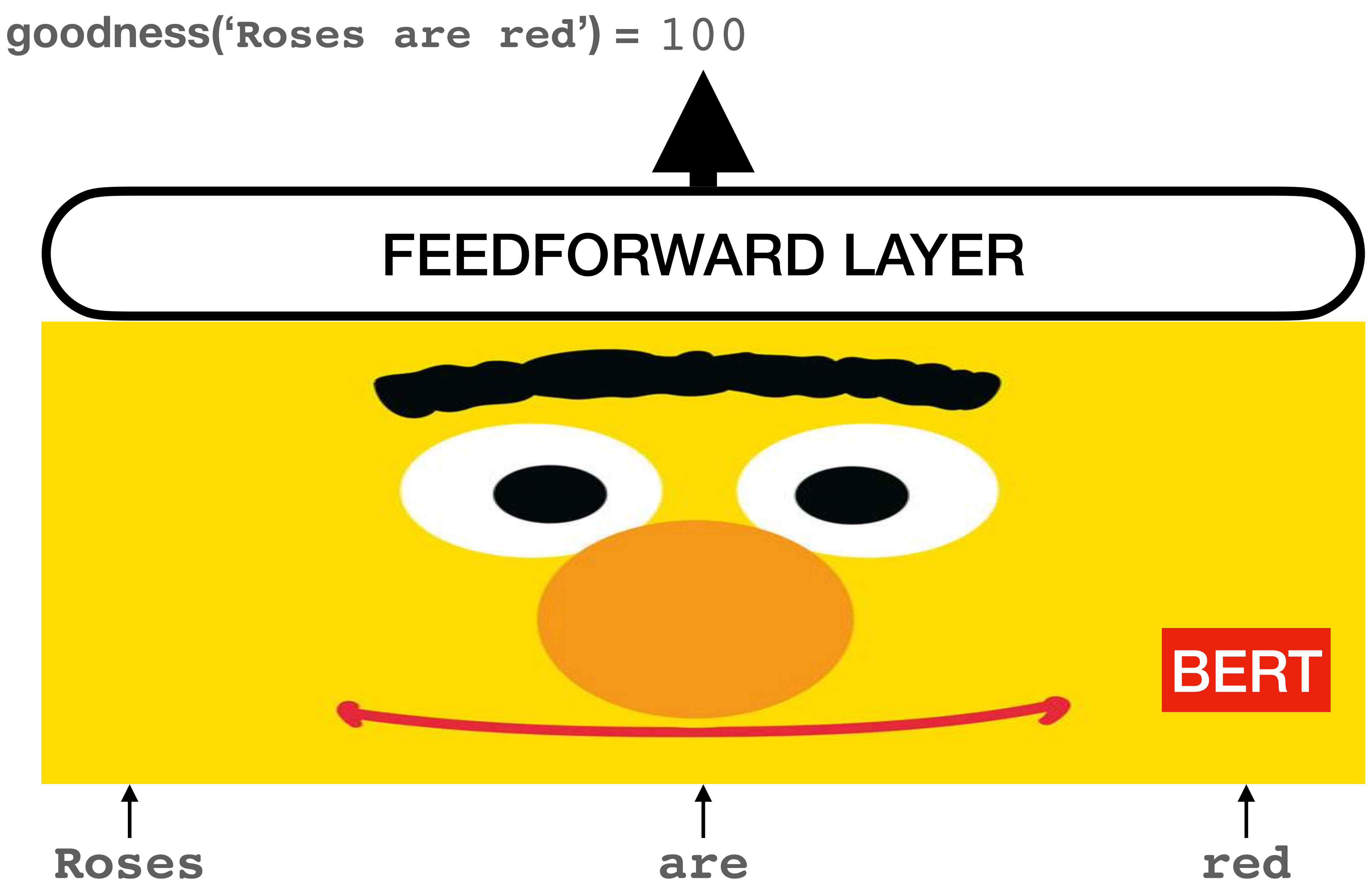
energy-based models (EBMs)

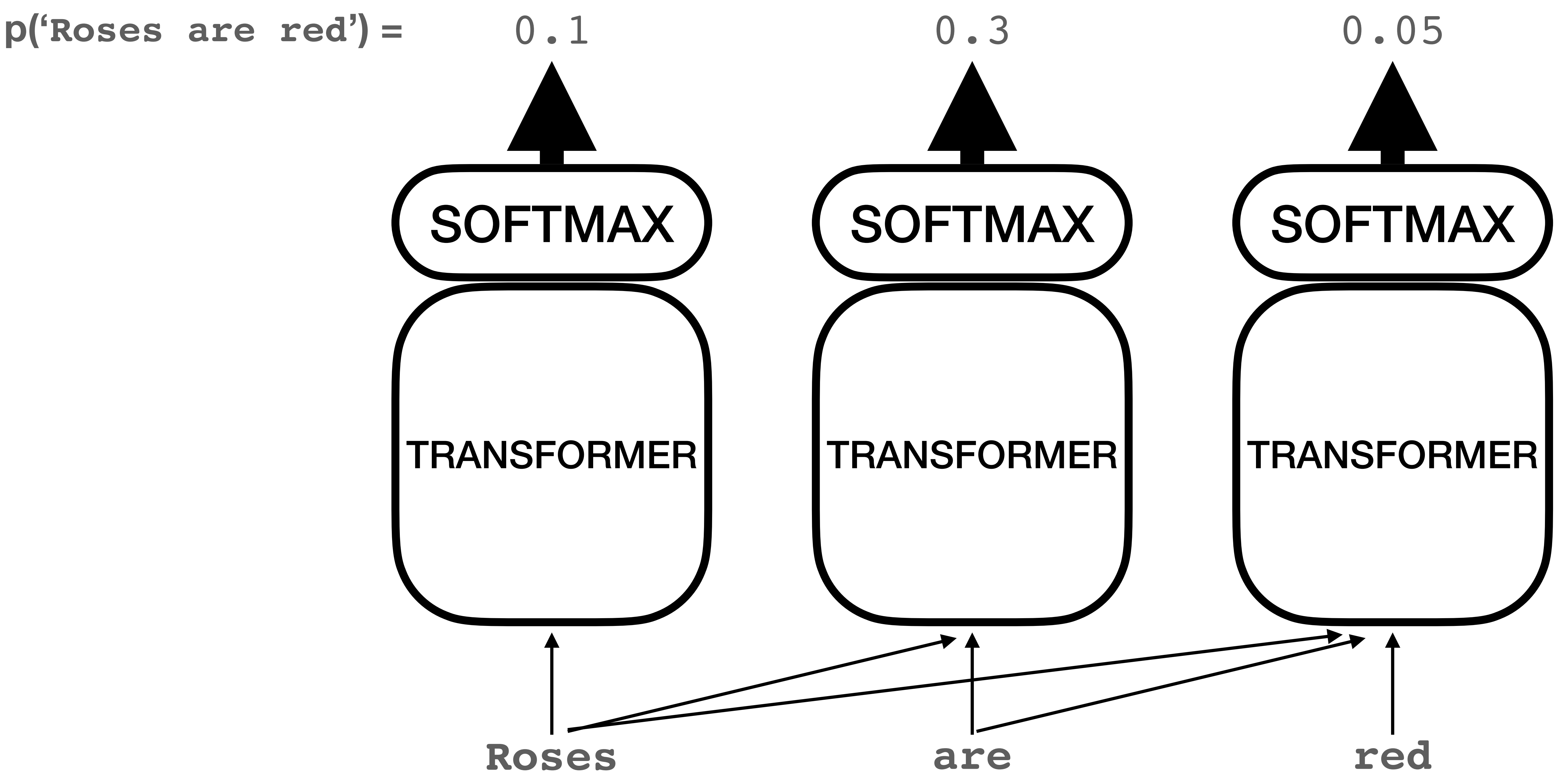
autoregressive models

This talk


goodness('Roses are red') = 100
FEEDFORWARD LAYER
BERT
Roses
are
red
p('Roses are red') =
0.1
0.3
0.05
SOFTMAX
SOFTMAX
SOFTMAX
TRANSFORMER
TRANSFORMER
TRANSFORMER
Roses
are
red
energy-based models (EBMs)
autoregressive models

This talk





energy-based models (EBMs) $\supsetneq$ autoregressive models

Outline

- Autoregressive models are not as expressive as other model families, energy-based models in particular.
    - And having more parameters helps little!
- Model families that are more expressive than autoregressive models made their own trade-offs

Outline

- **Autoregressive models are not as expressive as other model families, energy-based models in particular.**
  - **And having more parameters helps little!**
- Model families that are more expressive than autoregressive models made their own trade-offs

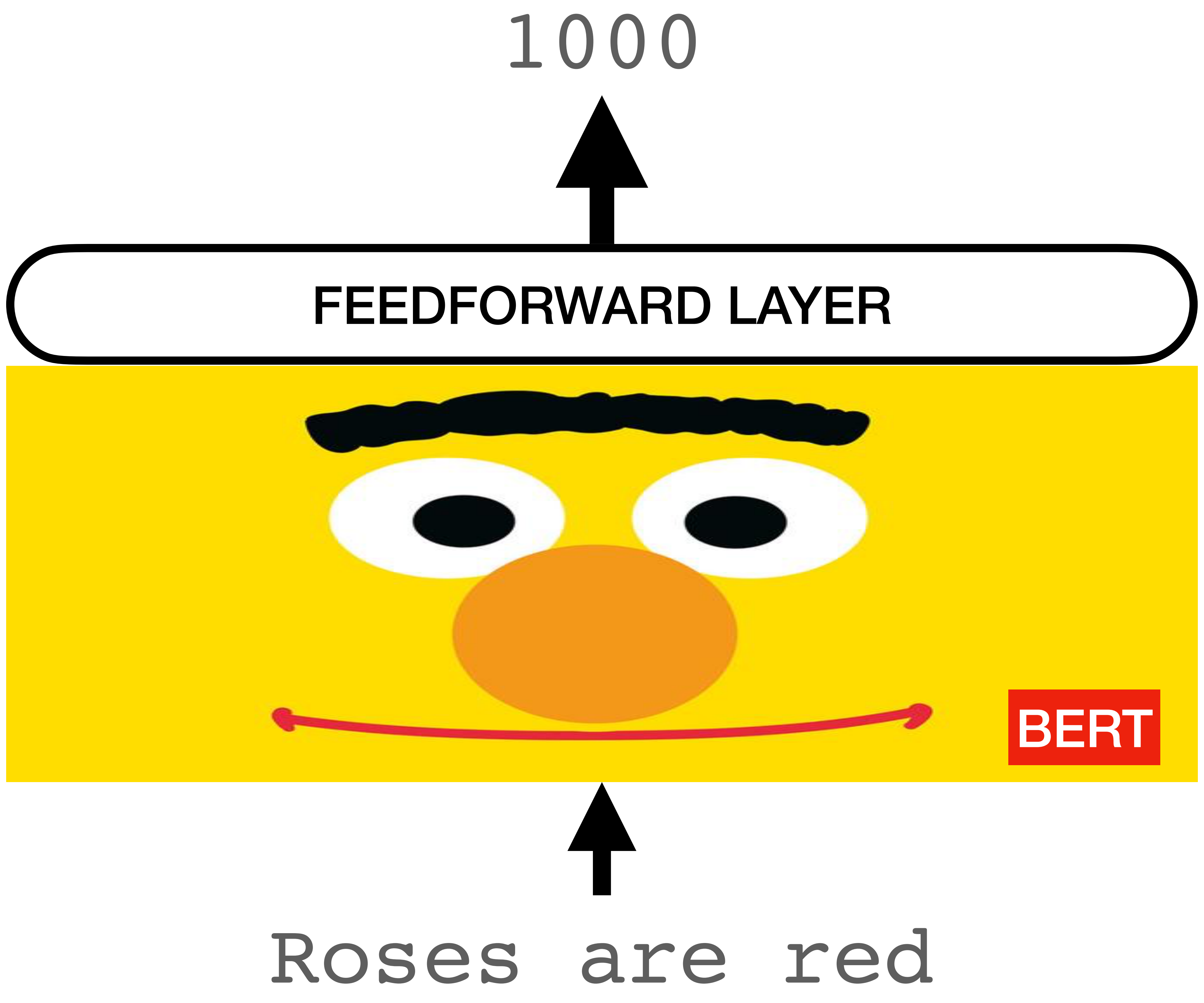
Many NLP tasks are about scoring strings

- Language modeling
  - **Good**:
    `Roses are red`
  - **Maybe**:
    `Roses are nosy`
  - **Bad**:
    `Roses queen sierra`
- Machine translation
  - **Good**:
    `Roses are red -> Las rosas son` **`rojas`**
  - **Bad**:
    `Roses are red -> Las rosas son` **`rojos`**

Many NLP tasks are about scoring strings

we want to measure their goodness quantitatively with an NN

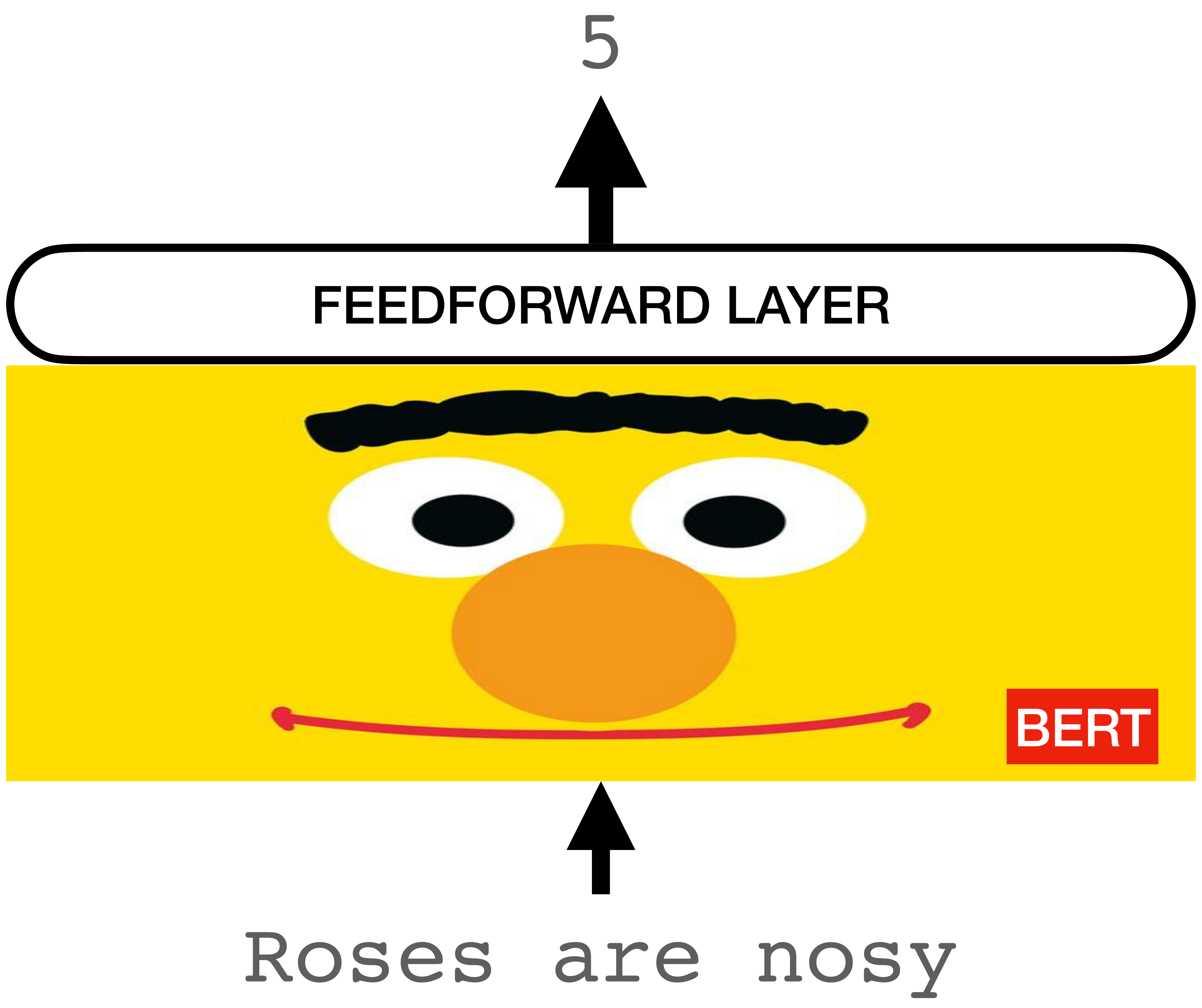
- **Good**:
  `Roses are red`
- **Maybe**:
  `Roses are nosy`
- **Bad**:
  `Roses queen sierra`



Many NLP tasks are about scoring strings

we want to measure their goodness quantitatively with an NN

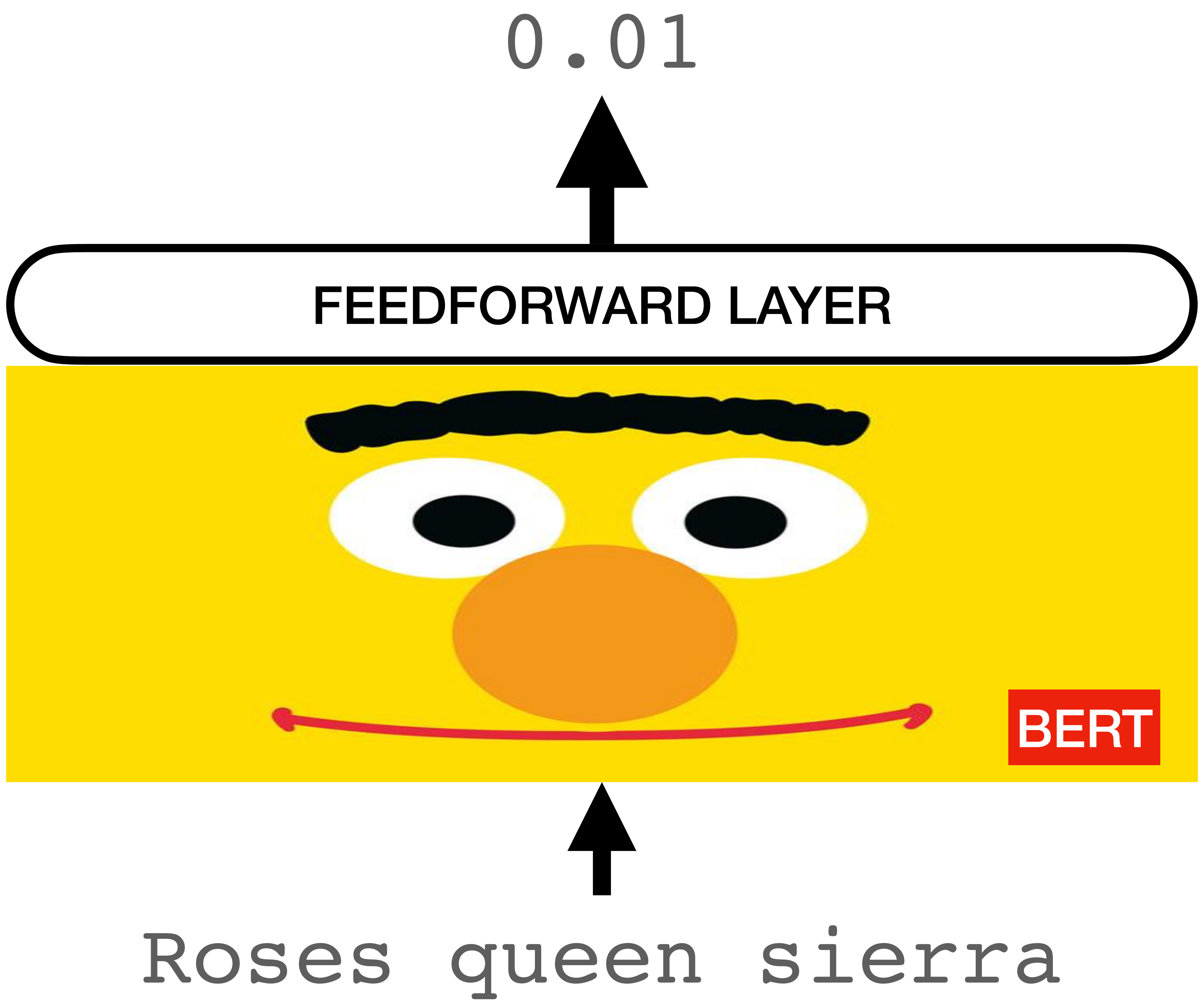
- **Good**:
  `Roses are red`
- **Maybe**:
  `Roses are nosy`
- **Bad**:
  `Roses queen sierra`



Many NLP tasks are about scoring strings

we want to measure their goodness quantitatively with an NN

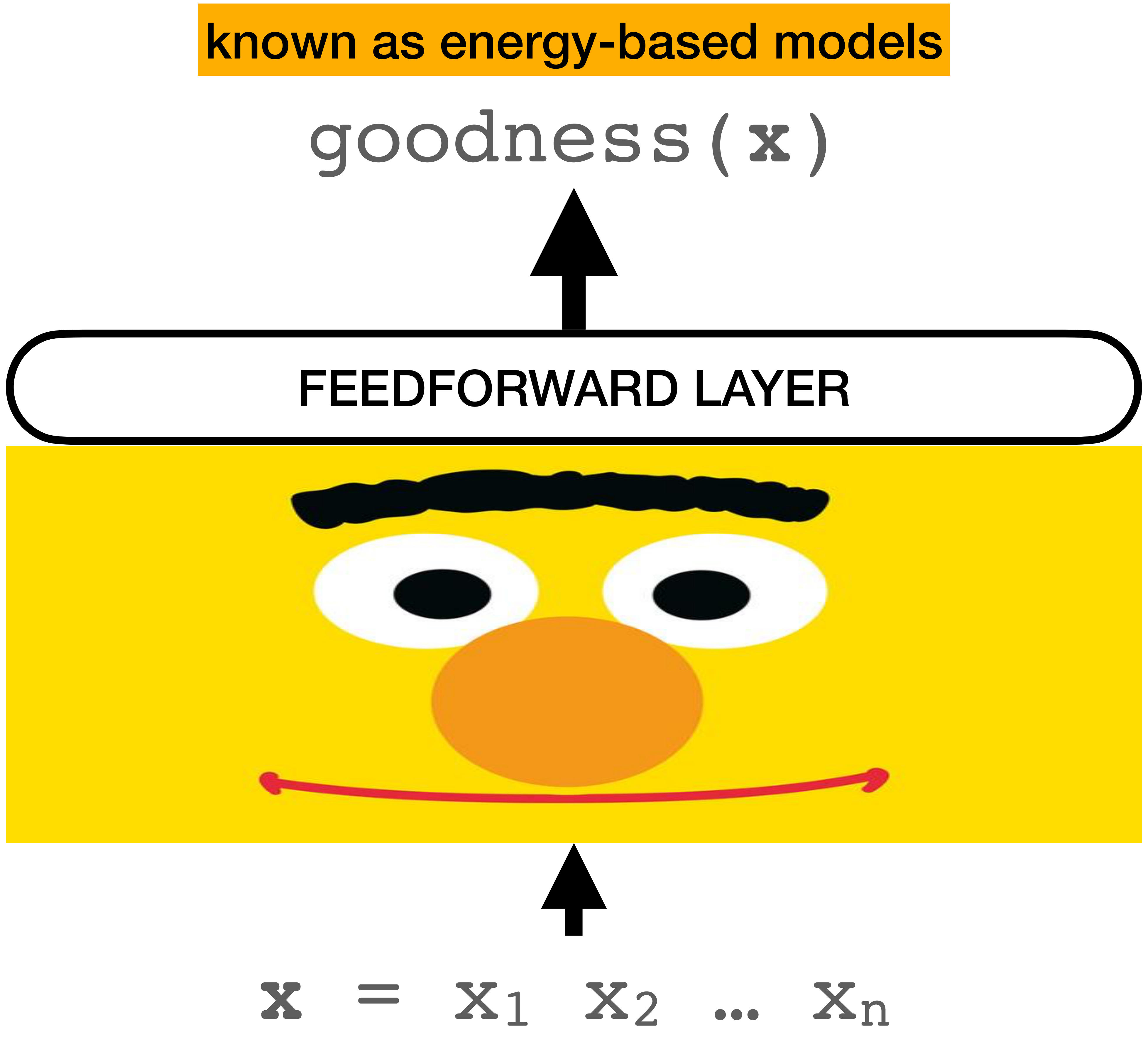
- **Good**:
  `Roses are red`
- **Maybe**:
  `Roses are nosy`
- **Bad**:
  `Roses queen sierra`



Many NLP tasks are about scoring strings

we want to measure their goodness quantitatively with an NN

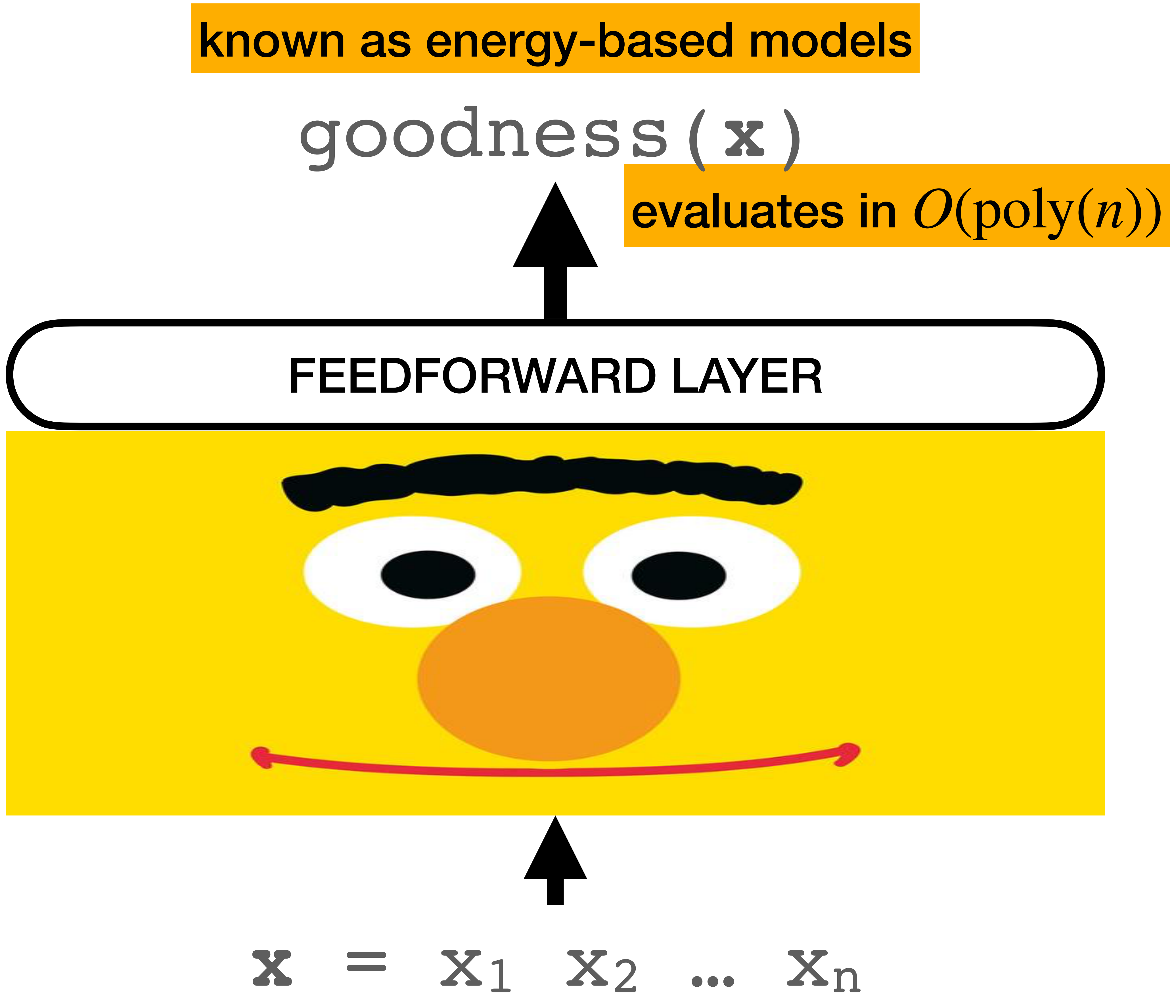
- goodness(“`Roses are red`”) = 1000
- goodness(“`Roses are nosy`”) = 5
- goodness(“`Roses queen sierra`”) = 0.01
- support of goodness:
    - set of strings whose goodness > 0



Many NLP tasks are about scoring strings

we want to measure their goodness quantitatively with an NN

- goodness(“`Roses are red`”) = 1000
- goodness(“`Roses are nosy`”) = 5
- goodness(“`Roses queen sierra`”) = 0.01
- support of goodness:
  - set of strings whose goodness > 0

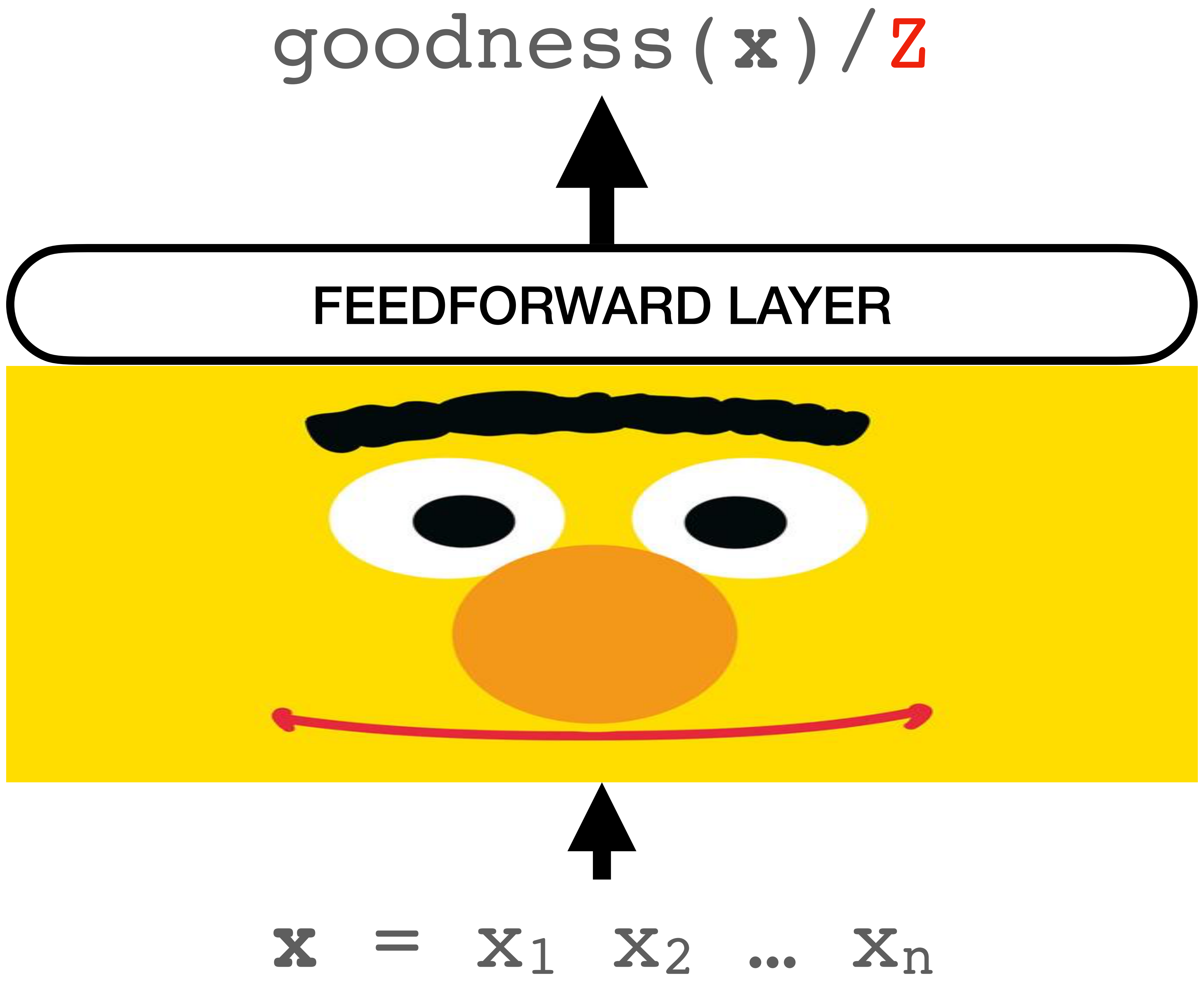


Many NLP tasks are about scoring strings

we want to measure their goodness on a scale between 0 and 1

- Z = goodness(“`Roses are red`”)
  + goodness(“`Roses are nosy`”)
  + goodness(“`Roses queen sierra`”)
  + …
  - intractable!

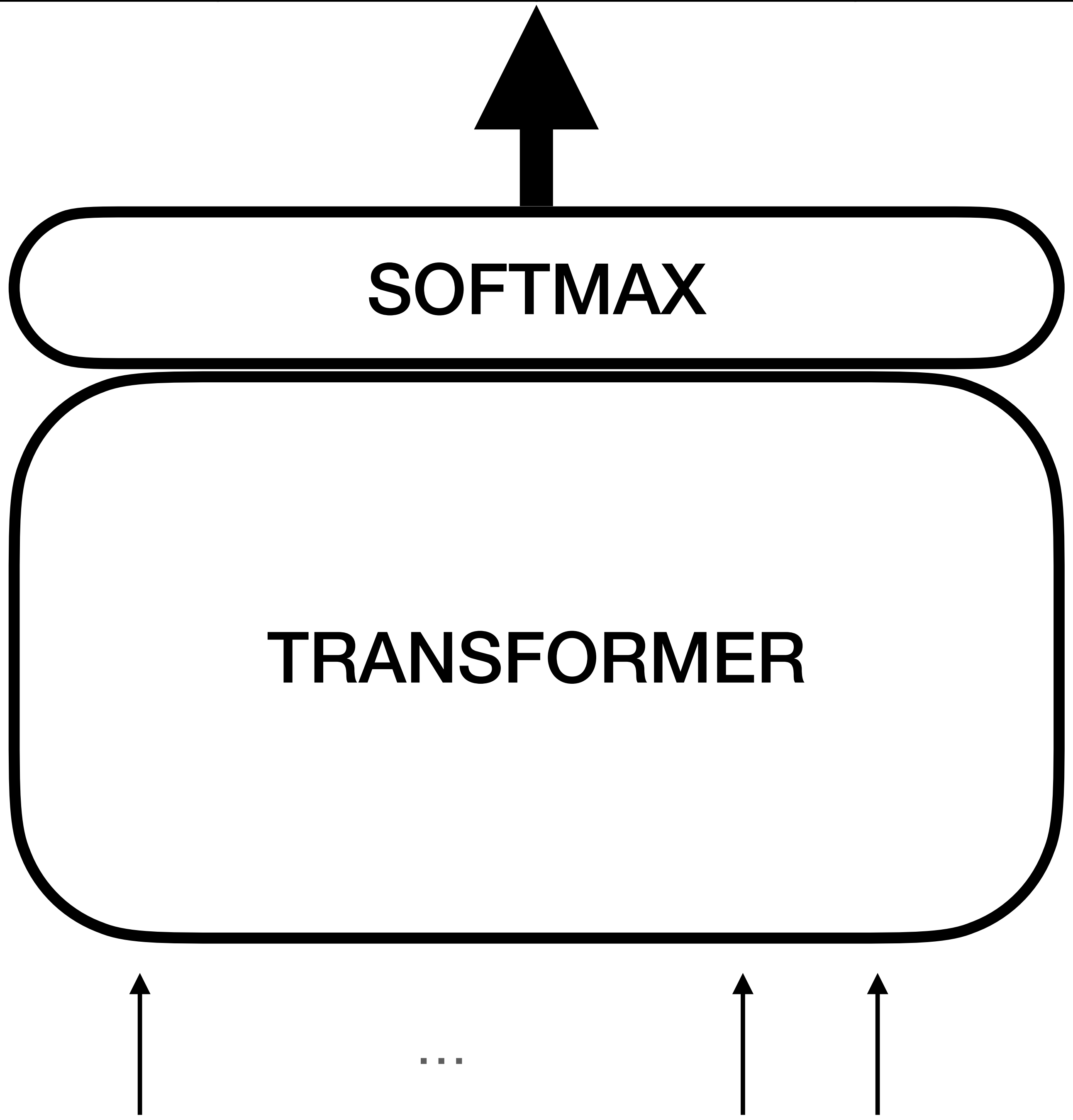
exponentially (or even infinitely) many columns!					
Roses are red	Roses are nosy	Roses queen	Las rosas son rojos	…	SUM
1000	5	0.01	0.00001	…	1000+5+0.01+0.00001+…



Autoregressive parametrization

- goodness(“`Roses are red`”) =…

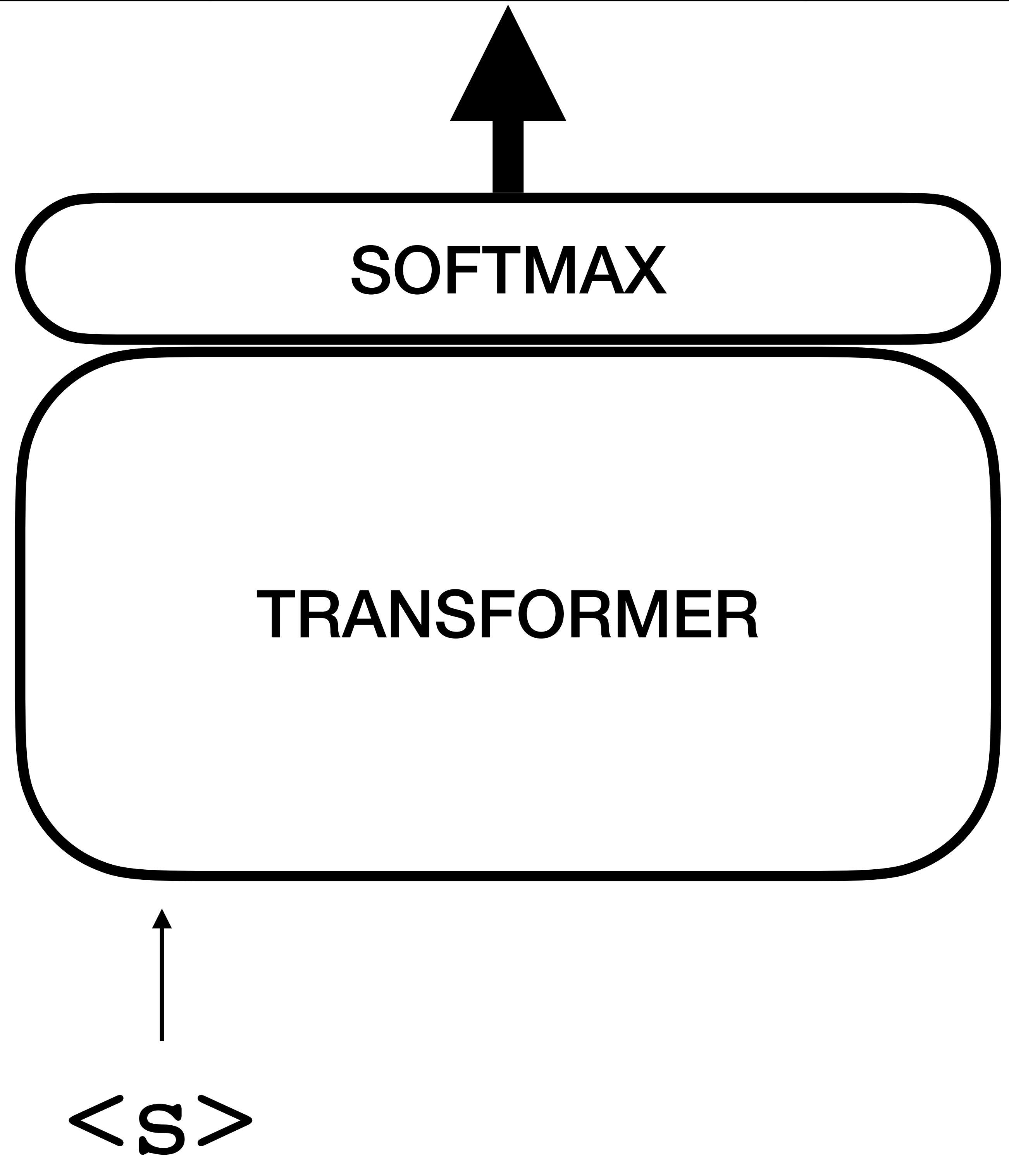
# of columns: vocabulary size					
red	**Roses**	**are**	**sierra**	**…**	**SUM**
…	…	…	…	…	**1**



Autoregressive parametrization

- goodness(“`Roses are red`”) =
  p(“`Roses`”) …
  = 0.1…

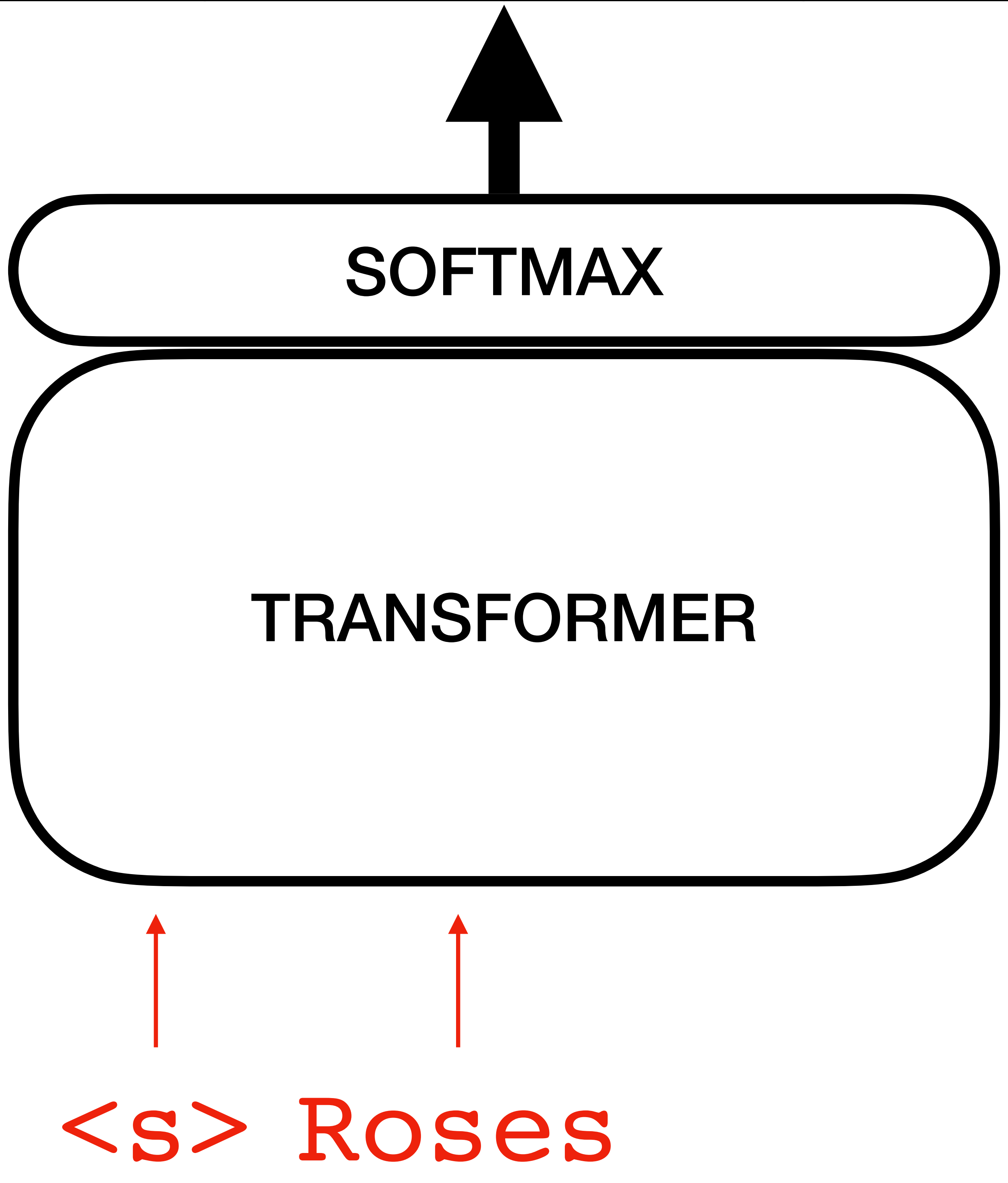
red	Roses	are	sierra	...	SUM
0.001	0.1	0.002	0.0001	…	1



Autoregressive parametrization

- goodness(“Roses are red”) =
  p(“Roses”) *
  p(“are” | “Roses”) …
  = 0.1 * 0.3 …

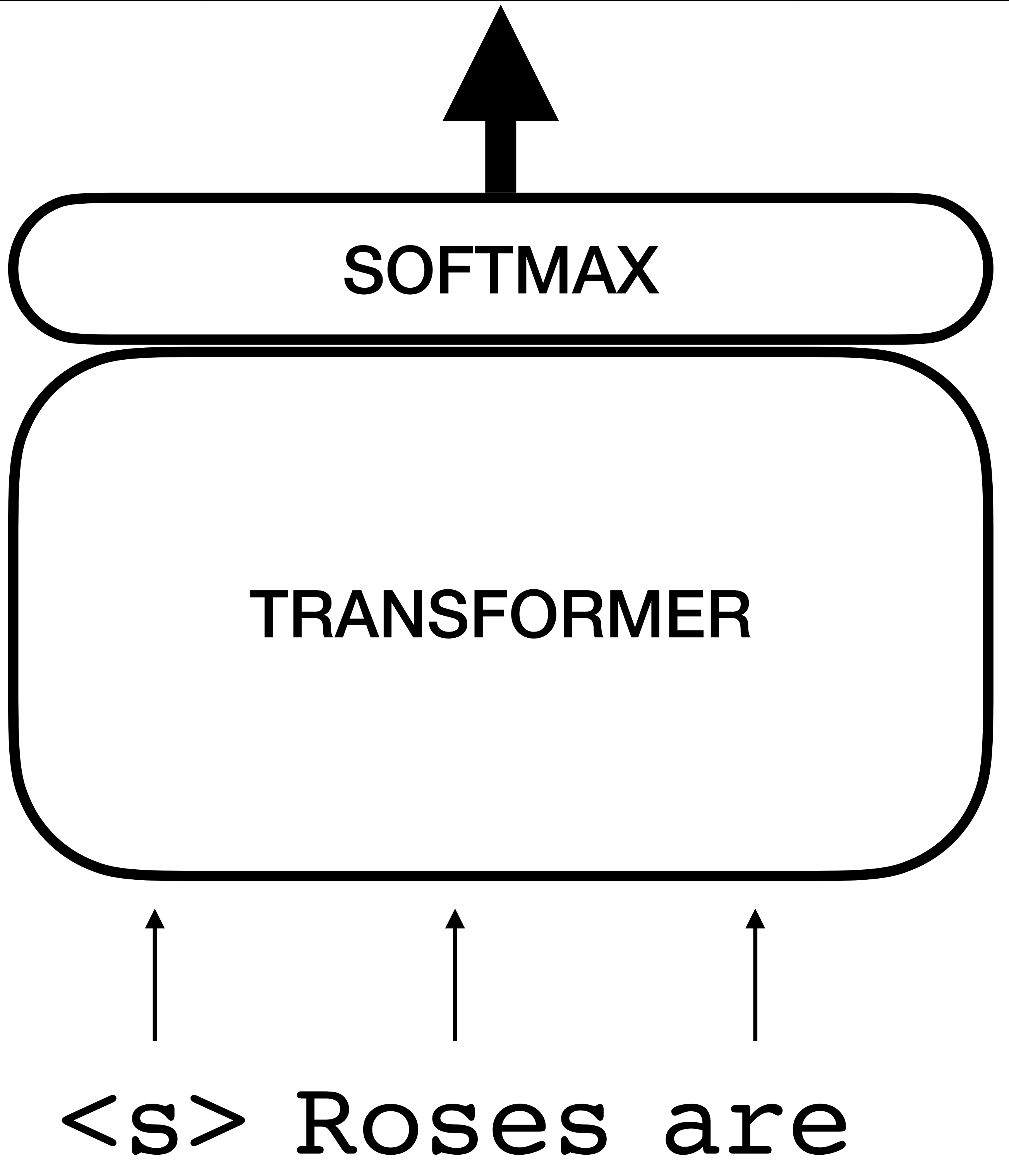
red	Roses	are	sierra	…	SUM
0.001	0.002	0.3	0.00002	…	1



Autoregressive parametrization

- goodness(“`Roses are red`”) =
  p(“`Roses`”) *
  p(“`are`” | “`Roses`”) *
  p(“`red`” | “`Roses are`”)
  = 0.1 * 0.3 * 0.05

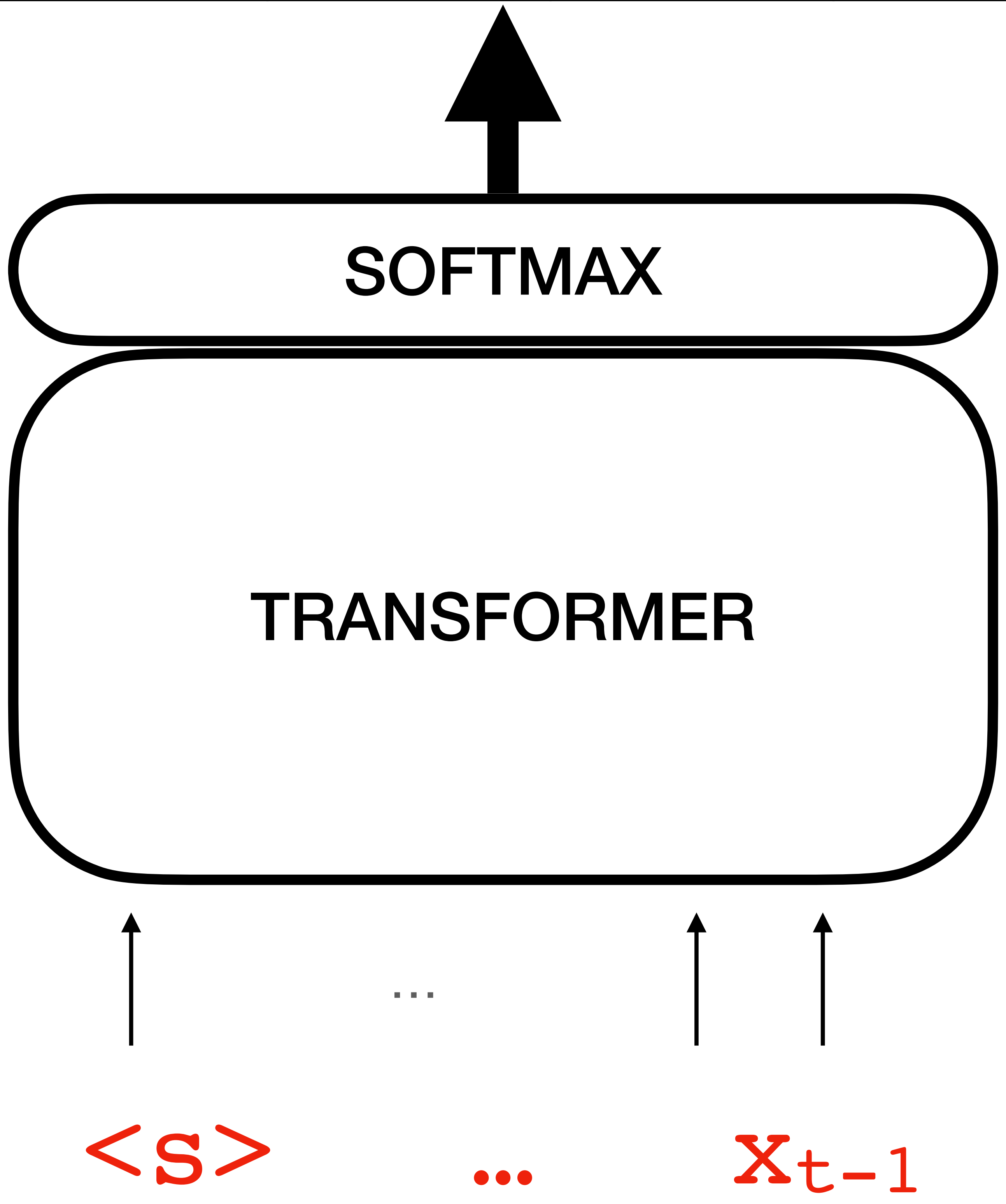
red	Roses	are	sierra	…	SUM
0.05	0.02	0.0007	0.0006	…	1



Autoregressive parametrization

- goodness(“Roses are red”) =0.0015
- goodness(“Roses are nosy”) =0.0000076
- goodness(“Roses queen sierra”) =0.000000015

x_t=	red	Roses	are	…
$p(x_t \| \mathbf{x}_{<t})$=				…

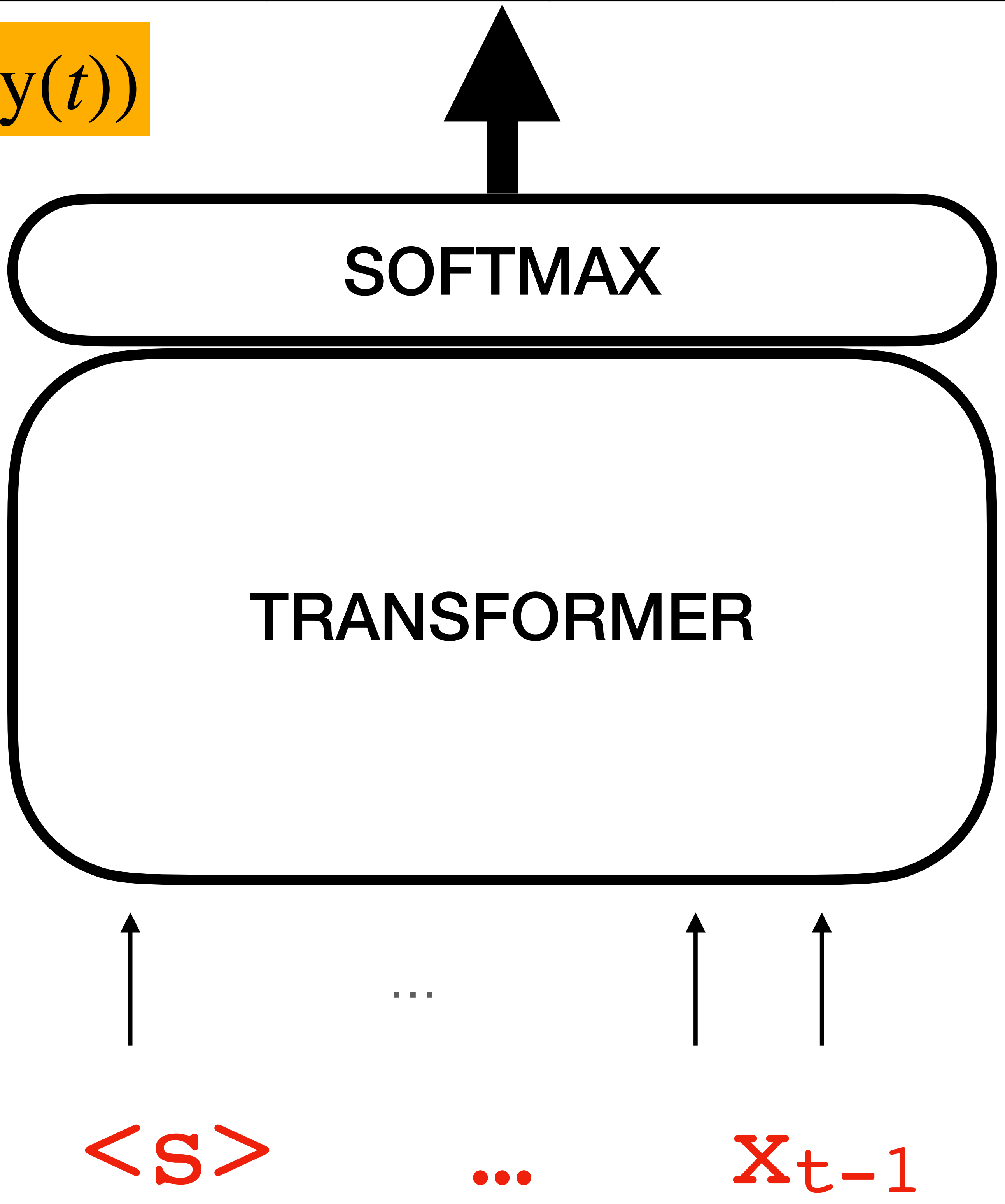


Autoregressive parametrization

- goodness("Roses are red") =0.0015
- goodness("Roses are nosy") =0.0000076
- goodness("Roses queen sierra") =0.000000015

x_t=	red	Roses	are	…
$p(x_t \mid \mathbf{x}_{<t})$=				…

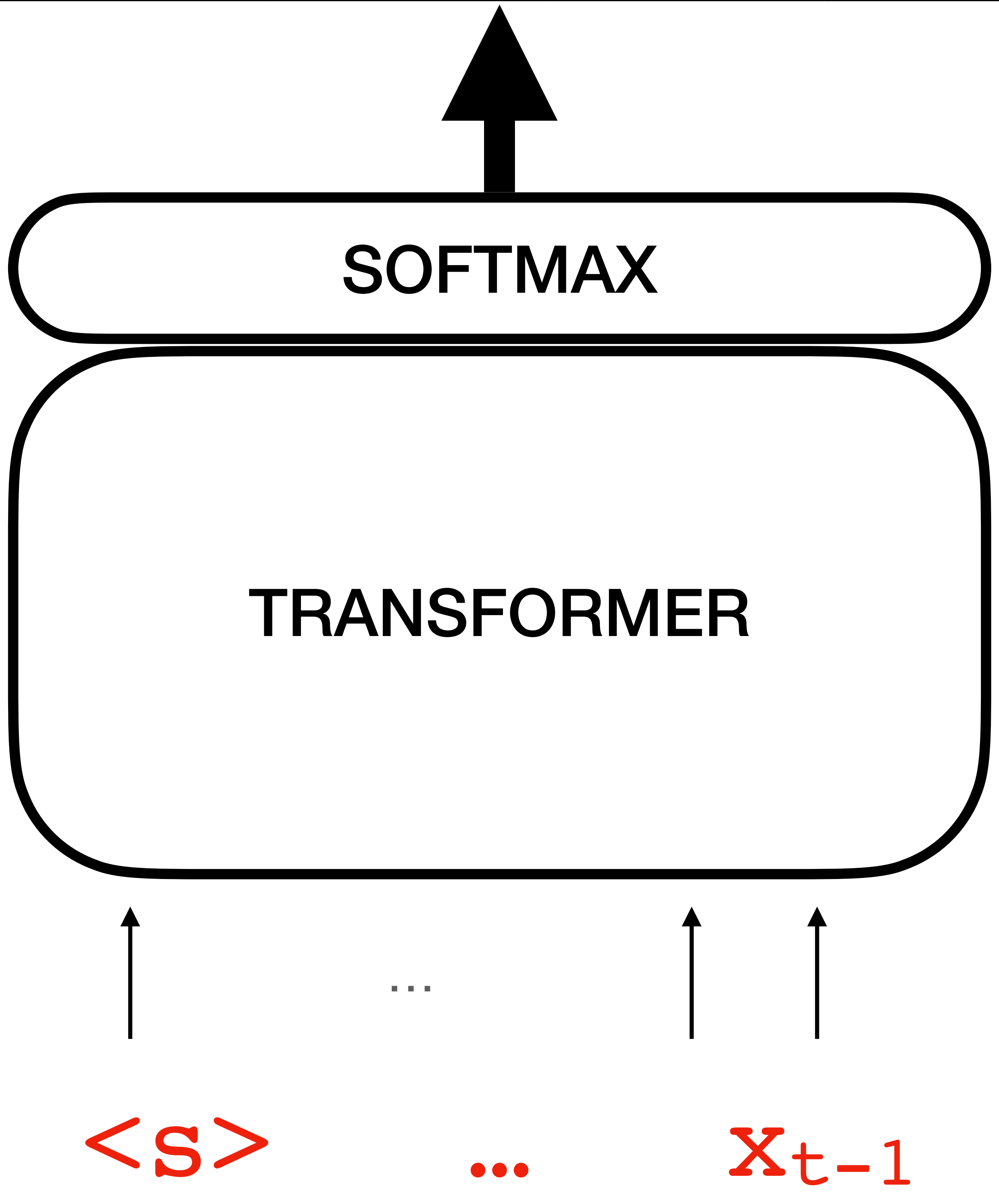
evaluates in $O(\mathrm{poly}(t))$



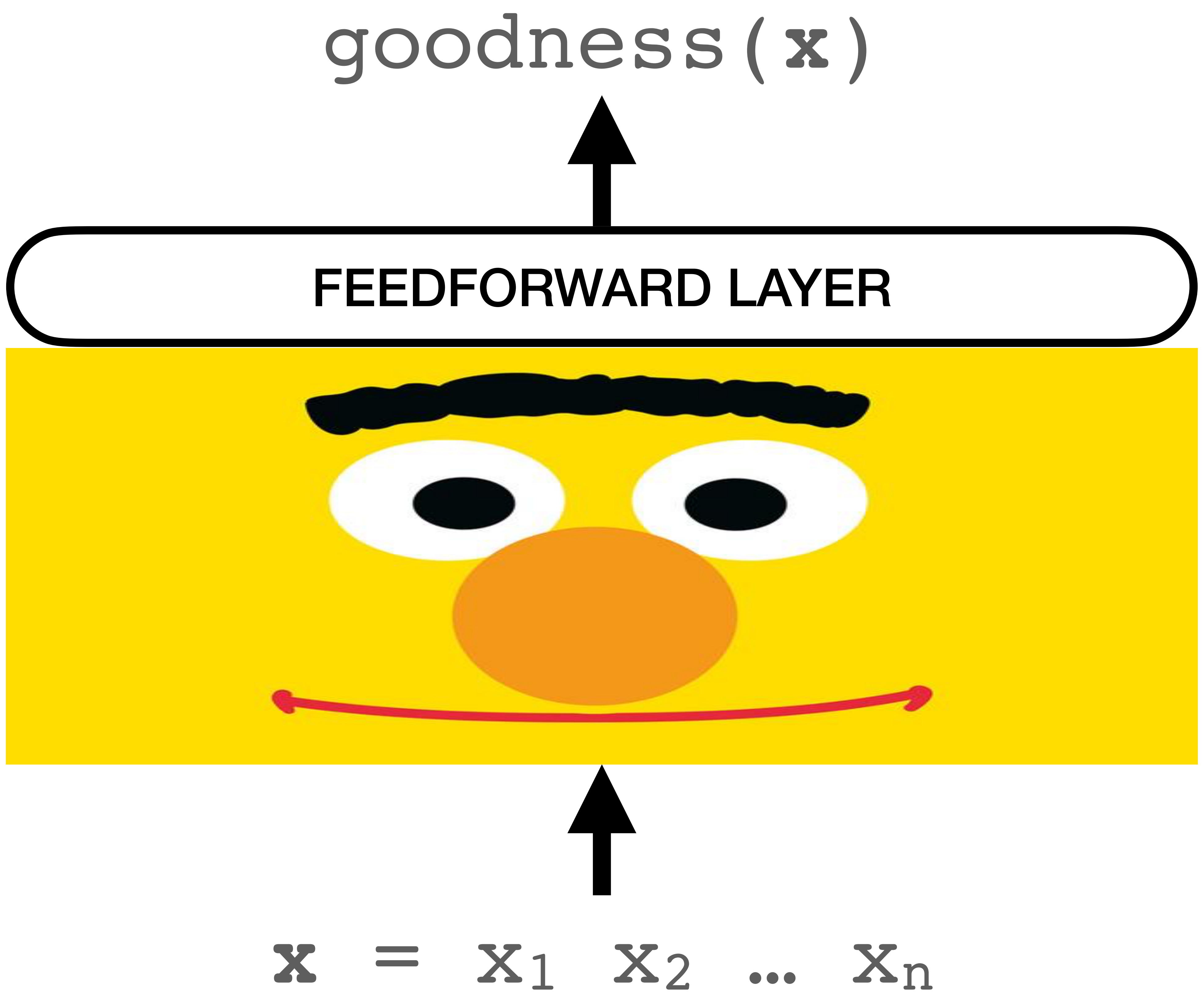
Autoregressive parametrization

- Autoregressive models guarantee Z = 1
- goodness(“Roses are red”) = 0.0015/Z = 0.0015
- goodness(“Roses are nosy”) = 0.0000076/Z = 0.0000076
- goodness(“Roses queen sierra”) = 0.000000015/Z = 0.000000015

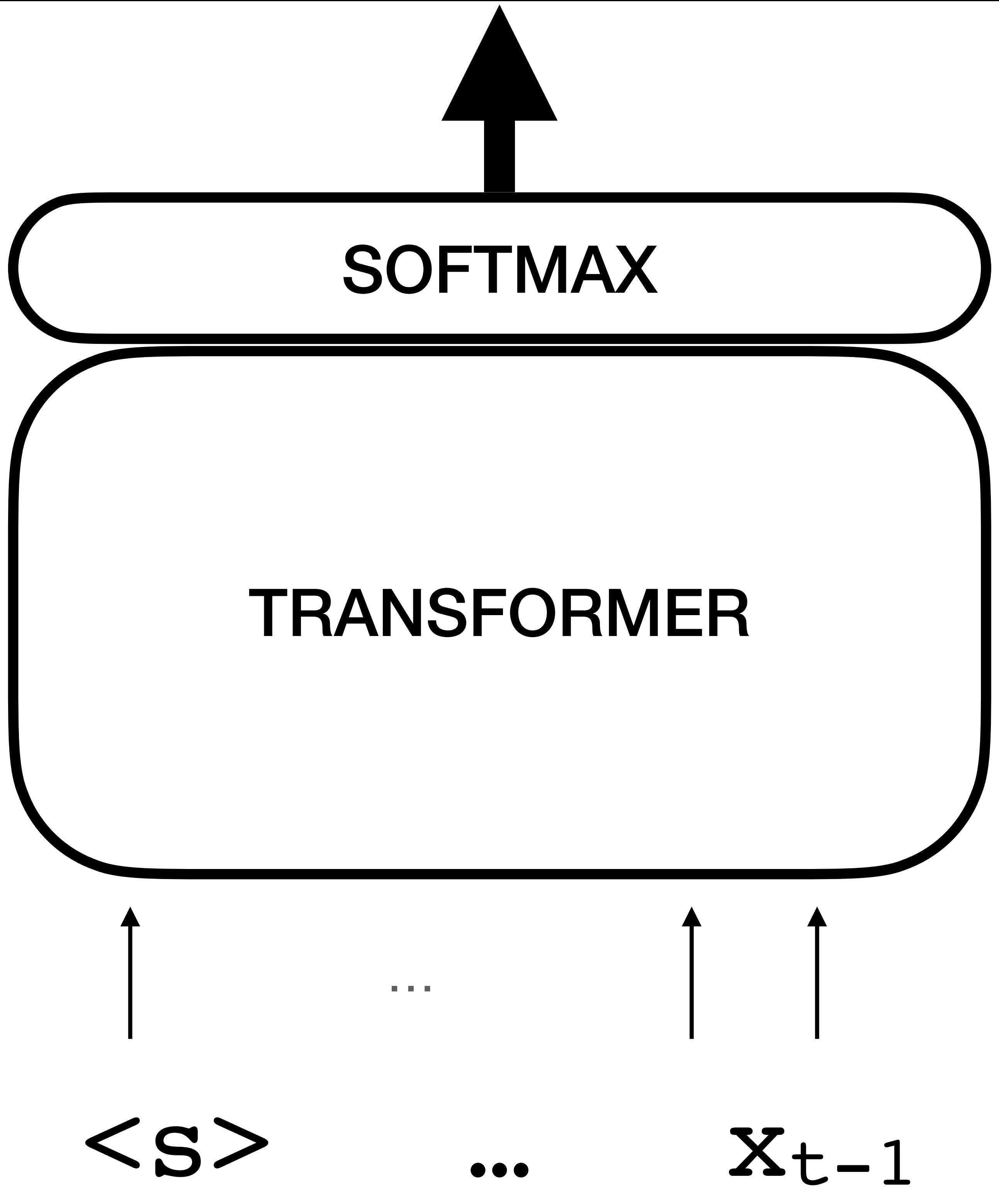
x_t=	red	Roses	are	…
$p(x_t \mid \mathbf{x}_{<t})$=				…



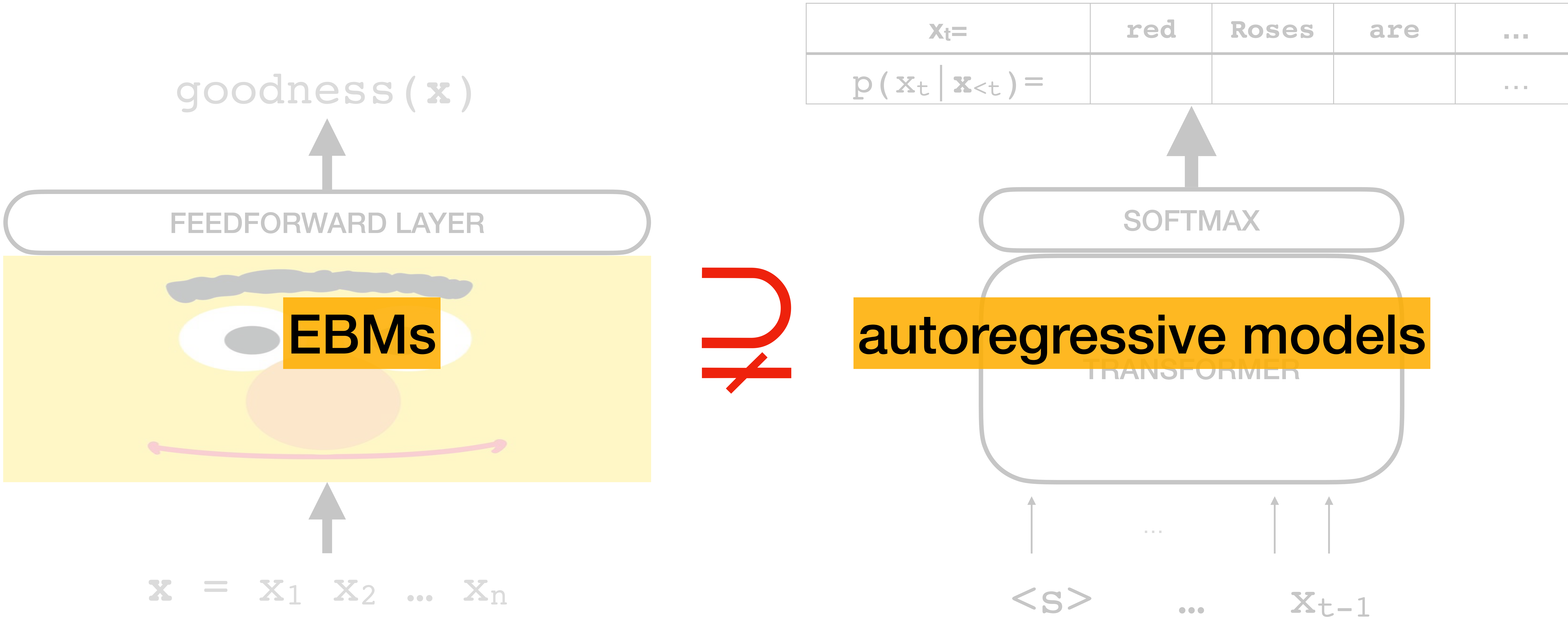
EBMs vs autoregressive models



x_t=	red	Roses	are	...
$p(x_t \mid \mathbf{x}_{<t})$=				...



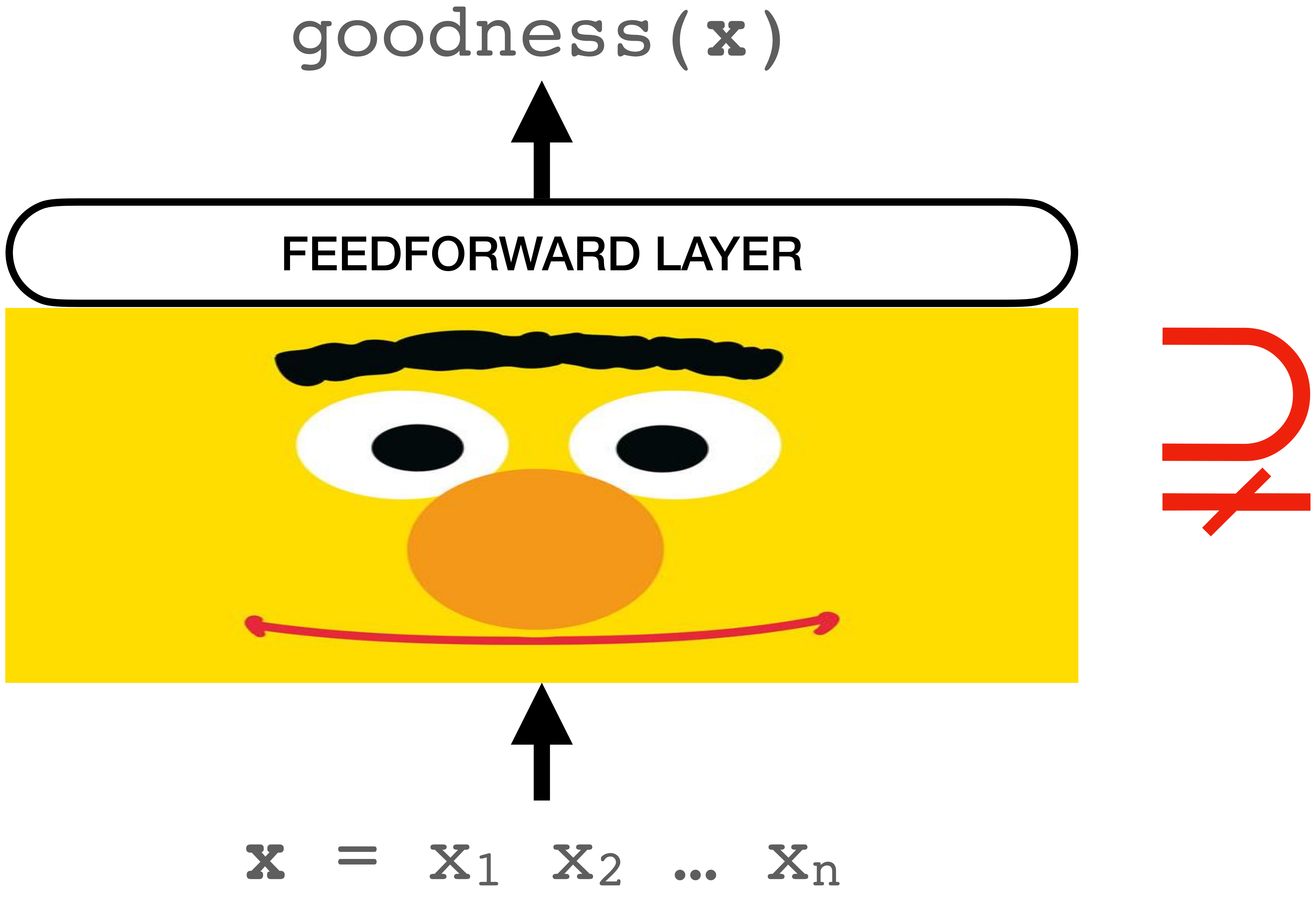
EBMs are more powerful!



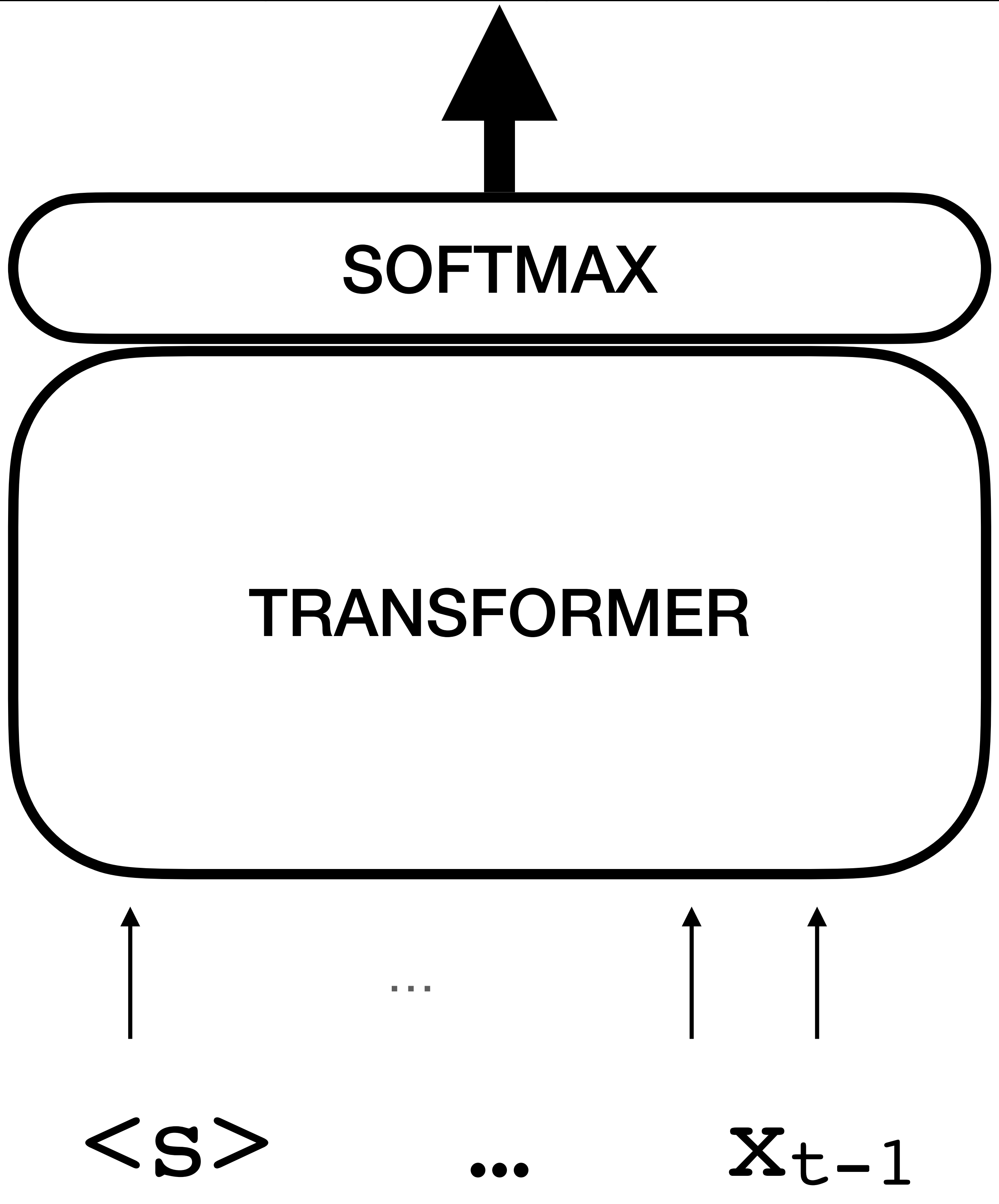
This work

lookup models $\supsetneq$ (autoregressive) latent variable models $\supsetneq$ EBMs $\supsetneq$ autoregressive models

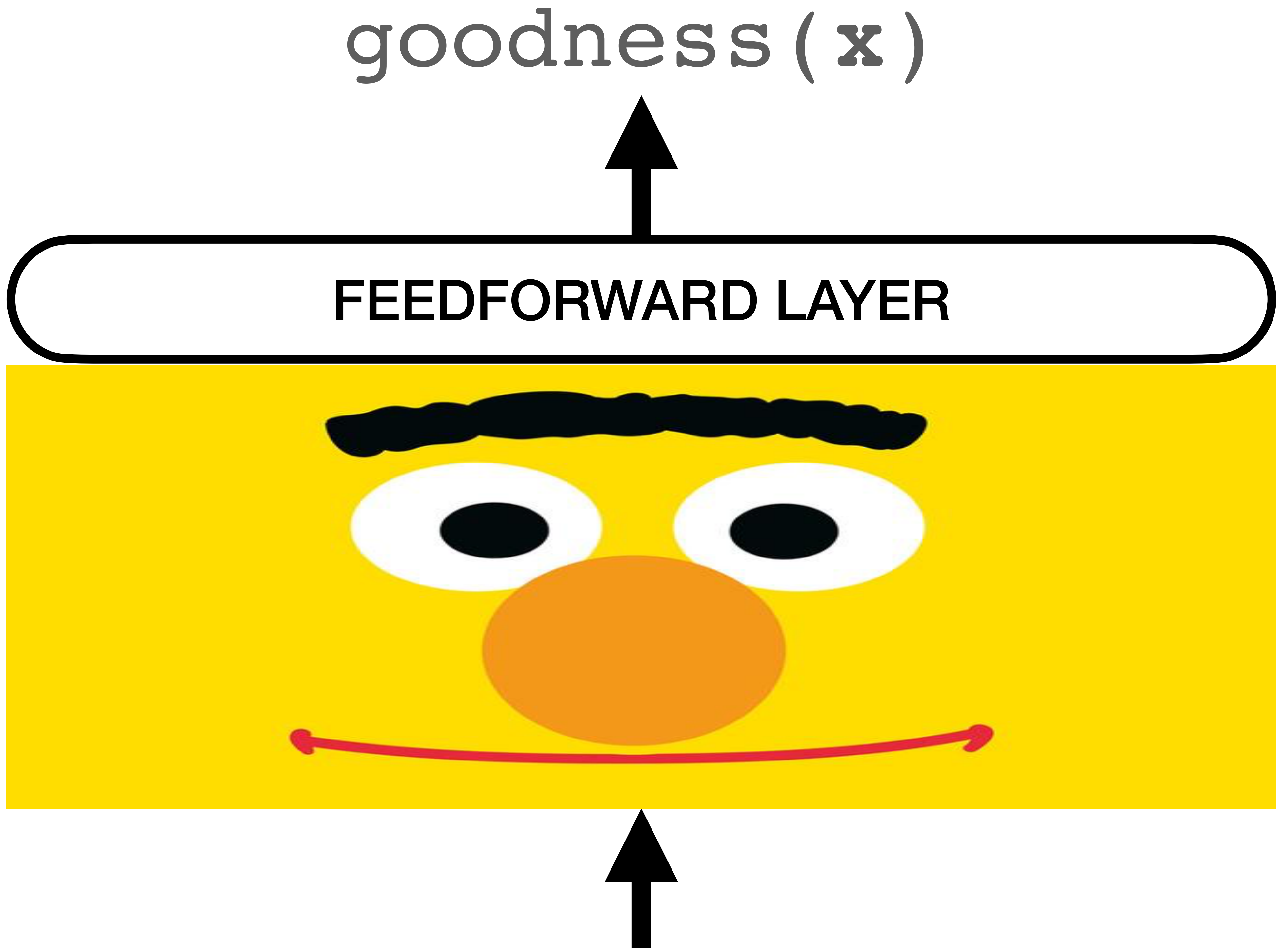
Why EBMs $\supsetneq$ autoregressive models?



x_t=	red	Roses	are	…
$p(x_t \mid \mathbf{x}_{<t})$=				…



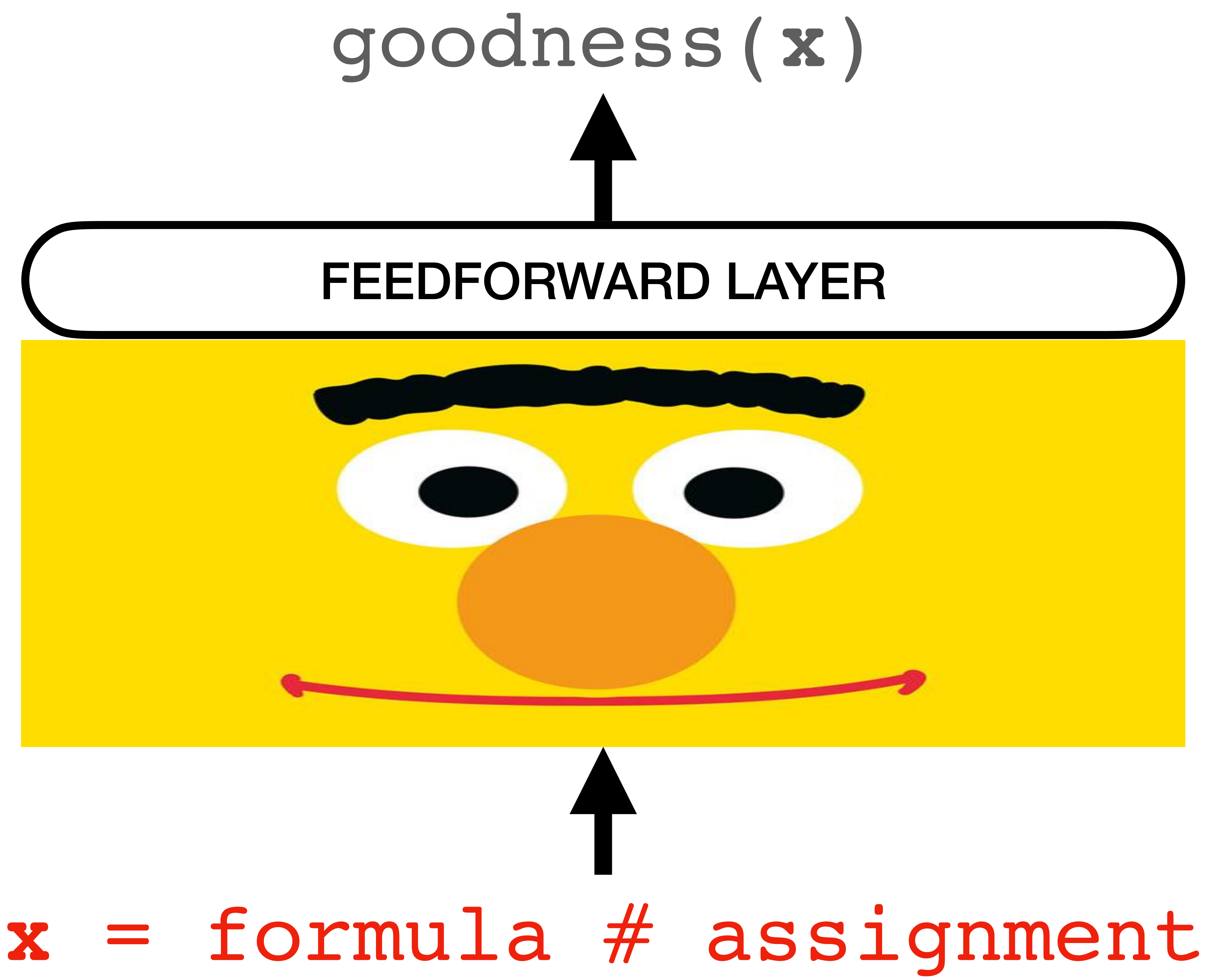
Why EBMs $\supsetneq$ autoregressive models?



`x = formula # assignment`

- formula:
  `(A₁ or not A₂) and (A₃)`
- assignment:
  `101`

Why EBMs $\supsetneq$ autoregressive models?



- formula:
  $(A_1 \texttt{ or not } A_2) \texttt{ and } (A_3)$
- assignment:
  `101`
- goodness(**x**)
  - > 0 if assignment satisfies formula
  - =0 otherwise

Why EBMs $\supsetneq$ autoregressive models?

`goodness(x)`

FEEDFORWARD LAYER

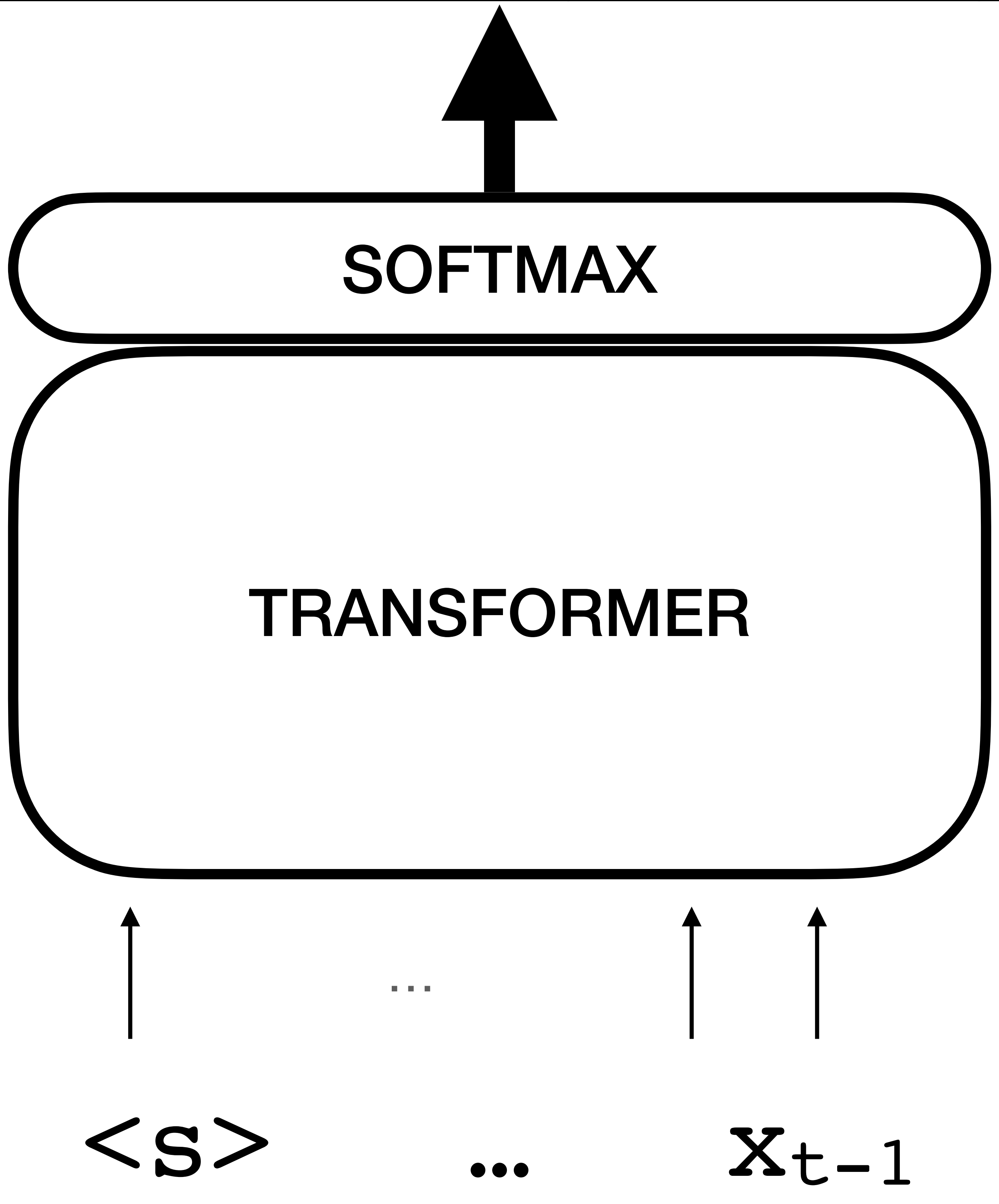
`x = formula # assignment`

- formula:
  `(A₁ or not A₂) and (A₃)`
- assignment:
  `101`
- goodness(**x**)
  - >0 if assignment satisfies formula
  - $=0$ otherwise
- goodness(**x**) can be constructed as an RNN with size $O(|\mathbf{x}|^3)$

Why EBMs $\supsetneq$ autoregressive models?

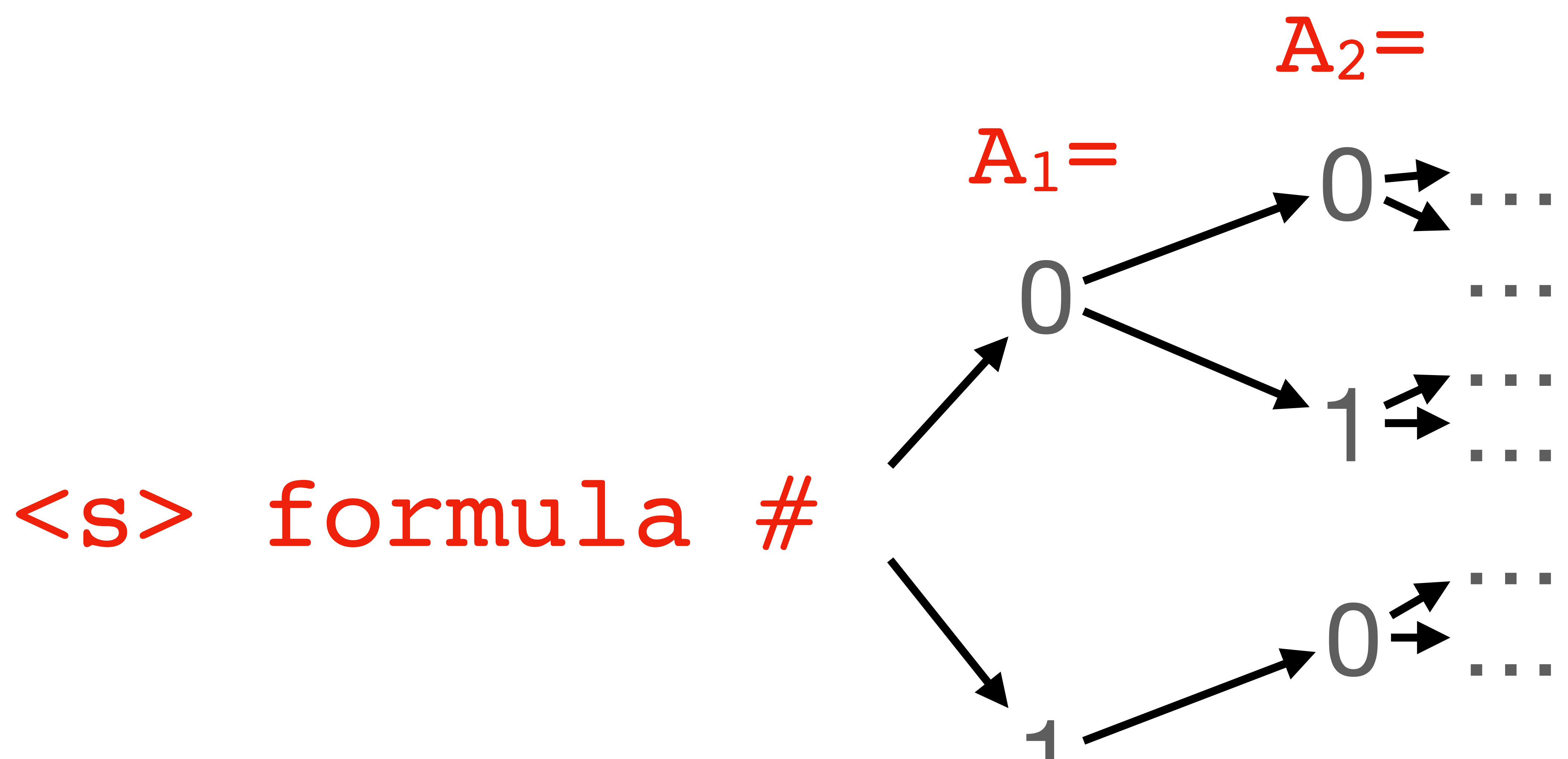
- Now let's look at autoregressive models.
- Can we implement goodness(**x**) using an autoregressive model?

x_t=	0	1
$p(x_t \mid \mathbf{x}_{<t})=$	?	?



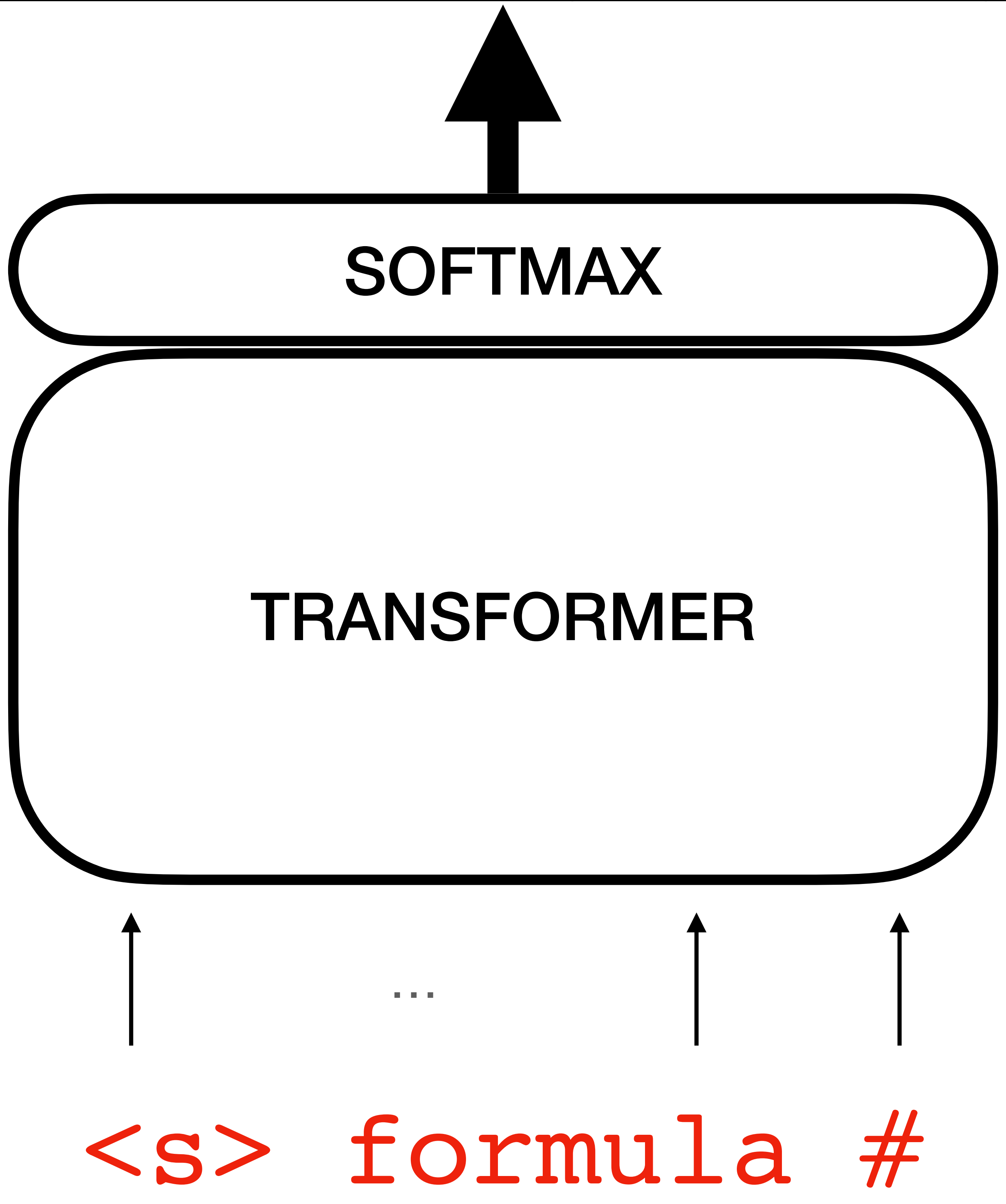
Why EBMs $\supsetneq$ autoregressive models?

- Computing the first token right after `<s> formula #` is as hard as determining if `formula` is satisfiable…



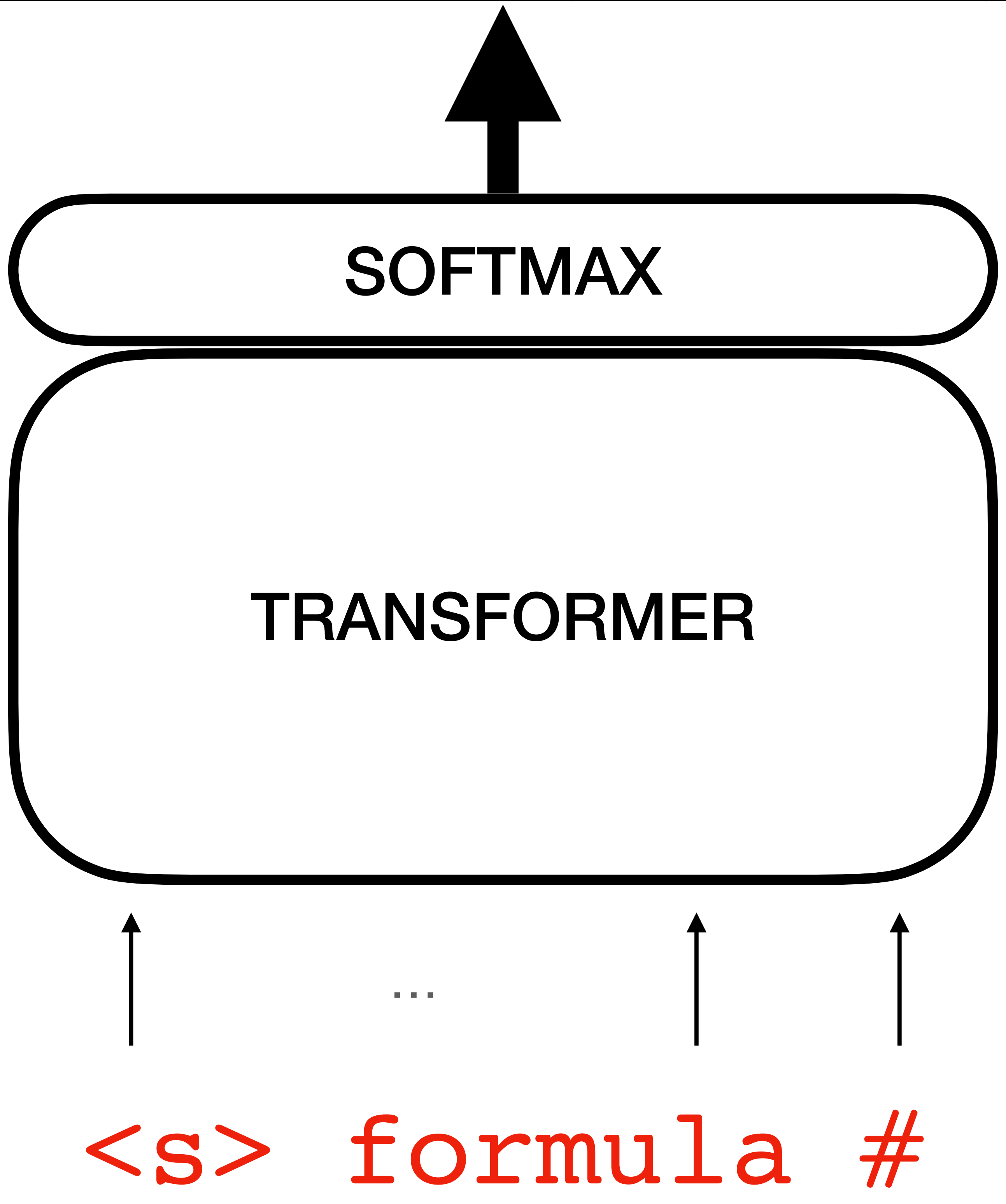
In contrast, EBMs need only to *check* a *single* assignment.

$x_t=$	0	1
$p(x_t \mid \mathbf{x}_{<t}) =$	?	?



Why EBMs $\supsetneq$ autoregressive models?

x_t=	0	1
$p(x_t \mid \mathbf{x}_{<t})$ =	?	?



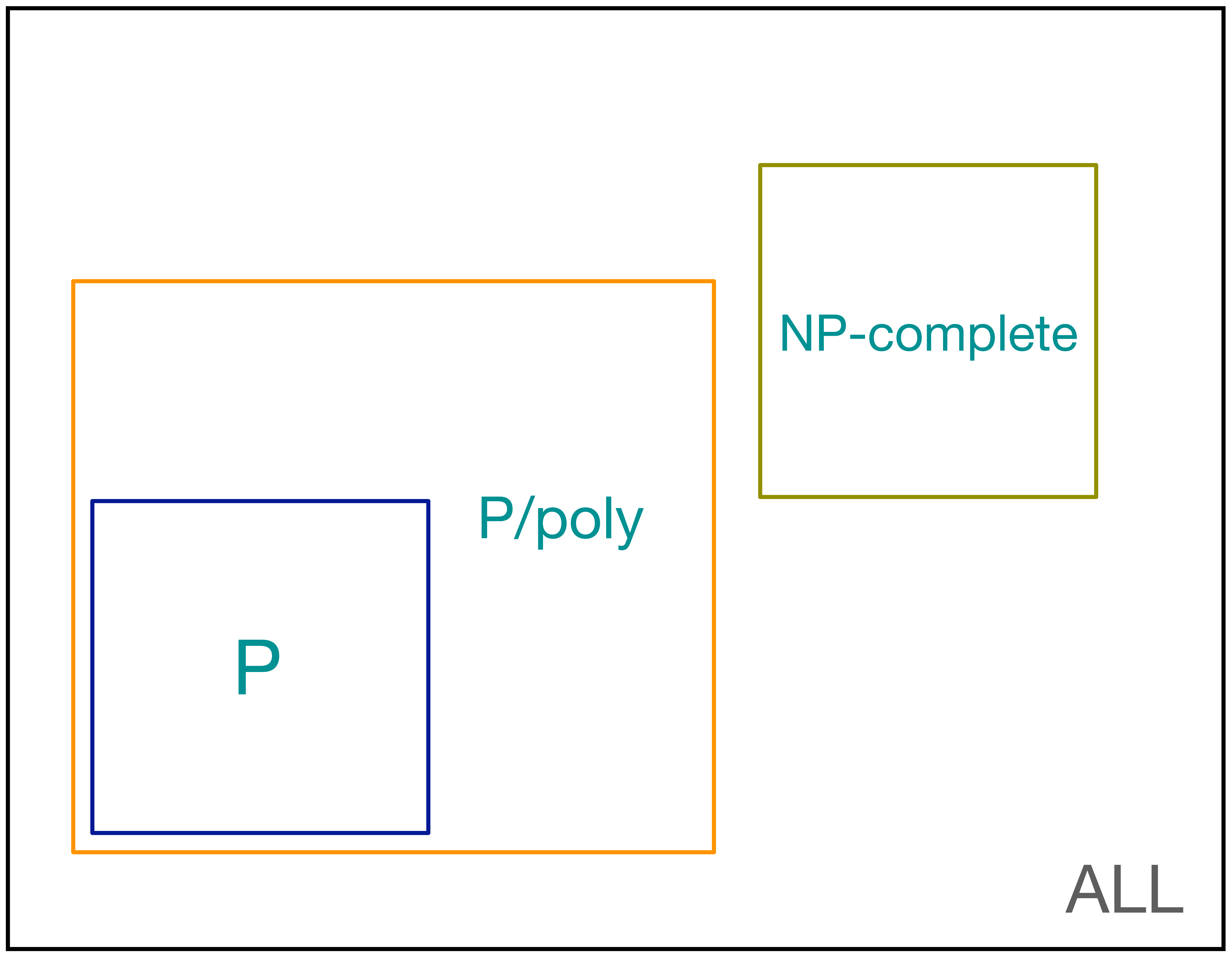
- Computing the first token right after `<s> formula #` is as hard as determining if `formula` is satisfiable…
  - which is NP-complete!
- Thus, **no** polynomial-time autoregressive model can model such distributions if P≠NP.

Actually it got worse

- There are distributions that can be captured by EBMs.
  But no autoregressive model can:
  - capture those distributions exactly (Theorem 1)
  - approximate well enough to get the same ranking of strings (Theorem 2)
  - approximate within any multiplicative factor (Theorem 4)
- Why should we care if we only need to model finite datasets?
  - in other words, we can always make the model larger to handle longer sequences (if smaller models don't work)…right?

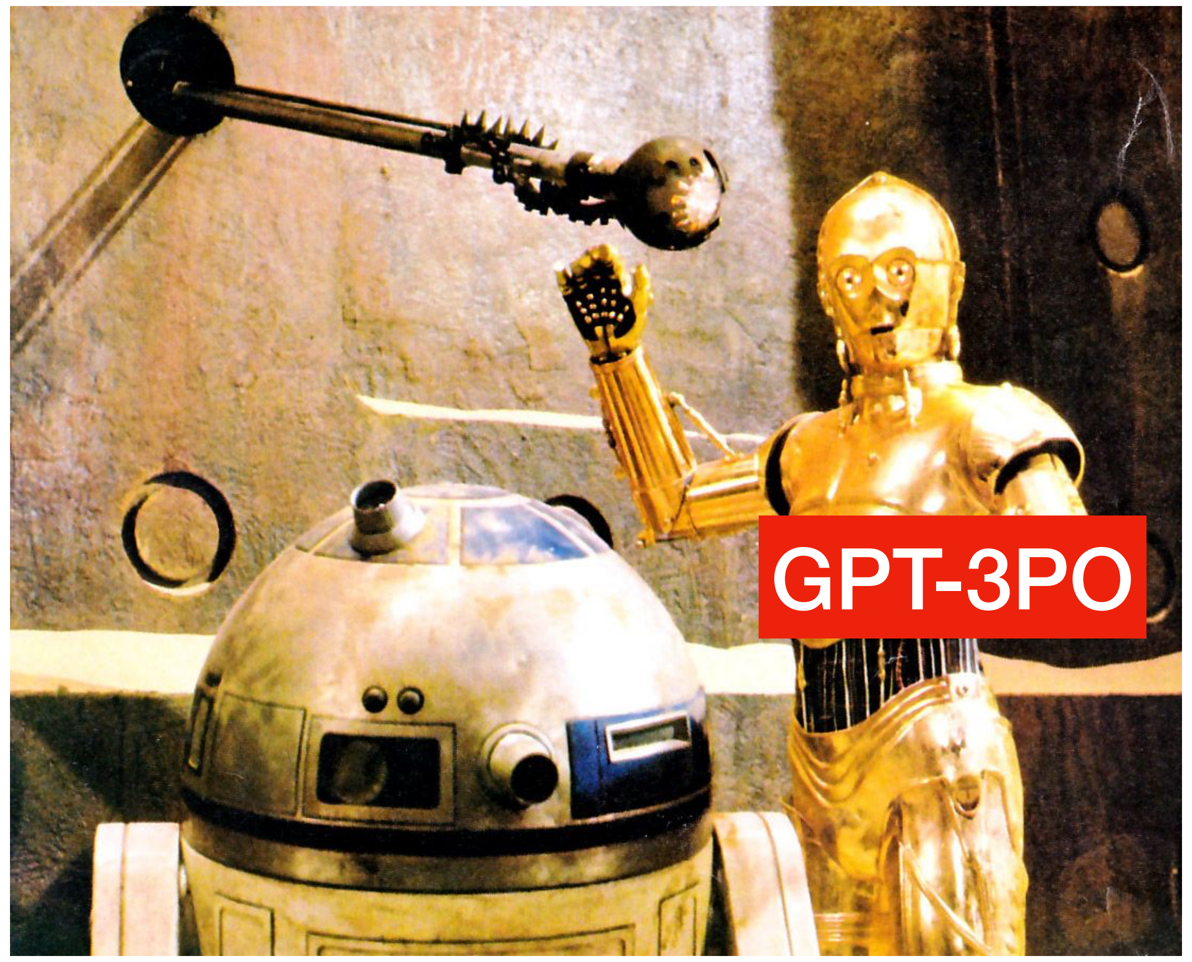
Just make the model larger?

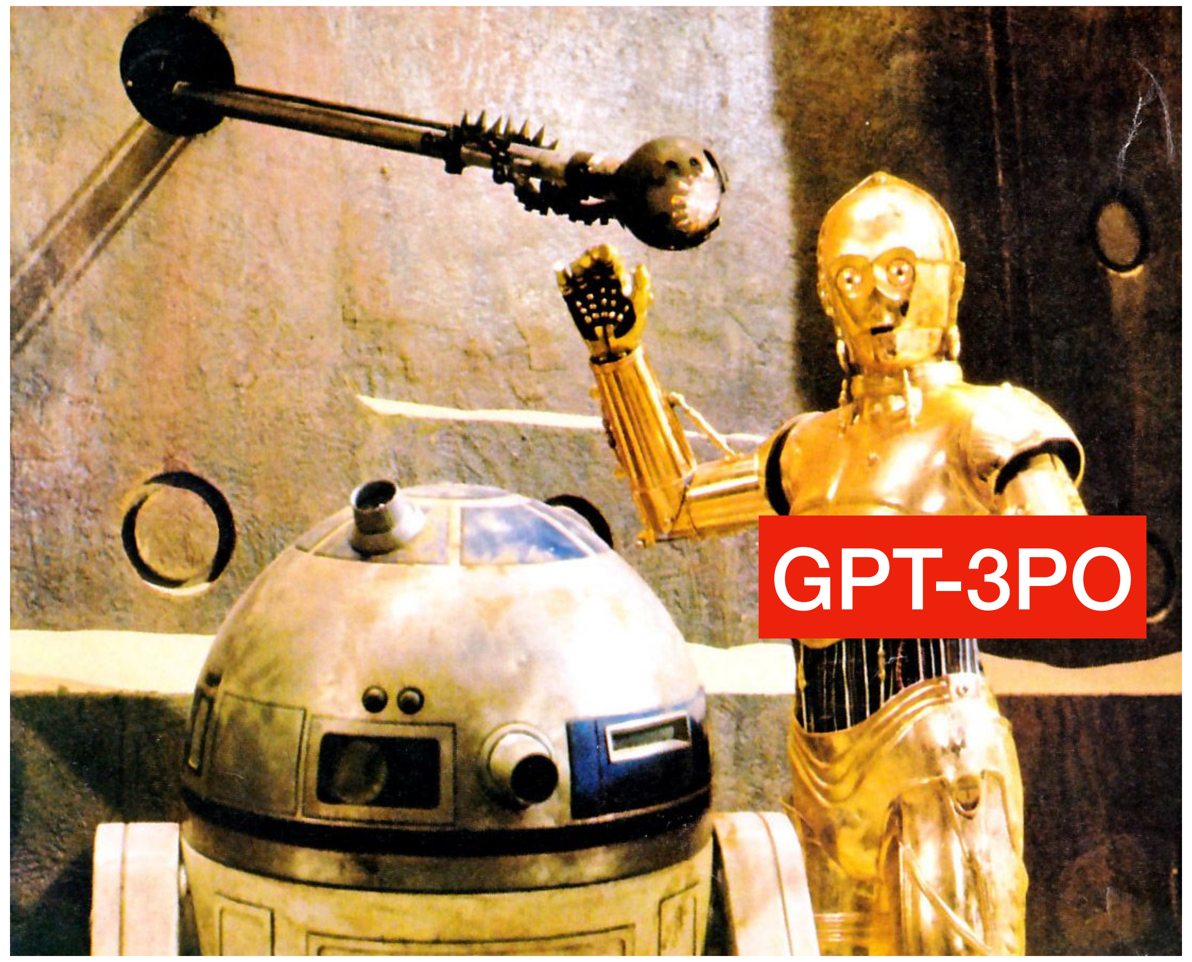
- If the model sizes *only* grow polynomially in sequence length, they belong in the P/poly class.
- It is widely believed NP$\not\subseteq$P/poly.
- So the models must **grow superpolynomially larger** and/or **run superpolynomially longer** in sequence length, to model longer problems (since they are NP-hard).
    - otherwise, the model simply won't fit even with access to an oracle in training!



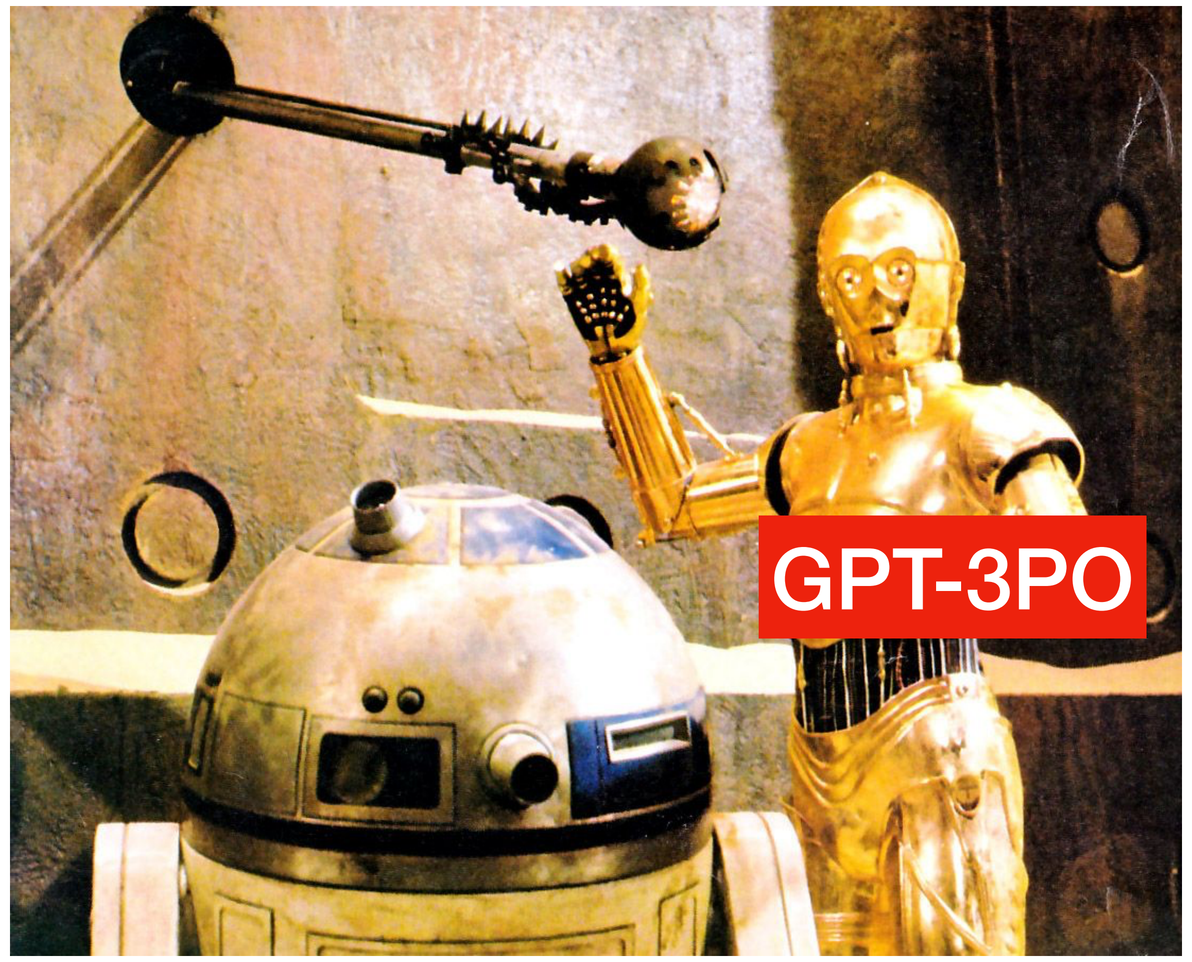
What if I am not interested in Boolean SAT problems?

- In general, autoregressive models cannot capture distributions over strings of the form `problem#solution`, where a `problem` is computationally hard to solve.
  - EBMs can capture such distributions
- Some CL/NLP `problems` are indeed computationally hard:
  - Parsing of many syntactic/semantic formalisms (e.g. AMR)
  - Propositional logic (NLI)
  - Optimality Theory
- Important linguistic regularities cannot be captured by autoregressive models!
- We use ~~propositional logic~~ generating a Star Wars movie script as an example.


GPT-3PO


GPT-3PO

A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....

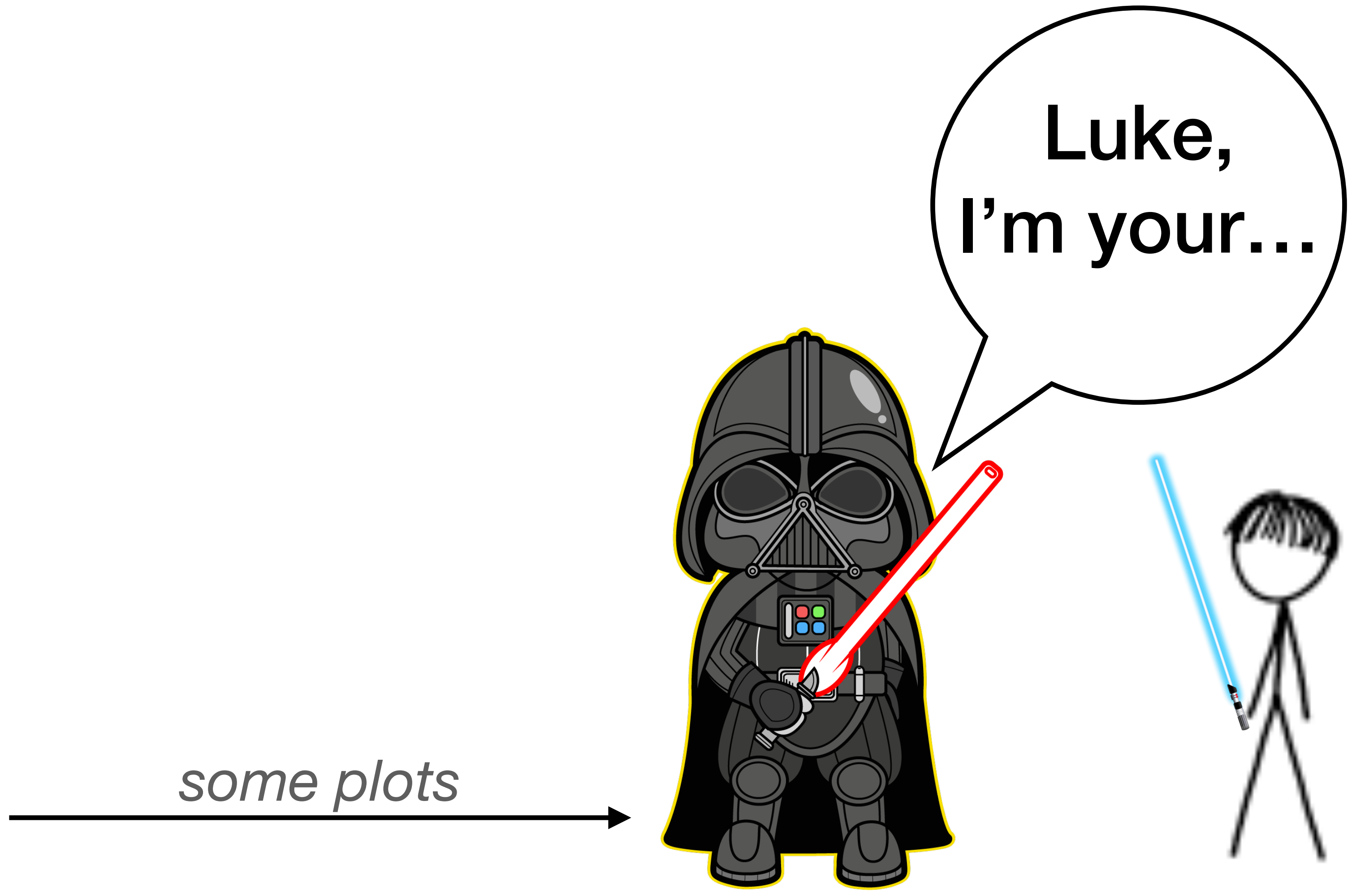


A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….

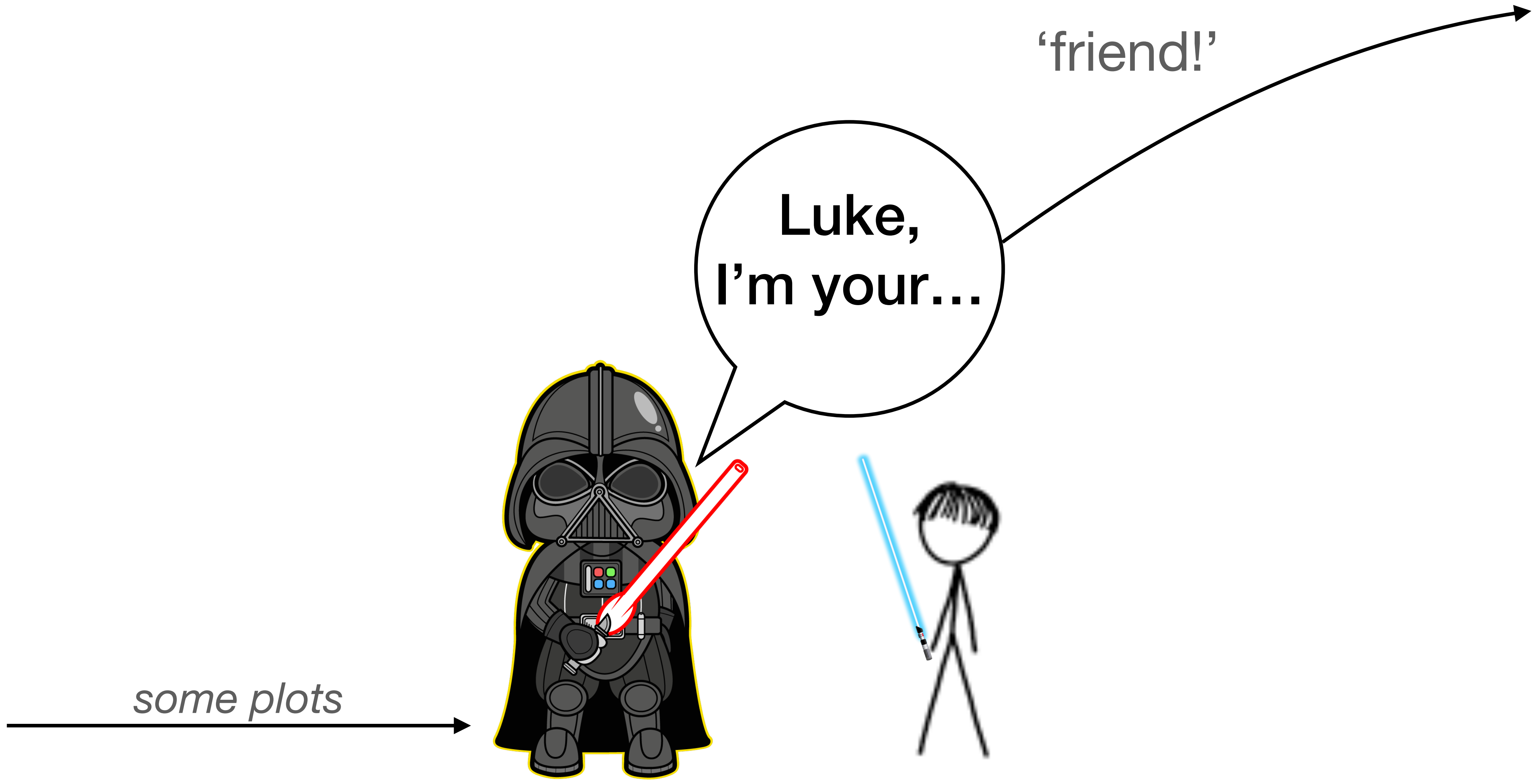
A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....

some plots

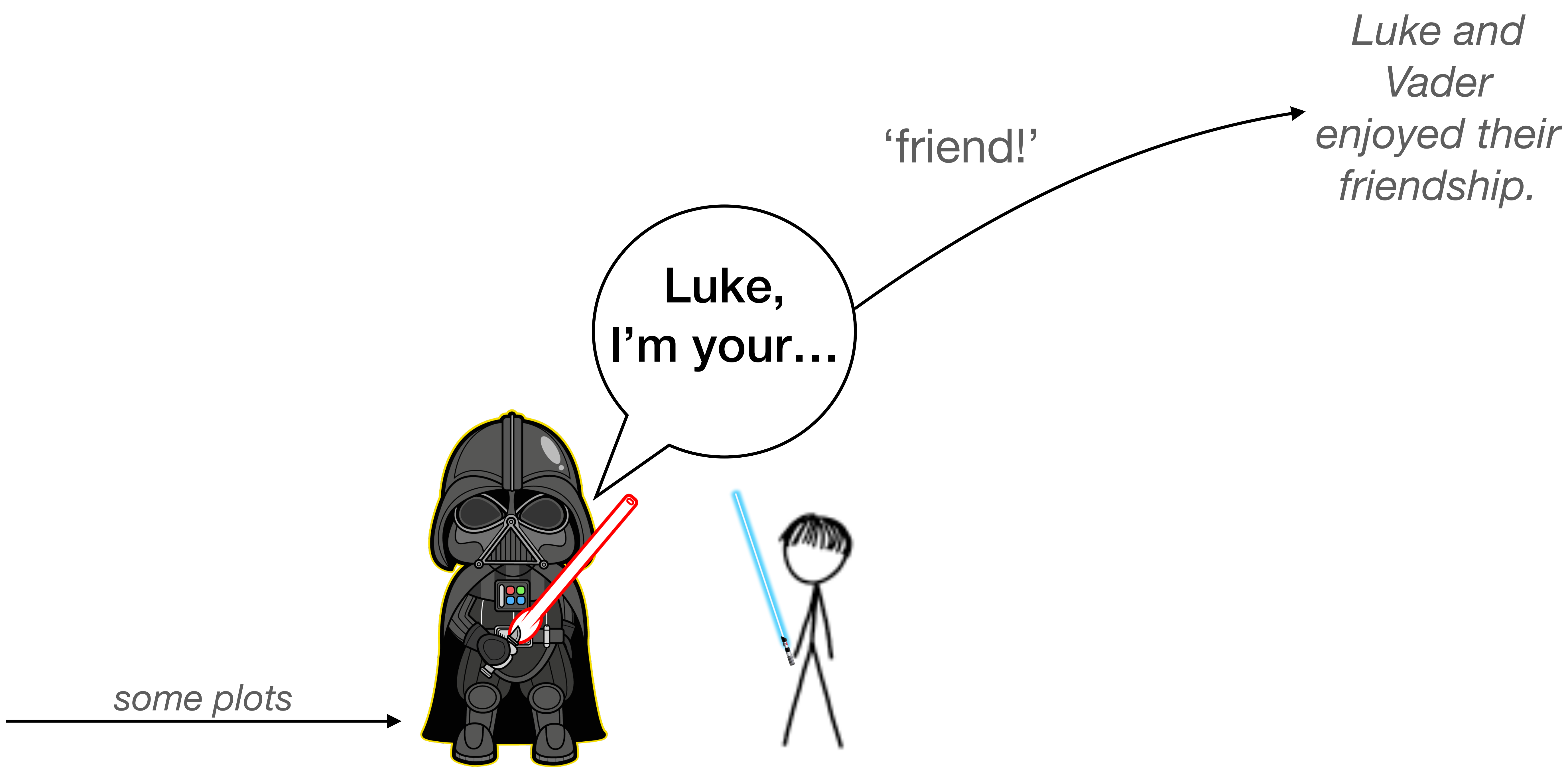
A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire, and eventually won.
The story started with Luke….



A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….



A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....



A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….

some plots

‘friend!’

Luke and Vader enjoyed their friendship.

A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire, and eventually won.
The story started with Luke….

some plots

'friend!'

Luke and Vader enjoyed their friendship.

Vader turned Luke to the Dark Side.

A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire, and eventually won.
The story started with Luke….

some plots

‘friend!’

Luke and Vader enjoyed their friendship.

Vader turned Luke to the Dark Side.

A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….

some plots

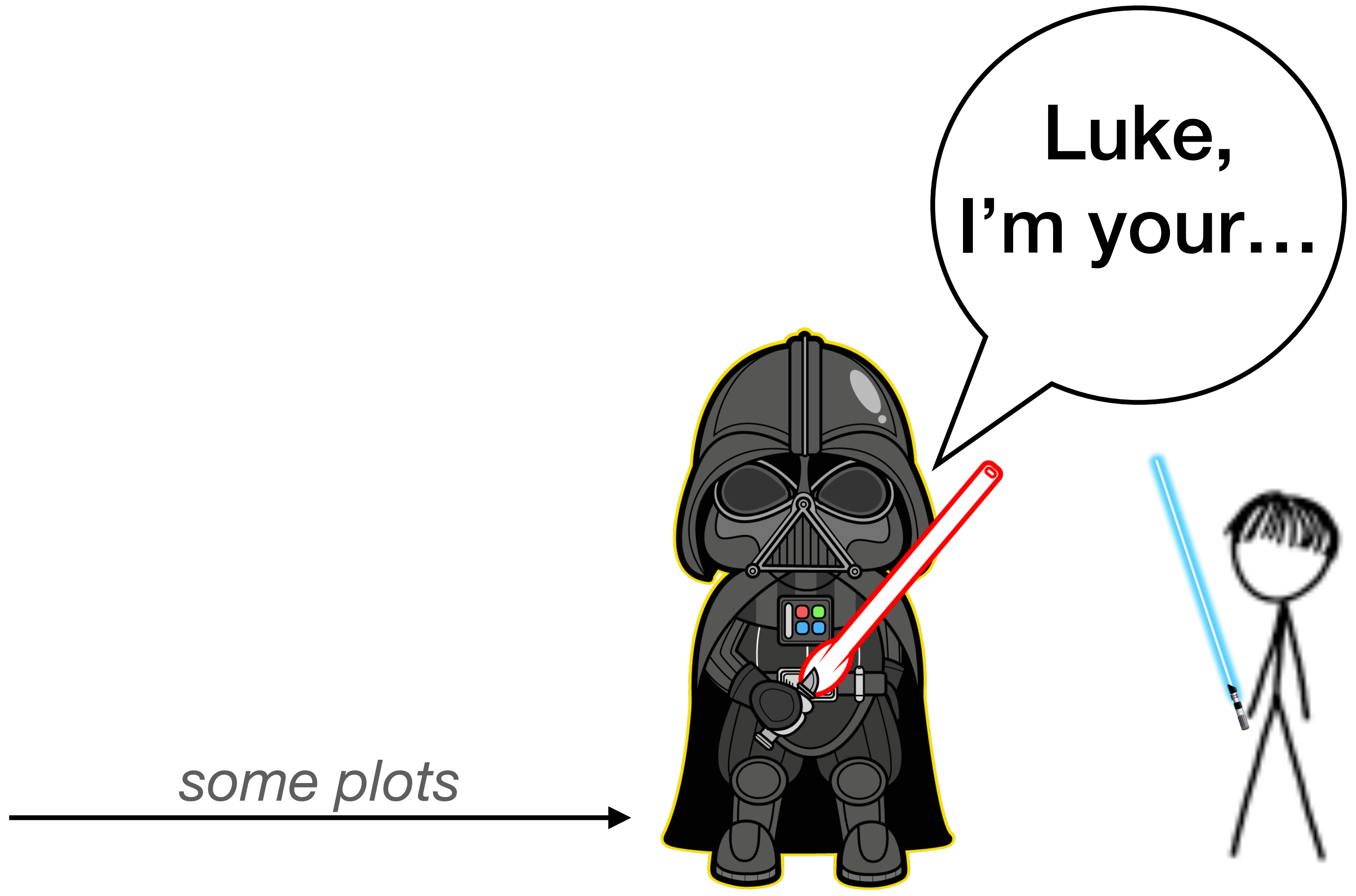
'friend!'

Luke and Vader enjoyed their friendship.

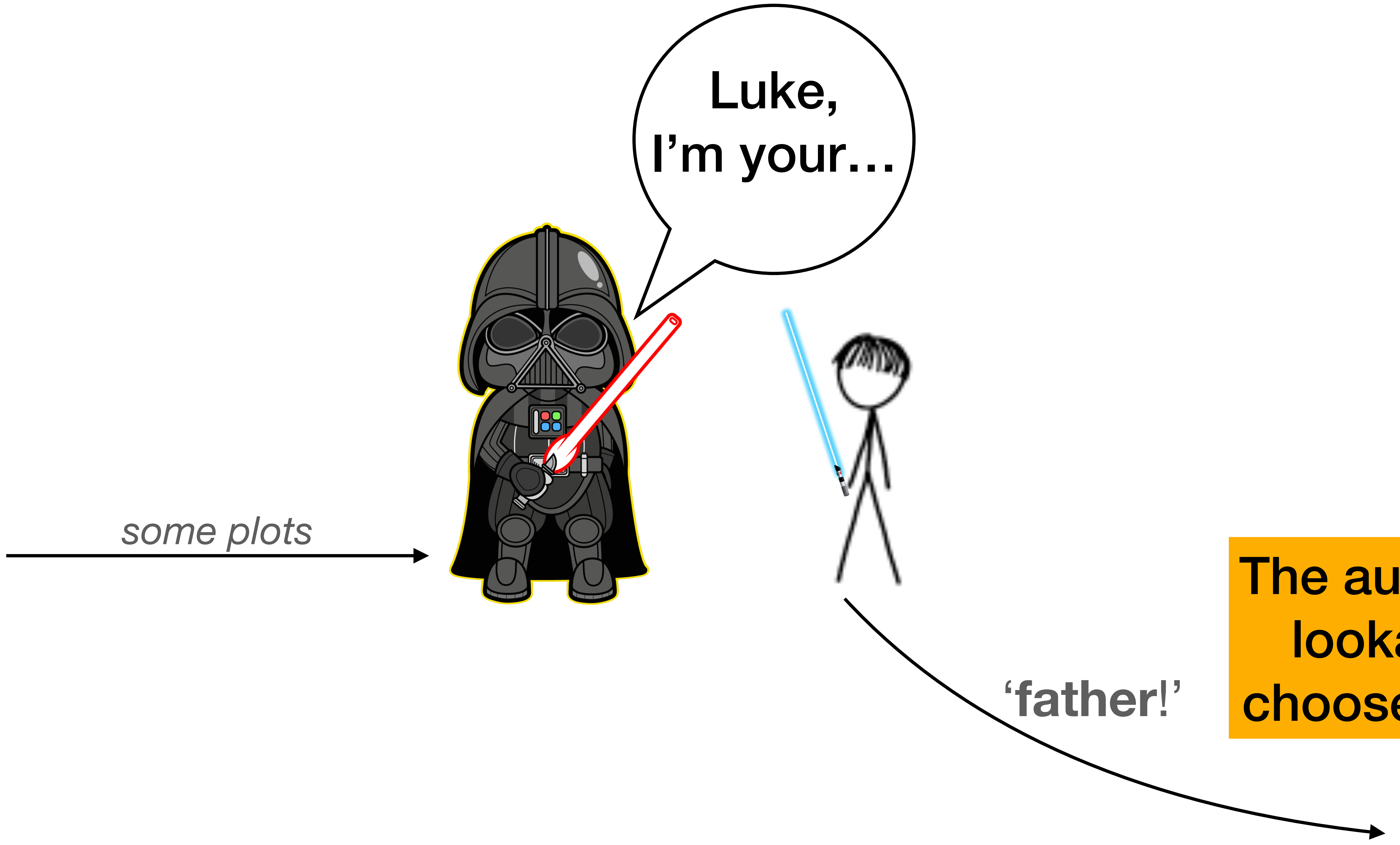
Vader turned Luke to the Dark Side.

The Rebels lost!
(violates what we said at the beginning)

A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....



A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire, and eventually won.
The story started with Luke….



The authors used lookahead to choose this word

A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….

some plots

‘**father**!’

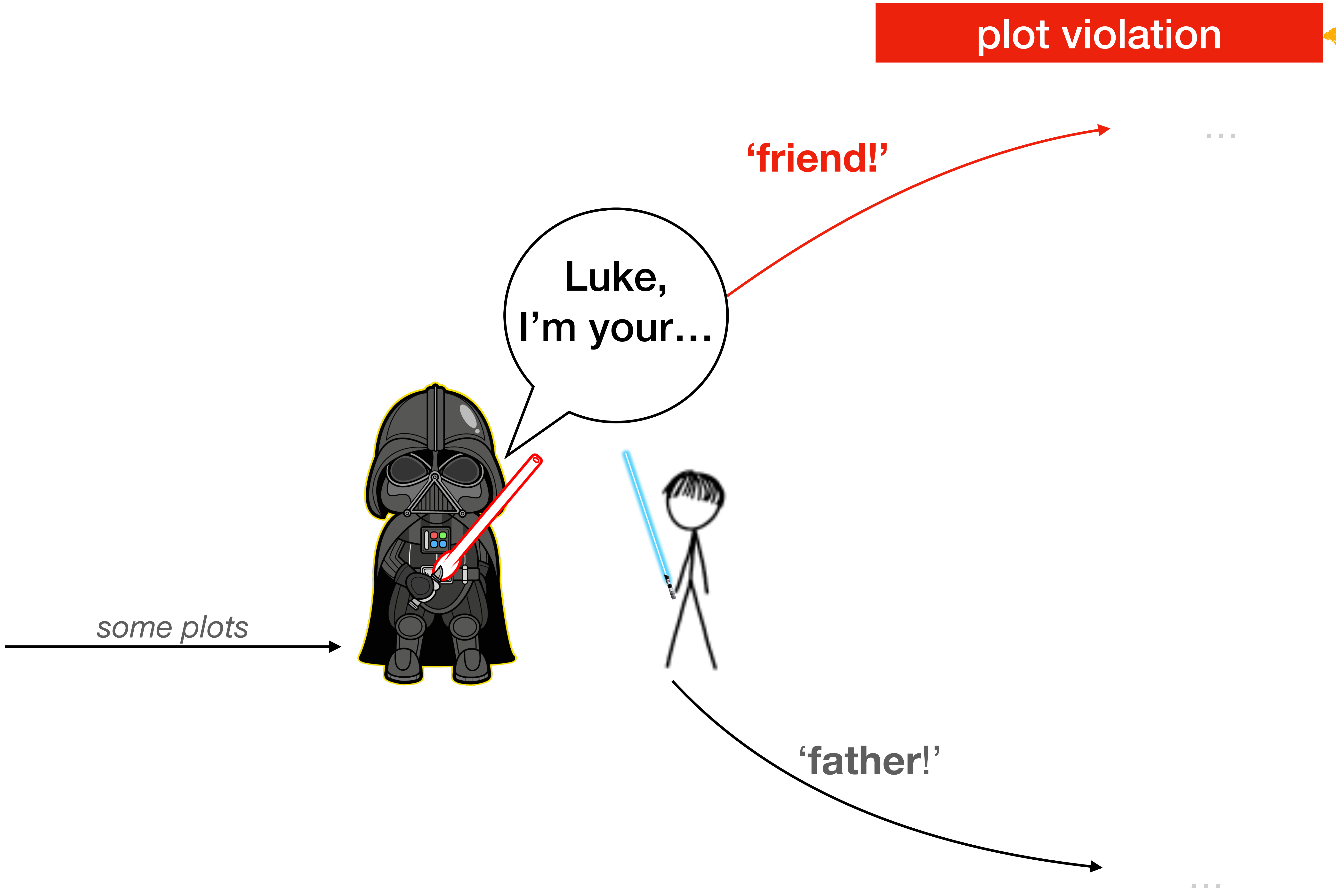
The authors used lookahead to choose this word

noooooooo

The Rebels won!

A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….

can we tell a continuation is
may be bad early?



A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….

can the autoregressive model learn that these conditional probabilities are small?

p('friend' | …Luke, I'm your)≃0

'friend!'

…

some plots

'father!'

…

A long time ago in a galaxy far, far away….
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke….

can the autoregressive model learn that these conditional probabilities are small?

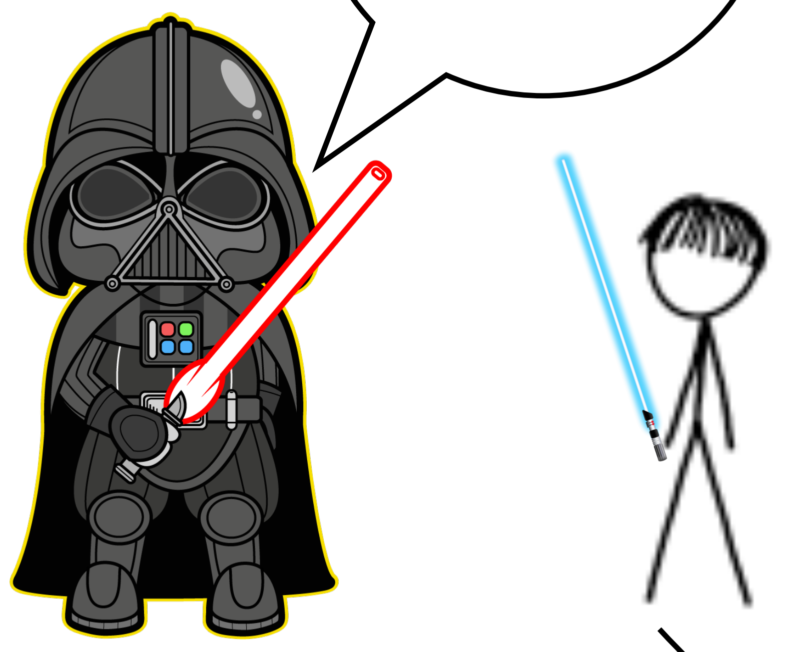
turns out we can't if P≠NP (Theorem 4)

p('friend' | …Luke, I'm your)≃0

'friend!'

…

Luke,
I'm your…



some plots

'father!'

…

A long time ago in a galaxy far, far away....
The Rebels fought against the evil Galactic Empire,
and eventually won.
The story started with Luke....

brief summary so far

- Autoregressive models cannot even guarantee that its generation is consistent (under propositional logic)!
  - This is really bad because checking their (in)consistency is indeed easy.
- Speaking very loosely, autoregressive models cannot tell between a surprising plot twist and an inconsistent continuation.

Outline

- Autoregressive models are not as expressive as other model families, energy-based models in particular.
    - And having more parameters helps little!
- **Model families that are more expressive than autoregressive models made their own trade-offs.**

Comparison of model families

	Compact parameters?	Efficient scoring?	Efficient sampling?	Support can be...
Autoregressive models	✔	✔	✔	Some but not all languages in P

Comparison of model families

	Compact parameters?	Efficient scoring?	Efficient sampling?	Support can be…
Autoregressive models	✔	✔	✔	Some but not all languages in P
Energy-based models				

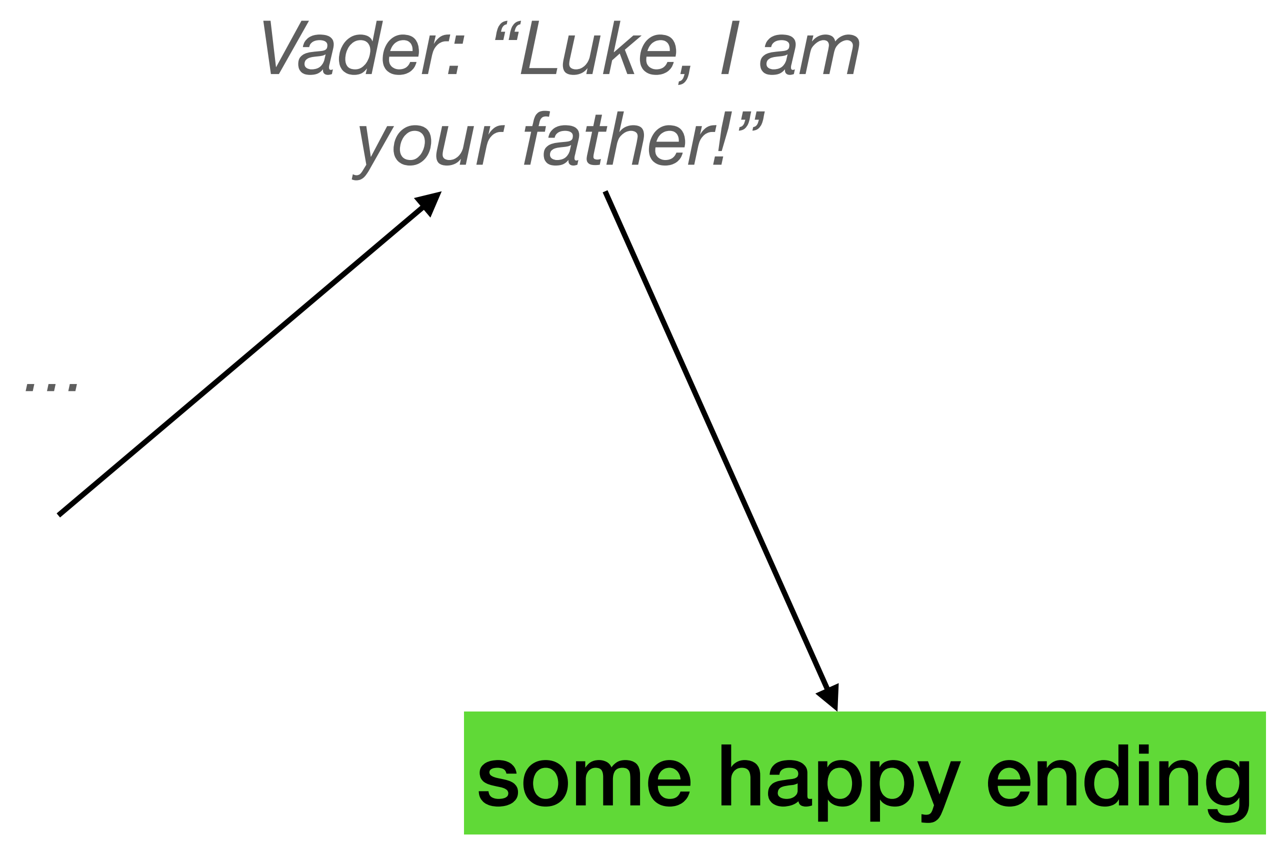
Comparison of model families

	Compact parameters?	Efficient scoring?	Efficient sampling?	Support can be…
Autoregressive models	✔	✔	✔	Some but not all languages in P
Energy-based models	✔	✔	✘ no efficient factorization	All languages in P

Comparison of model families

	Compact parameters?	Efficient scoring?	Efficient sampling?	Support can be…
Autoregressive models	✔	✔	✔	Some but not all languages in P
Energy-based models	✔	✔	✗	All languages in P
Autoregressive latent variable models				

An autoregressive model may not prefer plot twists
to other bland but logically inconsistent continuations…
But actually autoregressive models are capable of
generating such plot twists.
They just need some backstory
to help it justify the climax building.



Vader: "Luke, I am your father!"

…

some happy ending

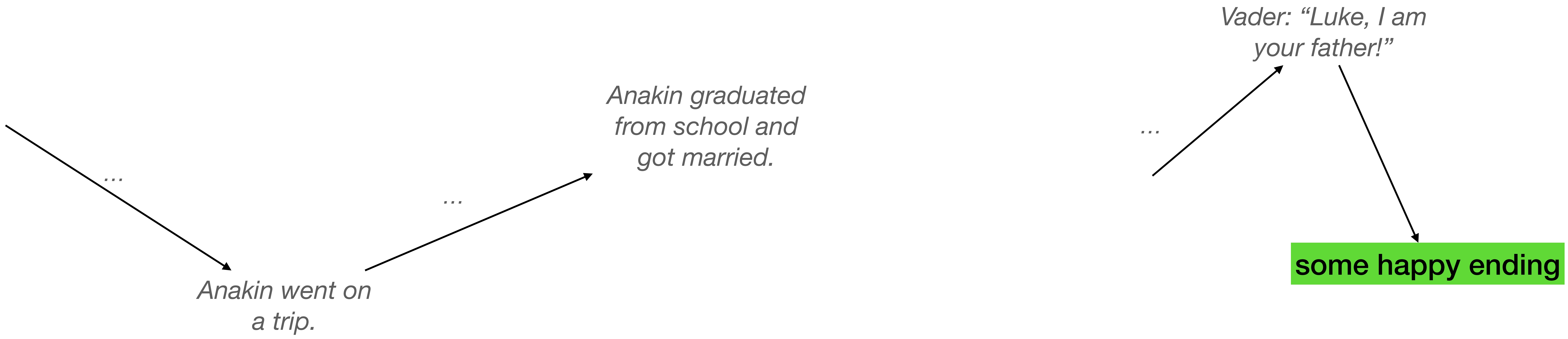
Vader: "Luke, I am your father!"

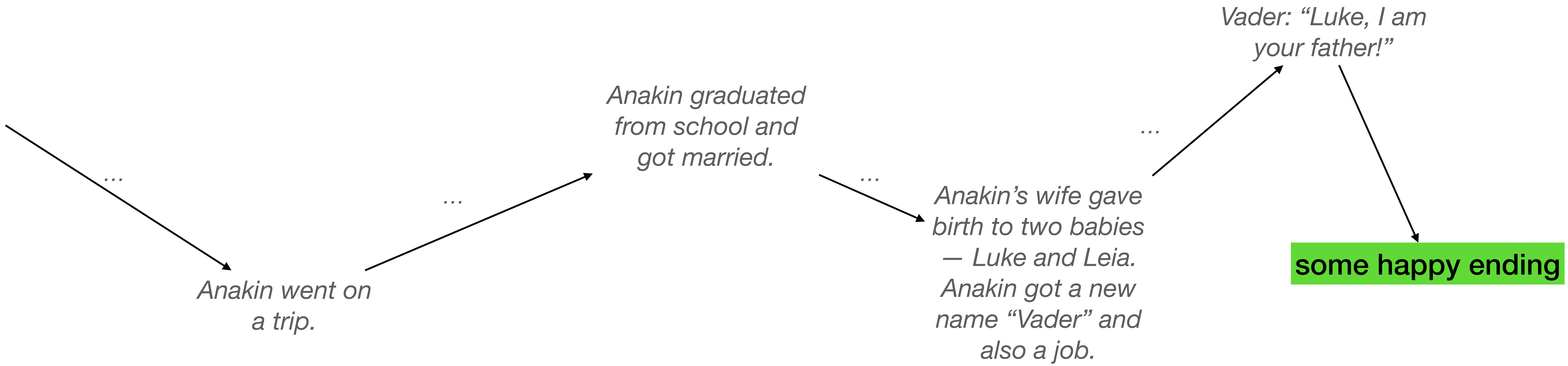
...

...

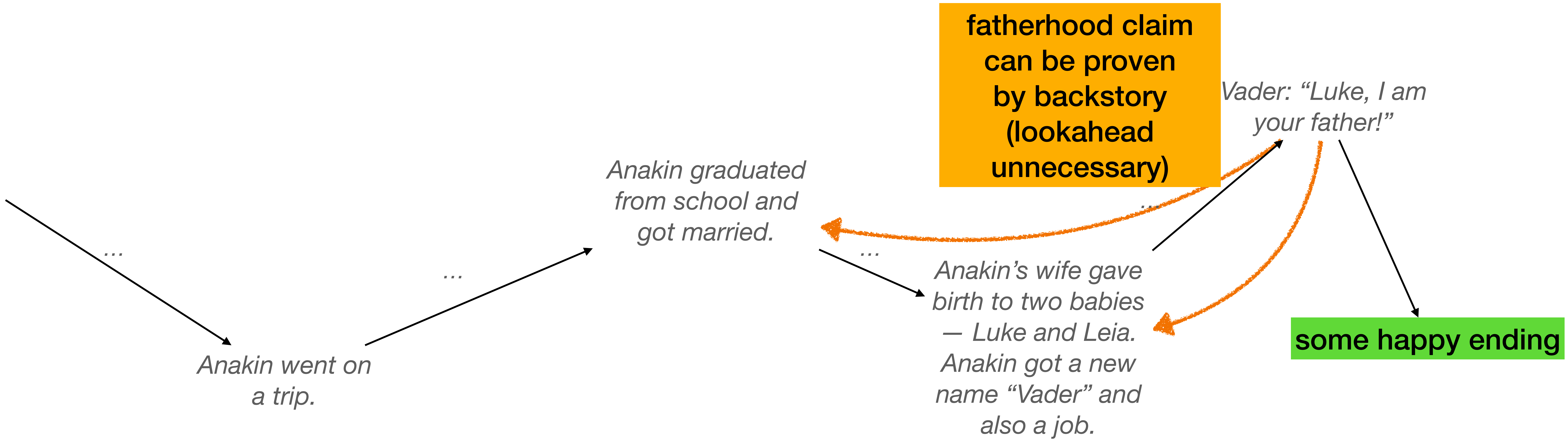
some happy ending

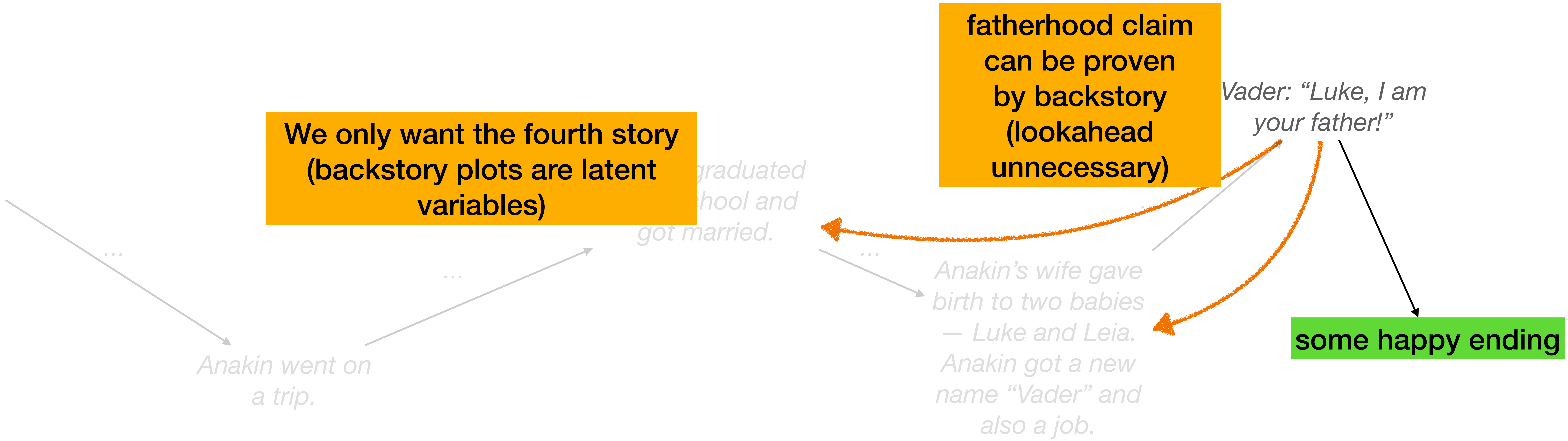
Anakin went on a trip.

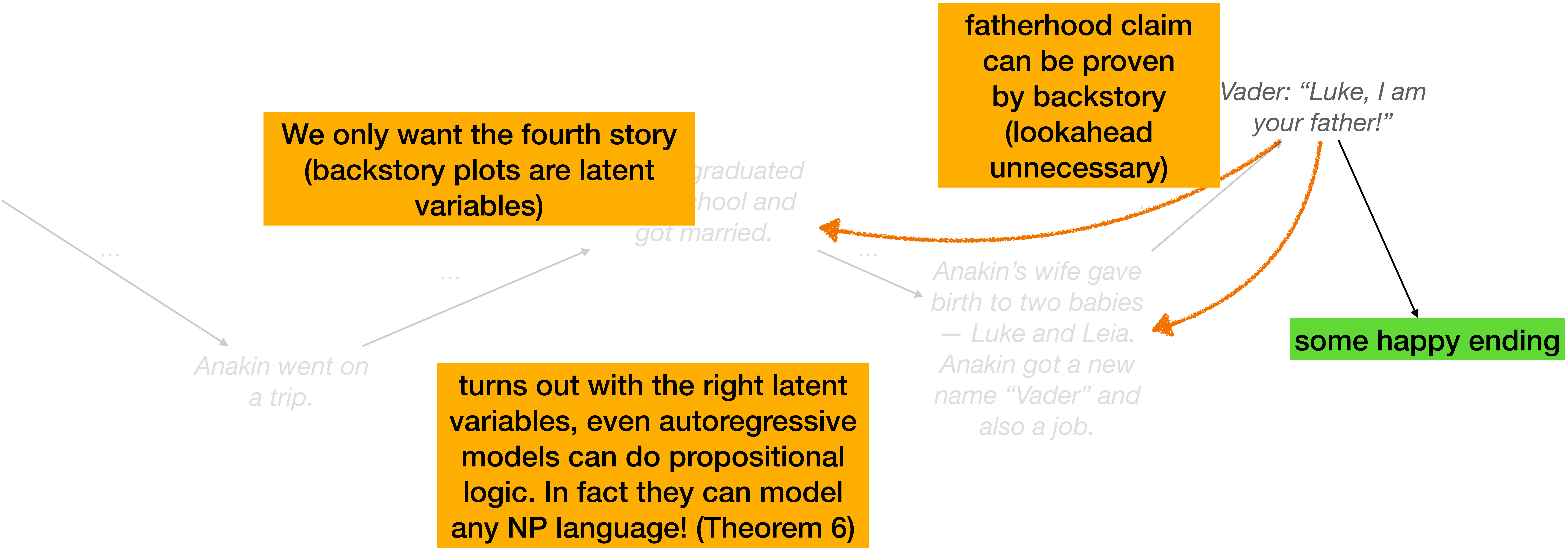

...
Anakin went on a trip.
...
Anakin graduated from school and got married.
...
Vader: “Luke, I am your father!”
some happy ending



...
Anakin went on a trip.
...
Anakin graduated from school and got married.
...
Anakin's wife gave birth to two babies — Luke and Leia. Anakin got a new name "Vader" and also a job.
...
Vader: "Luke, I am your father!"
some happy ending


fatherhood claim can be proven by backstory (lookahead unnecessary)
Vader: “Luke, I am your father!”
Anakin graduated from school and got married.
...
...
...
...
Anakin went on a trip.
Anakin’s wife gave birth to two babies — Luke and Leia. Anakin got a new name “Vader” and also a job.
some happy ending


We only want the fourth story (backstory plots are latent variables)
fatherhood claim can be proven by backstory (lookahead unnecessary)
Vader: "Luke, I am your father!"
some happy ending
...
...
Anakin went on a trip.
graduated
hool and
got married.
...
Anakin's wife gave birth to two babies — Luke and Leia. Anakin got a new name "Vader" and also a job.


We only want the fourth story (backstory plots are latent variables)
fatherhood claim can be proven by backstory (lookahead unnecessary)
Vader: “Luke, I am your father!”
…
…
graduated
hool and
got married.
…
Anakin went on a trip.
Anakin’s wife gave birth to two babies — Luke and Leia. Anakin got a new name “Vader” and also a job.
some happy ending
turns out with the right latent variables, even autoregressive models can do propositional logic. In fact they can model any NP language! (Theorem 6)

Comparison of model families

	Compact parameters?	Efficient scoring?	Efficient sampling?	Support can be…
Autoregressive models	✔	✔	✔	Some but not all languages in P
Energy-based models	✔	✔	✗	All languages in P
Autoregressive latent variable models	✔	✗ needs to marginalize	✔	All languages in NP

Comparison of model families

	Compact parameters?	Efficient scoring?	Efficient sampling?	Support can be…
Autoregressive models	✔	✔	✔	Some but not all languages in P
Energy-based models	✔	✔	✗	All languages in P
Autoregressive latent variable models	✔	✗	✔	All languages in NP
Lookup models				

Lookup models

- if the model size can be unbounded, we can model **any** finite language!
- Look up factoids in a database
  - there are sub-linear time retrieval methods.
- examples include kNNLM and adaptive semiparametric LMs.

Comparison of model families

	Compact parameters?	Efficient scoring?	Efficient sampling?	Support can be…
Autoregressive models	✔	✔	✔	Some but not all languages in P
Energy-based models	✔	✔	✘	All languages in P
Autoregressive latent variable models	✔	✘	✔	All languages in NP
Lookup models	✘ unbounded size	✔	✔	Anything

Conclusion

- Autoregressive models are inherently limited.
  - Some string distributions have ‘hard’ conditional probabilities, even though the joint (unnormalized) probabilities may be easy to evaluate.
- Alternative model families have their own tradeoffs.