Neuro-symbolic Representations for Commonsense Knowledge and Reasoning

Antoine Bosselut

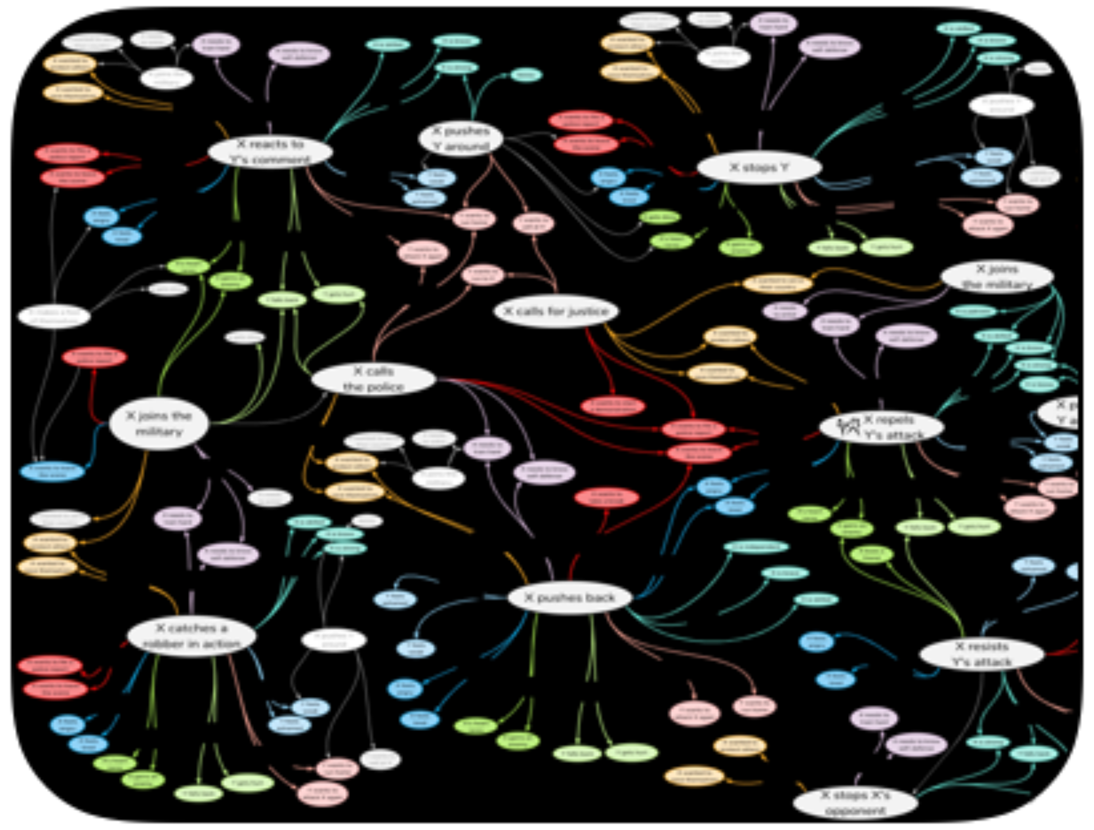
Paul G. Allen School of Computer Science & Engineering,
University of Washington

Filler Slide

Reasoning with Knowledge Graphs

Kai knew that things were getting out of control and managed to keep his temper in check

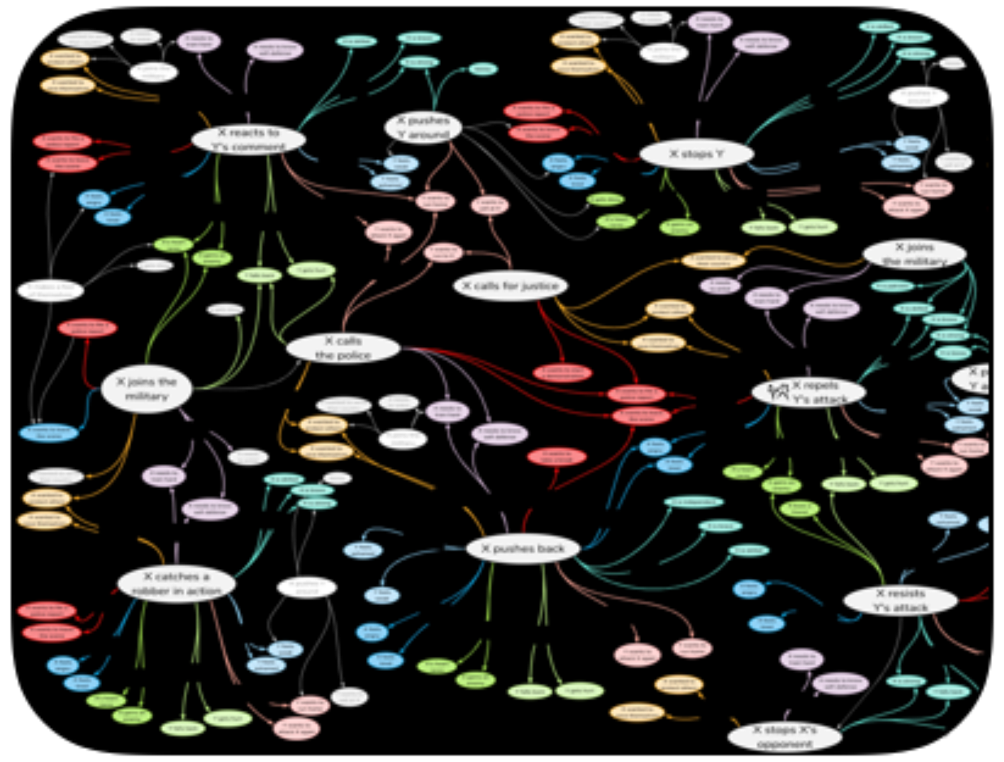
ATOMIC



Reasoning with Knowledge Graphs

- Situations rarely found **as-is** in commonsense knowledge graphs

ATOMIC



(Sap et al., 2019)

(X goes to the mall,
Effect on X, buys clothes)

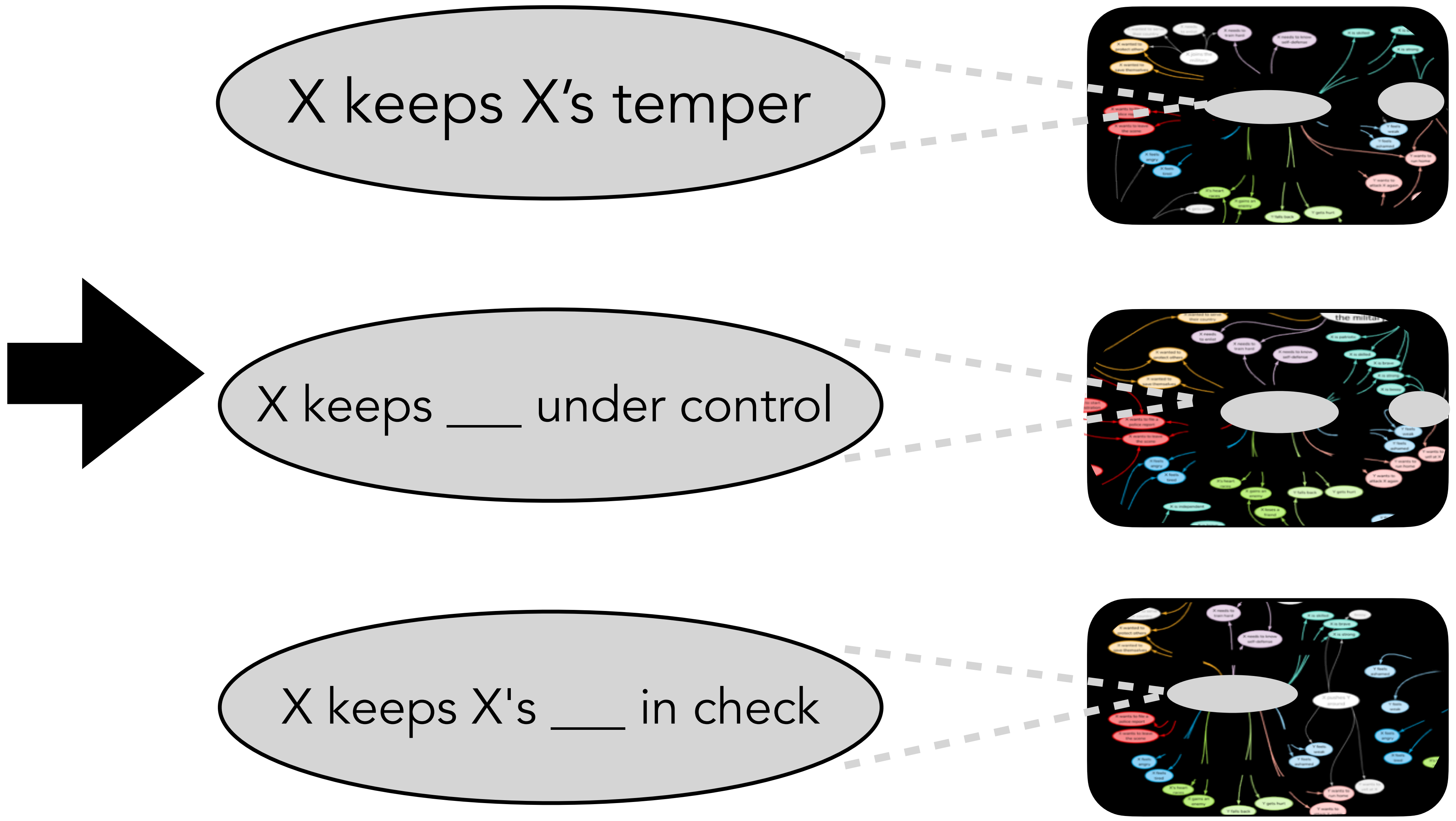
(X goes the mall,
Perception of X, rich)

(X gives Y some money,
Reaction of Y, grateful)

Reasoning with Knowledge Graphs

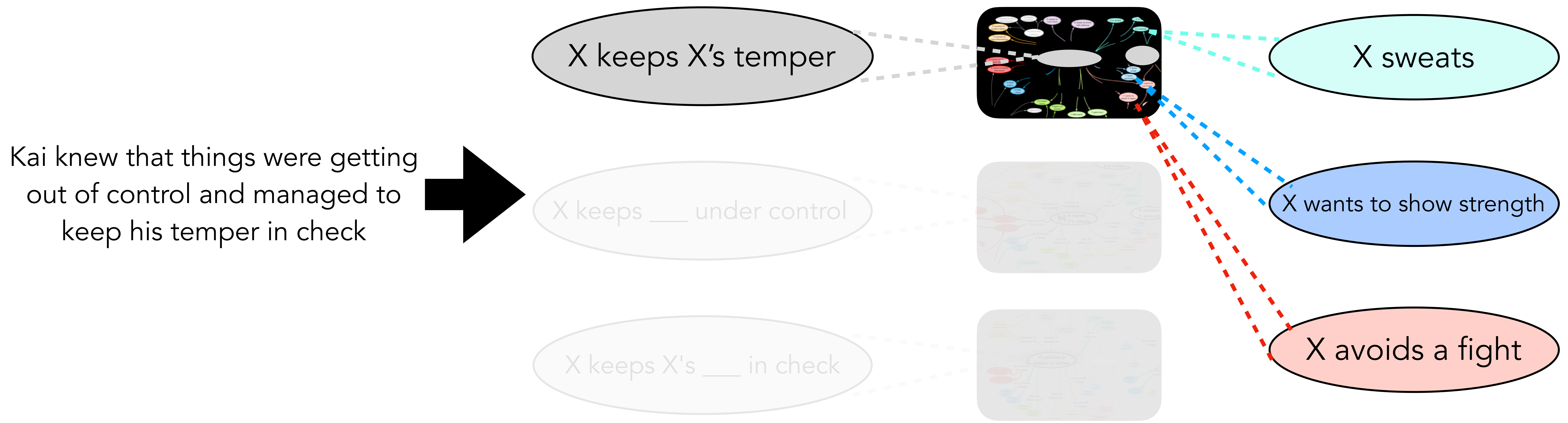
- Situations rarely found **as-is** in commonsense knowledge graphs
- Connecting to knowledge graphs can yield incorrect nodes

Kai knew that things were getting out of control and managed to keep his temper in check



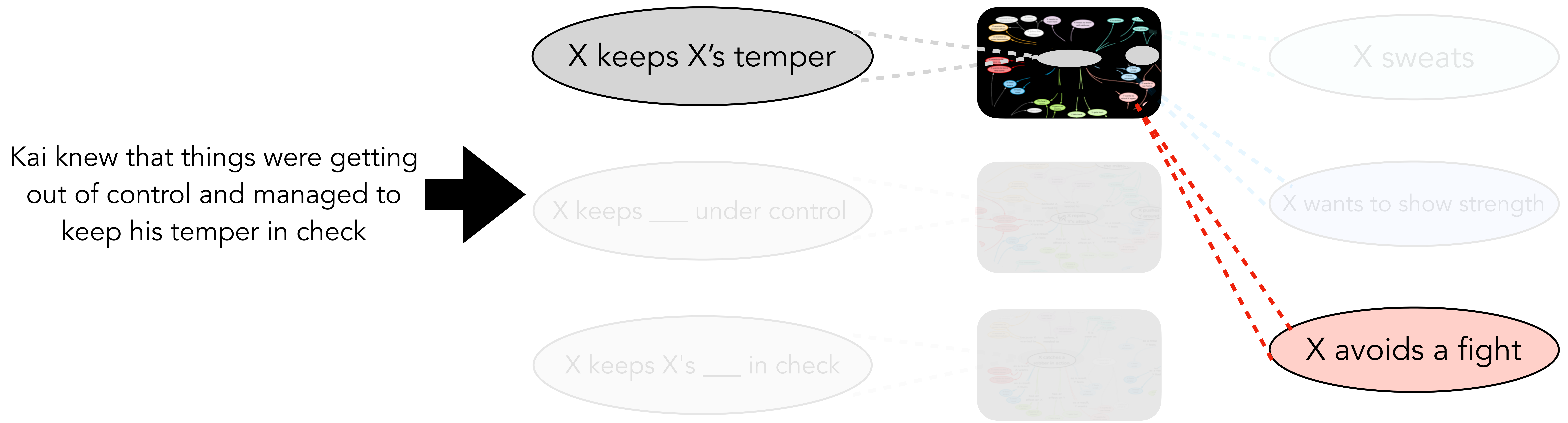
Reasoning with Knowledge Graphs

- Situations rarely found **as-is** in commonsense knowledge graphs
- Connecting to knowledge graphs can yield incorrect nodes



Reasoning with Knowledge Graphs

- Situations rarely found **as-is** in commonsense knowledge graphs
- Connecting to knowledge graphs can yield incorrect nodes

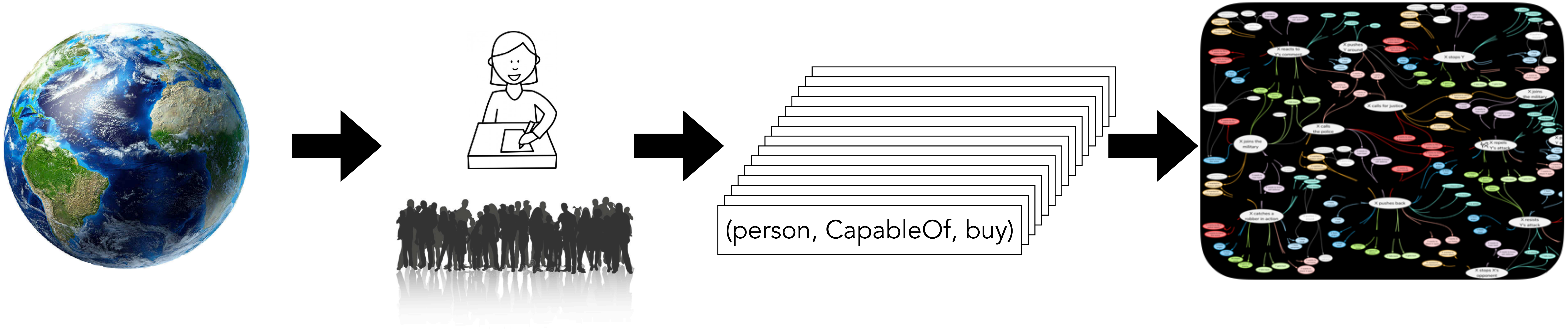


Constructing Symbolic Knowledge Graphs

Observe world

Write commonsense knowledge facts

Store facts in knowledge graph

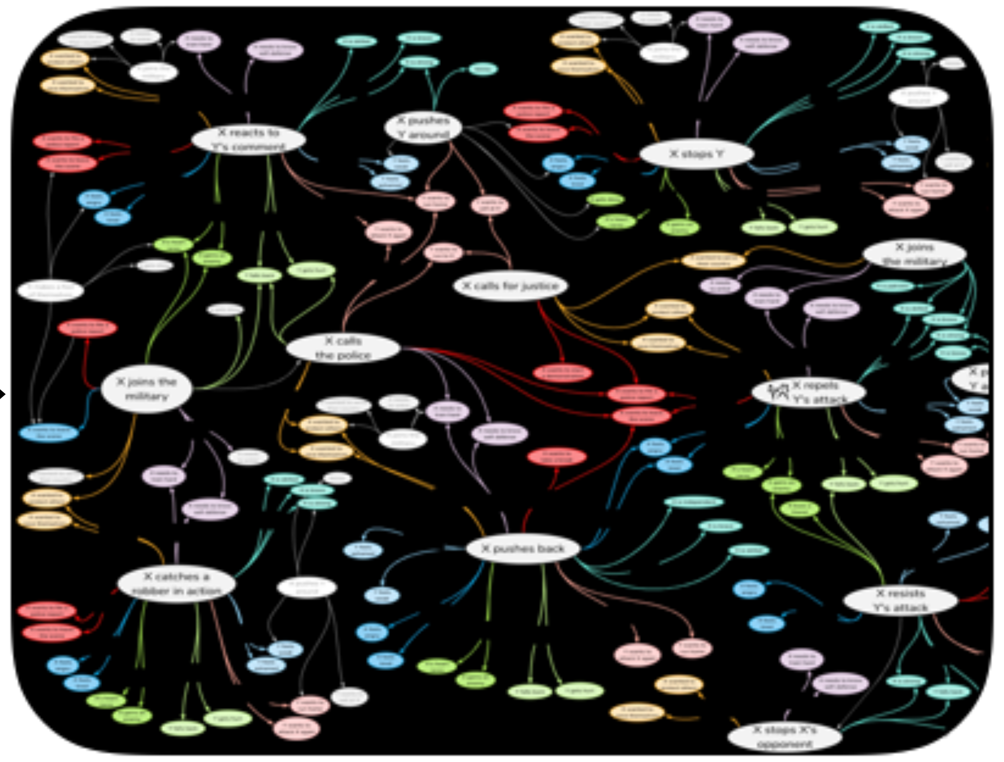


Constructing Symbolic Knowledge Graphs

Observe world

Write commonsense knowledge facts

Store facts in knowledge graph



(Miller, 1995)

(Singh et al., 2002)

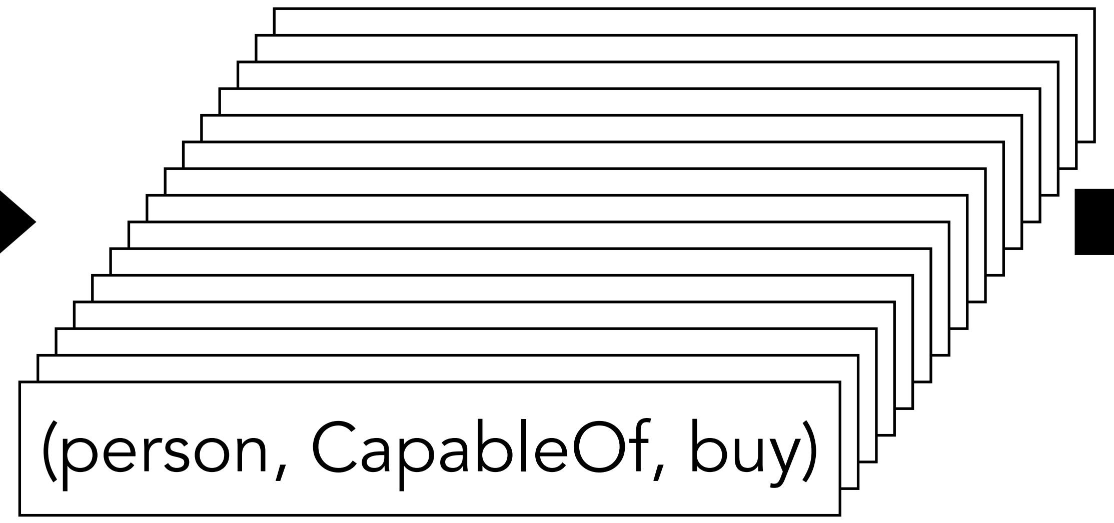
(Lenat, 1995)

(Sap et al., 2019)

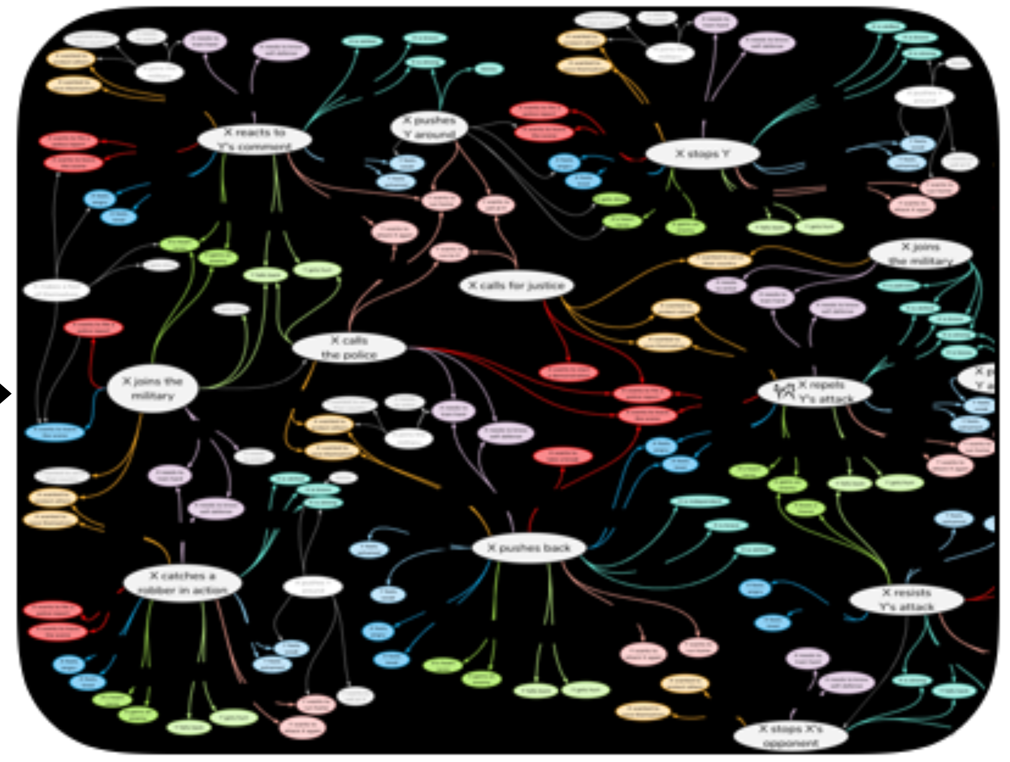
Constructing Symbolic Knowledge Graphs

Observe world

Write commonsense knowledge facts



Store facts in knowledge graph



(Miller, 1995)

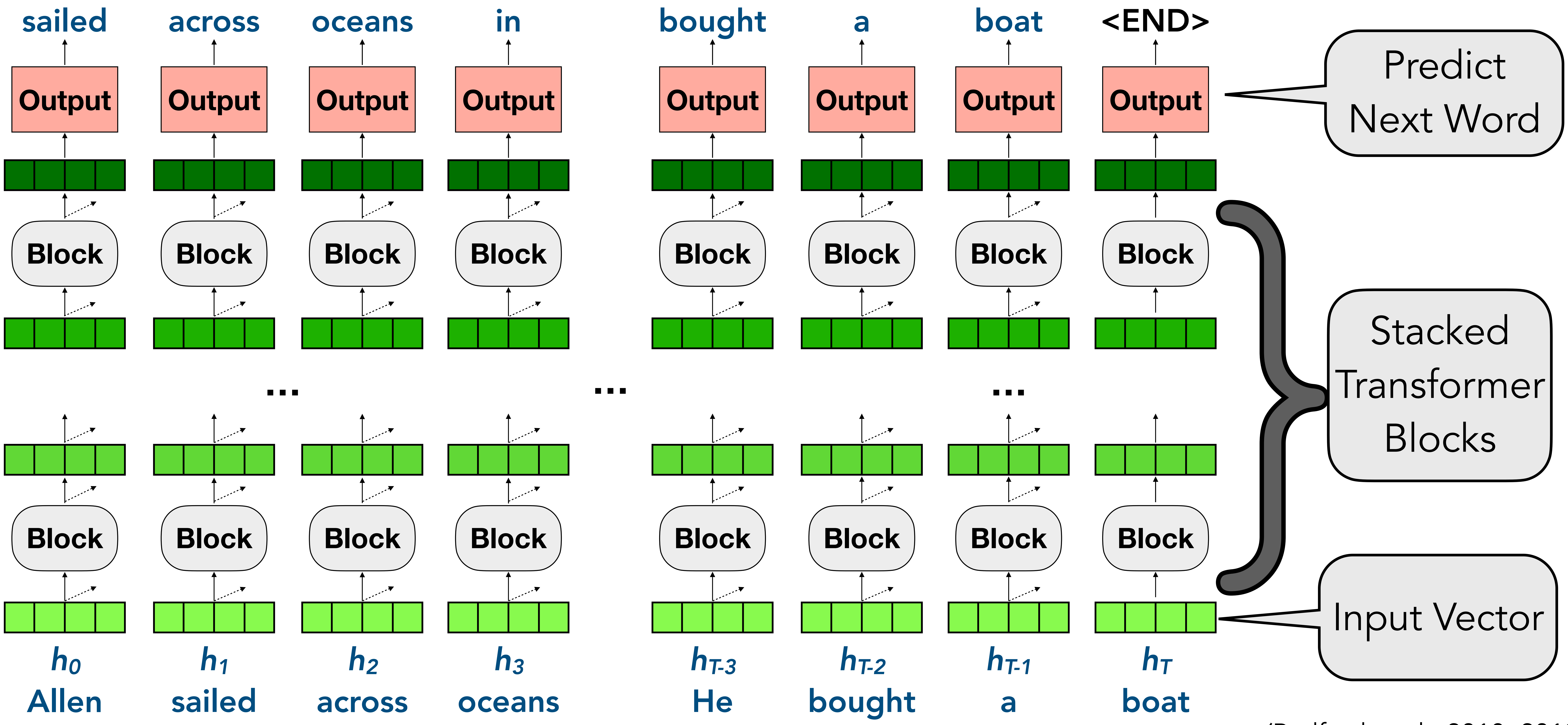
(Singh et al., 2002)

(Lenat, 1995)

(Sap et al., 2019)

Filler Slide

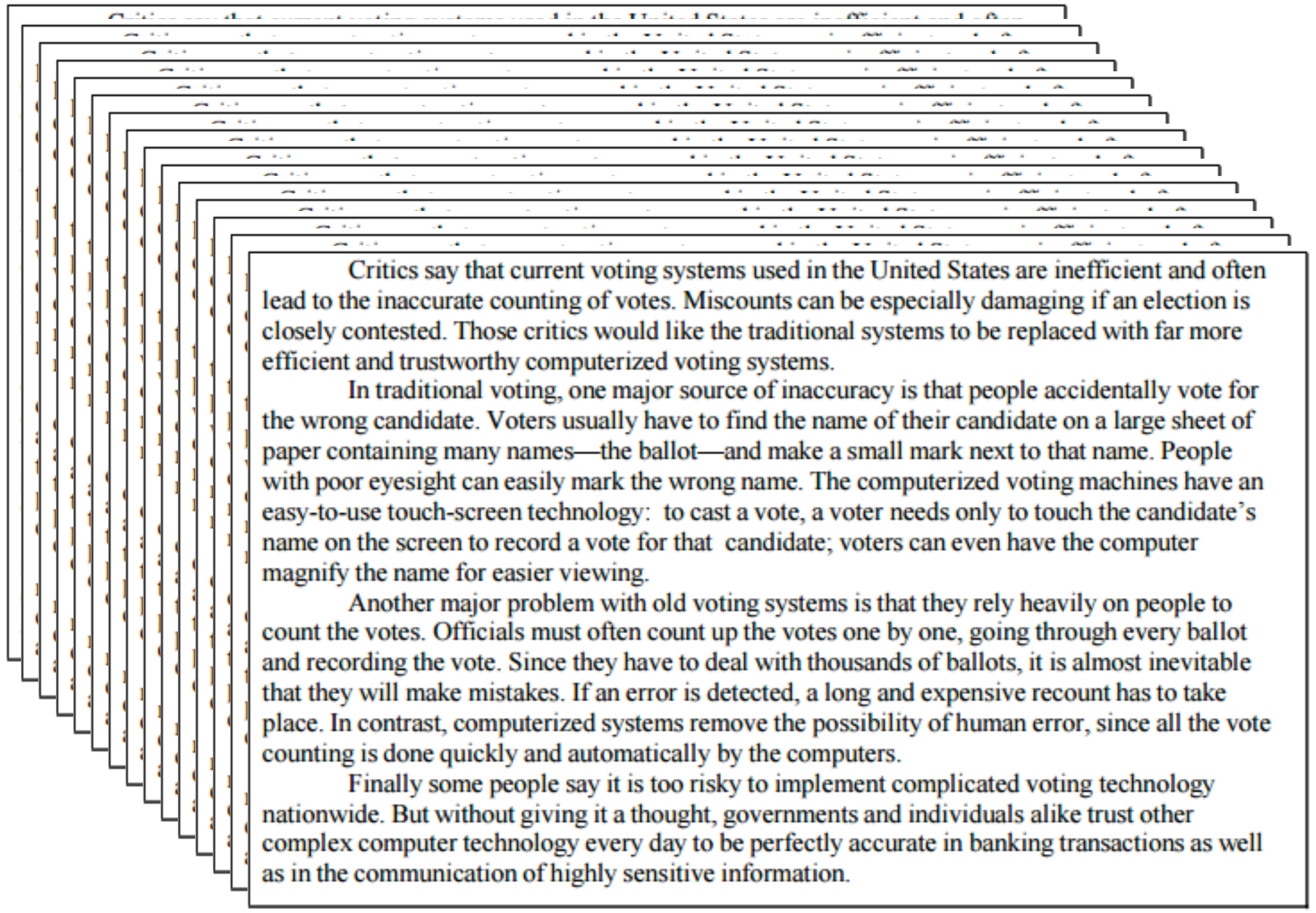
Transformer Language Models



(Radford et al., 2018, 2019)

Learning in Language Models

Text Corpus



Critics say that current voting systems used in the United States are inefficient and often lead to the inaccurate counting of votes. Miscounts can be especially damaging if an election is closely contested. Those critics would like the traditional systems to be replaced with far more efficient and trustworthy computerized voting systems.

In traditional voting, one major source of inaccuracy is that people accidentally vote for the wrong candidate. Voters usually have to find the name of their candidate on a large sheet of paper containing many names—the ballot—and make a small mark next to that name. People with poor eyesight can easily mark the wrong name. The computerized voting machines have an easy-to-use touch-screen technology: to cast a vote, a voter needs only to touch the candidate's name on the screen to record a vote for that candidate; voters can even have the computer magnify the name for easier viewing.

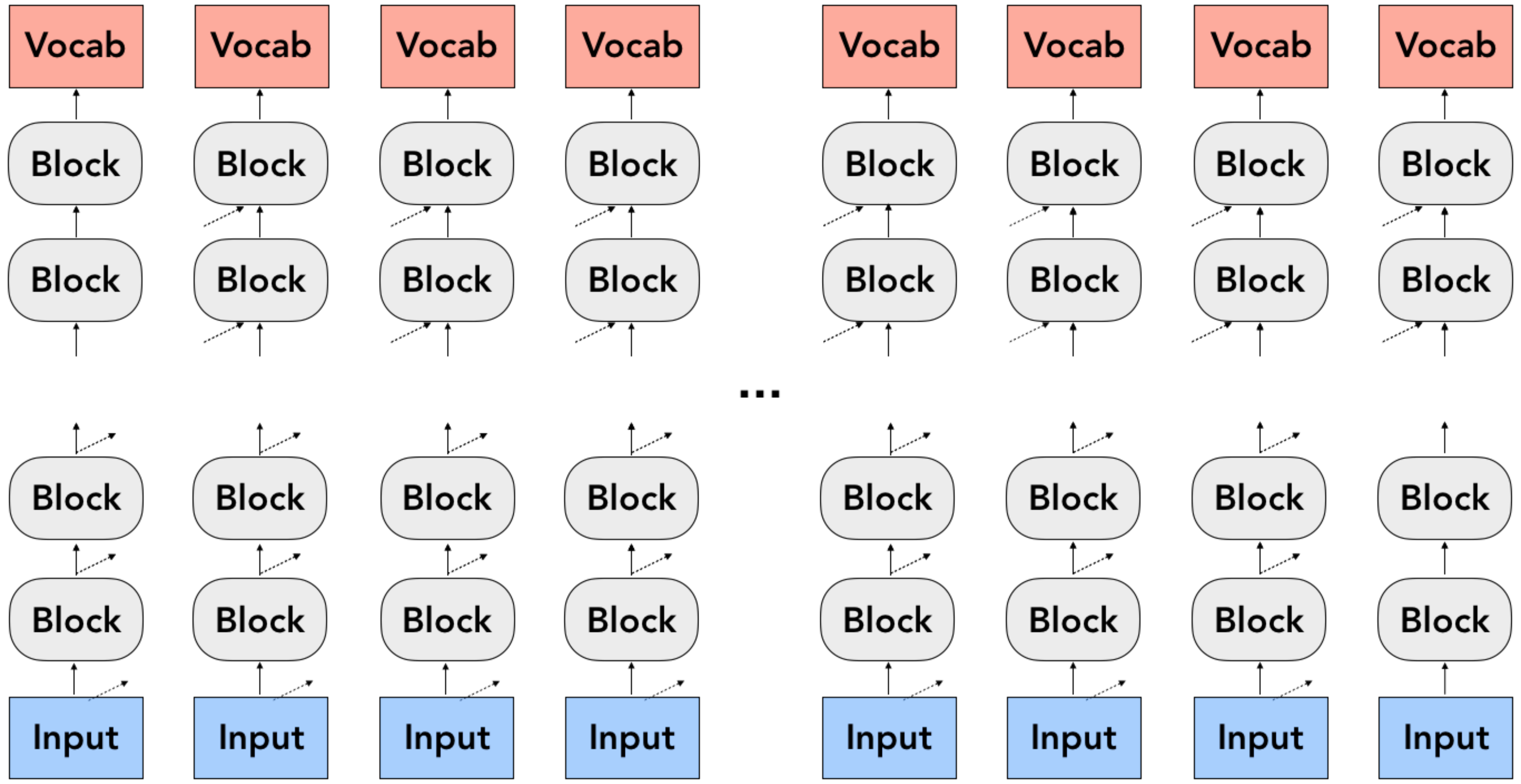
Another major problem with old voting systems is that they rely heavily on people to count the votes. Officials must often count up the votes one by one, going through every ballot and recording the vote. Since they have to deal with thousands of ballots, it is almost inevitable that they will make mistakes. If an error is detected, a long and expensive recount has to take place. In contrast, computerized systems remove the possibility of human error, since all the vote counting is done quickly and automatically by the computers.

Finally some people say it is too risky to implement complicated voting technology nationwide. But without giving it a thought, governments and individuals alike trust other complex computer technology every day to be perfectly accurate in banking transactions as well as in the communication of highly sensitive information.

Used to

Learn

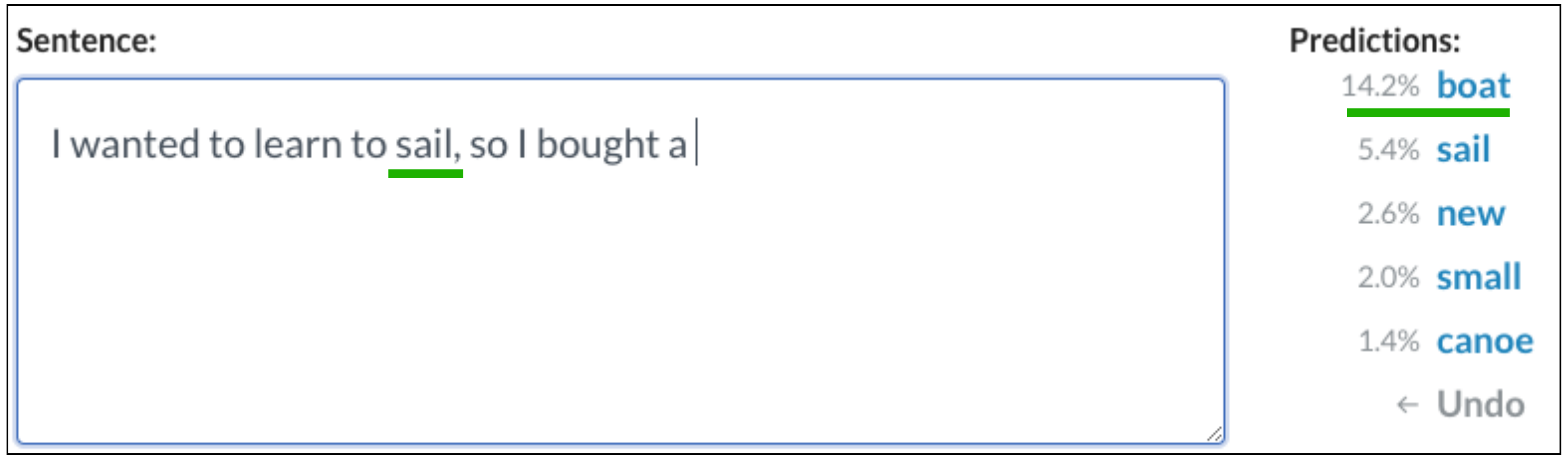
Transformer Language Model



Knowledge in Language Models

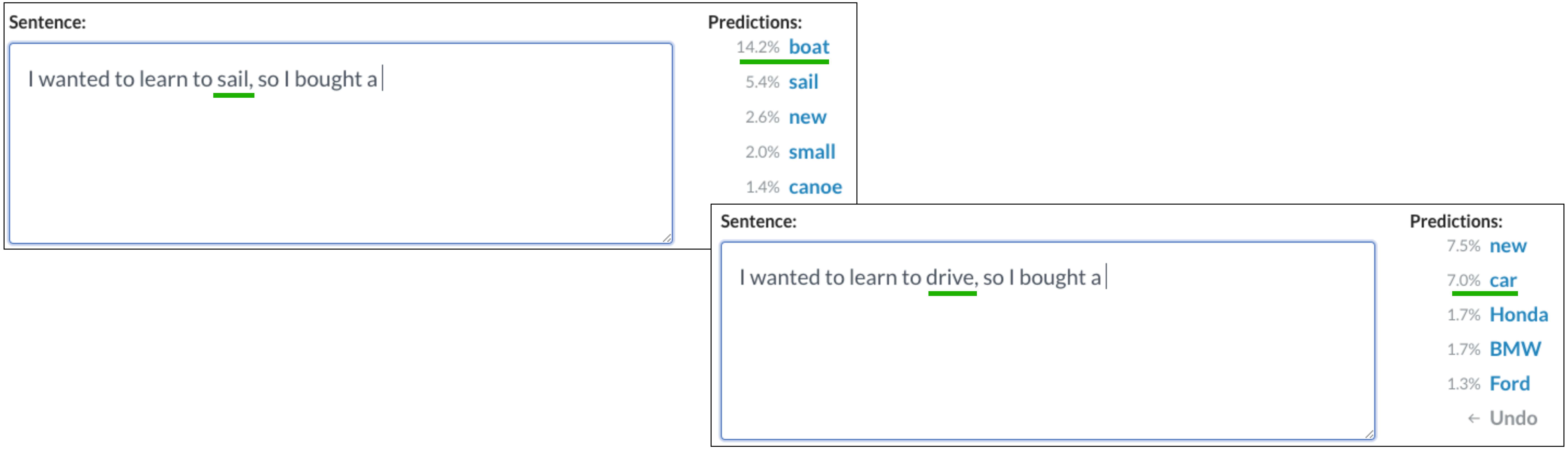
https://demo.allennlp.org/next-token-lm

Knowledge in Language Models



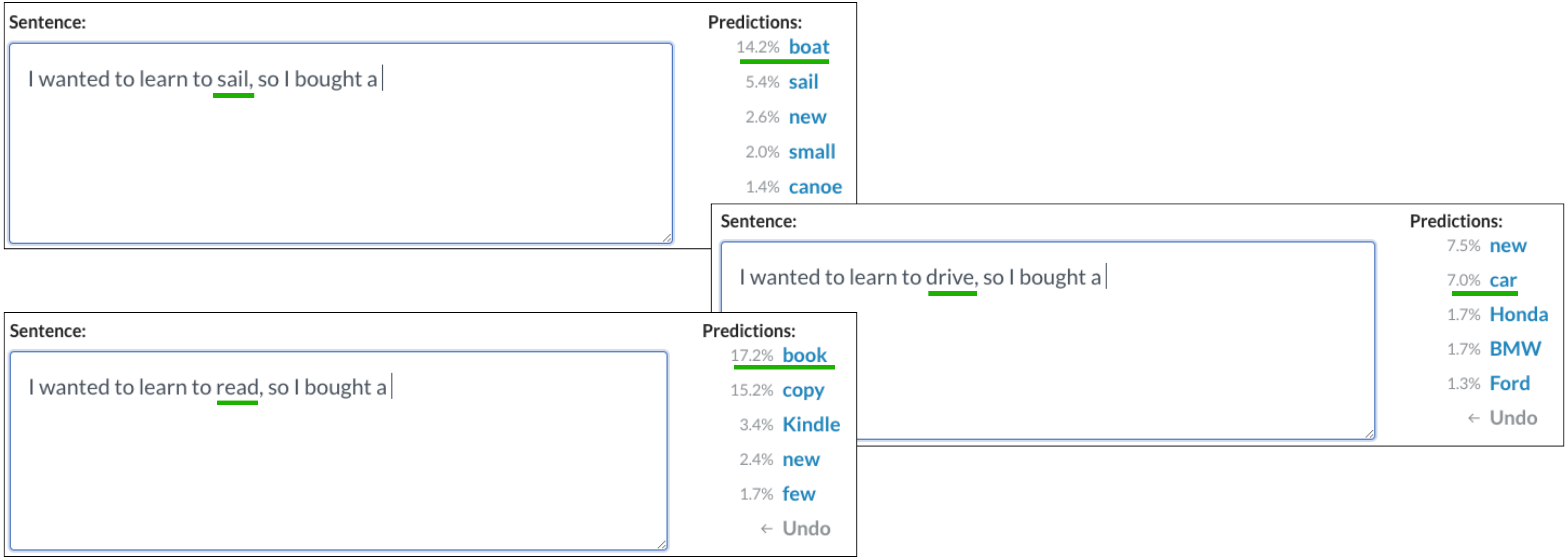
https://demo.allennlp.org/next-token-lm

Knowledge in Language Models



https://demo.allennlp.org/next-token-lm

Knowledge in Language Models



https://demo.allennlp.org/next-token-lm

Knowledge in Language Models

Sentence:

I wanted to learn to sail, so I bought a

Predictions:

- 14.2% boat
- 5.4% sail
- 2.6% new
- 2.0% small
- 1.4% canoe

Sentence:

I wanted to learn to drive, so I bought a

Predictions:

- 7.5% new
- 7.0% car
- 1.7% Honda
- 1.7% BMW
- 1.3% Ford

← Undo

Sentence:

I wanted to learn to read, so I bought a

Predictions:

- 17.2% book
- 15.2% copy
- 3.4% Kindle
- 2.4% new
- 1.7% few

Sentence:

I wanted to learn to fly, so I bought a

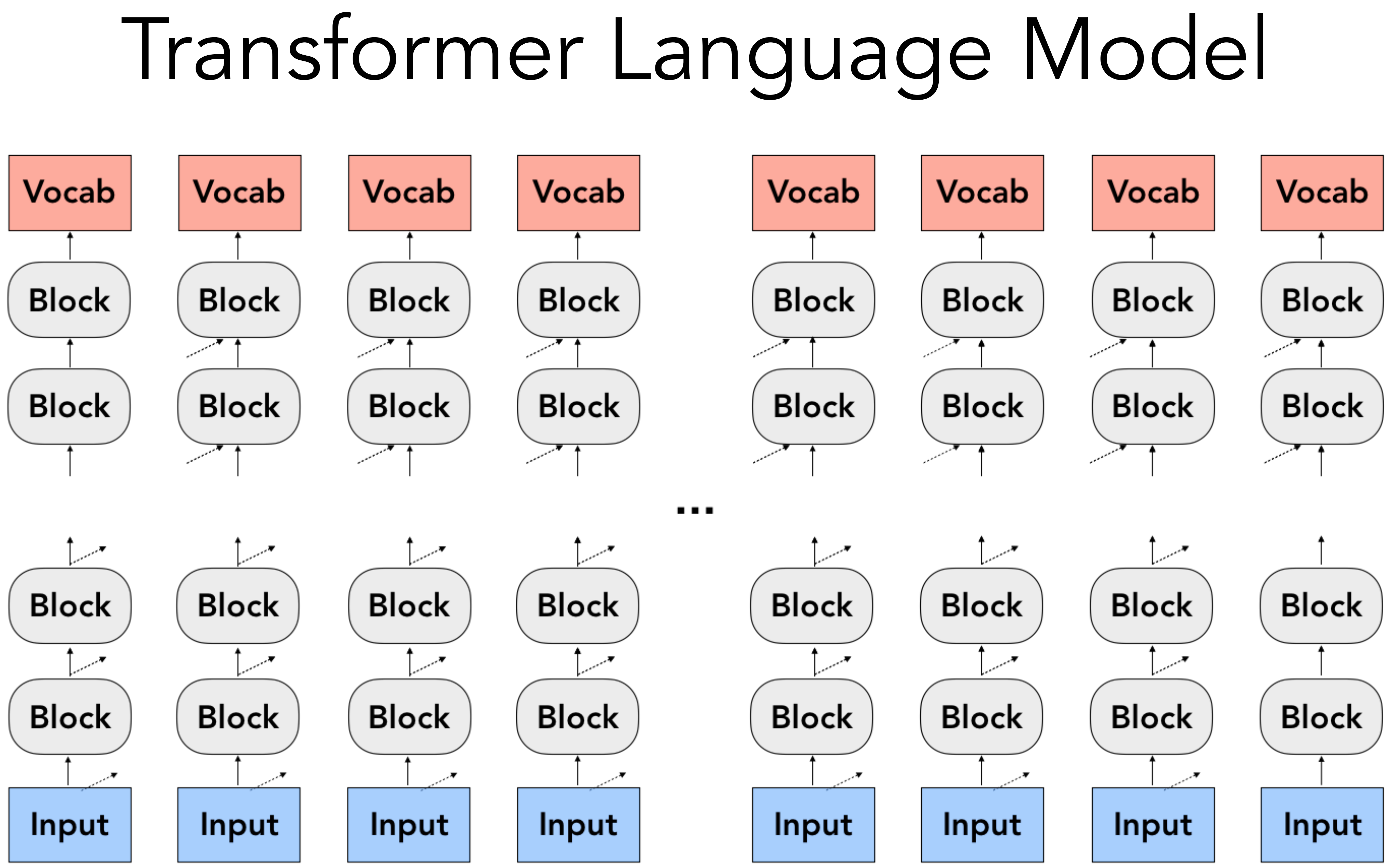
Predictions:

- 5.3% plane
- 3.8% new
- 1.6% small
- 1.6% Boeing
- 1.5% jet

← Undo

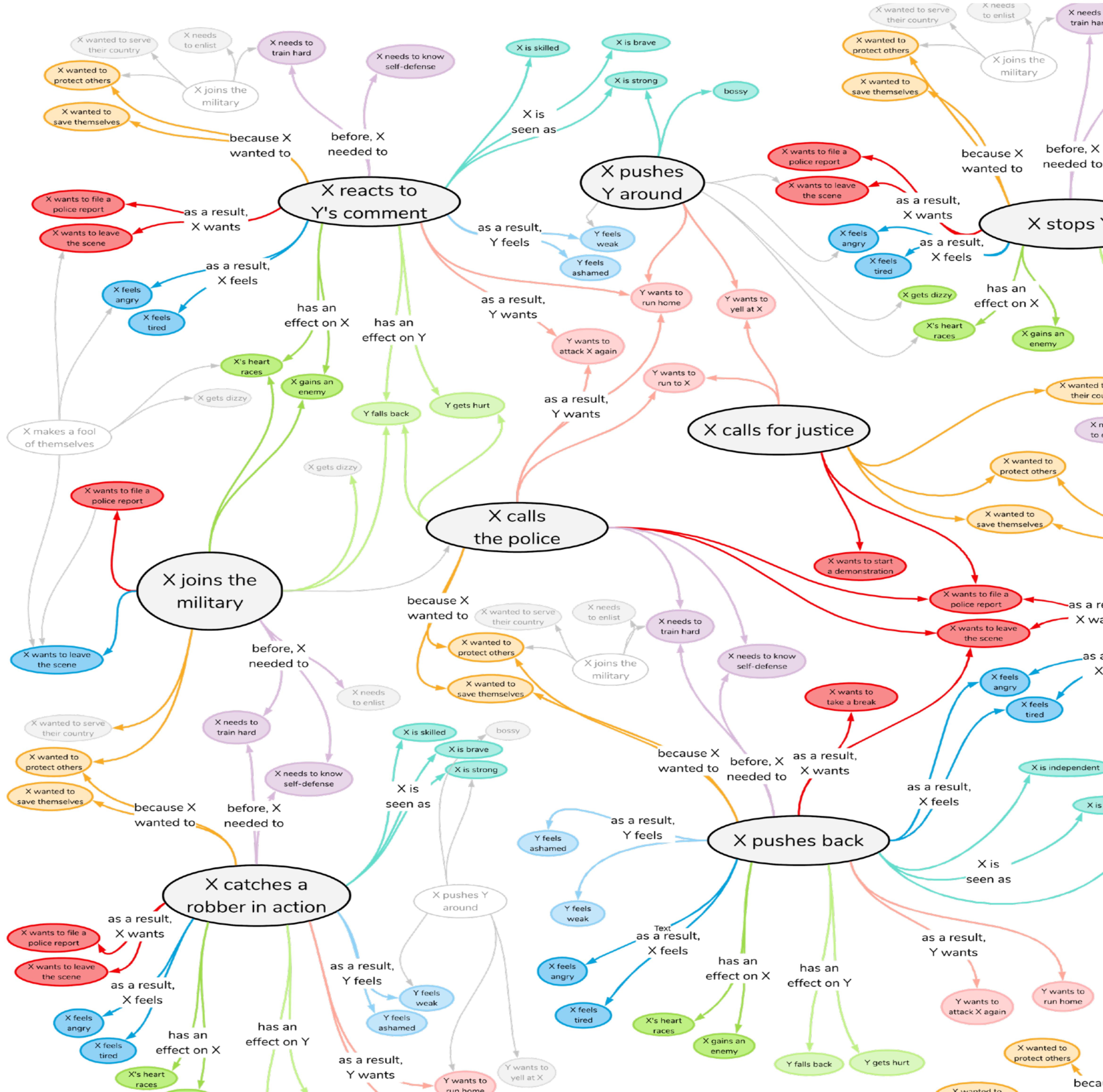
https://demo.allennlp.org/next-token-lm

From Unstructured to Structured Knowledge



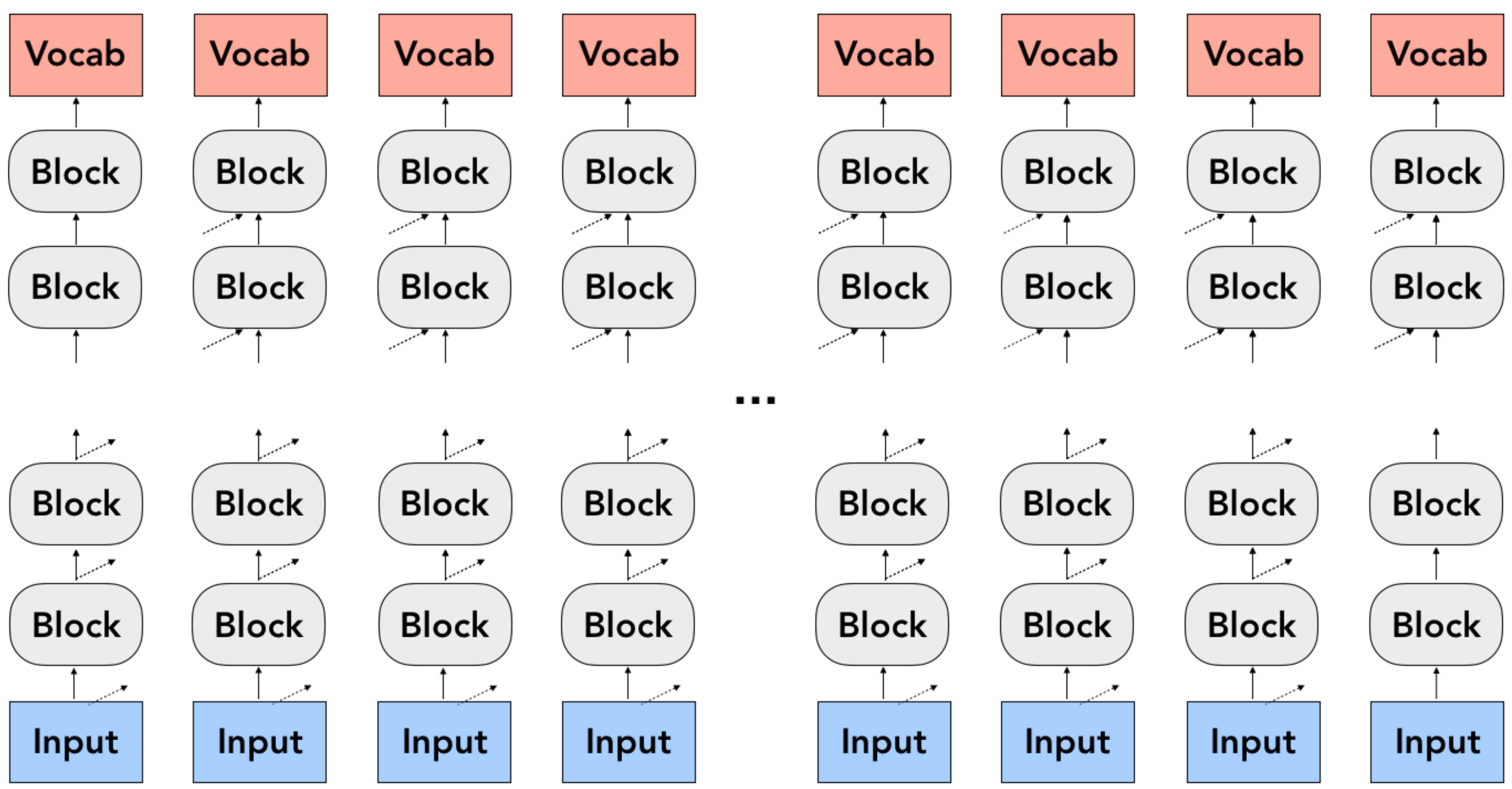
Used to

Generate



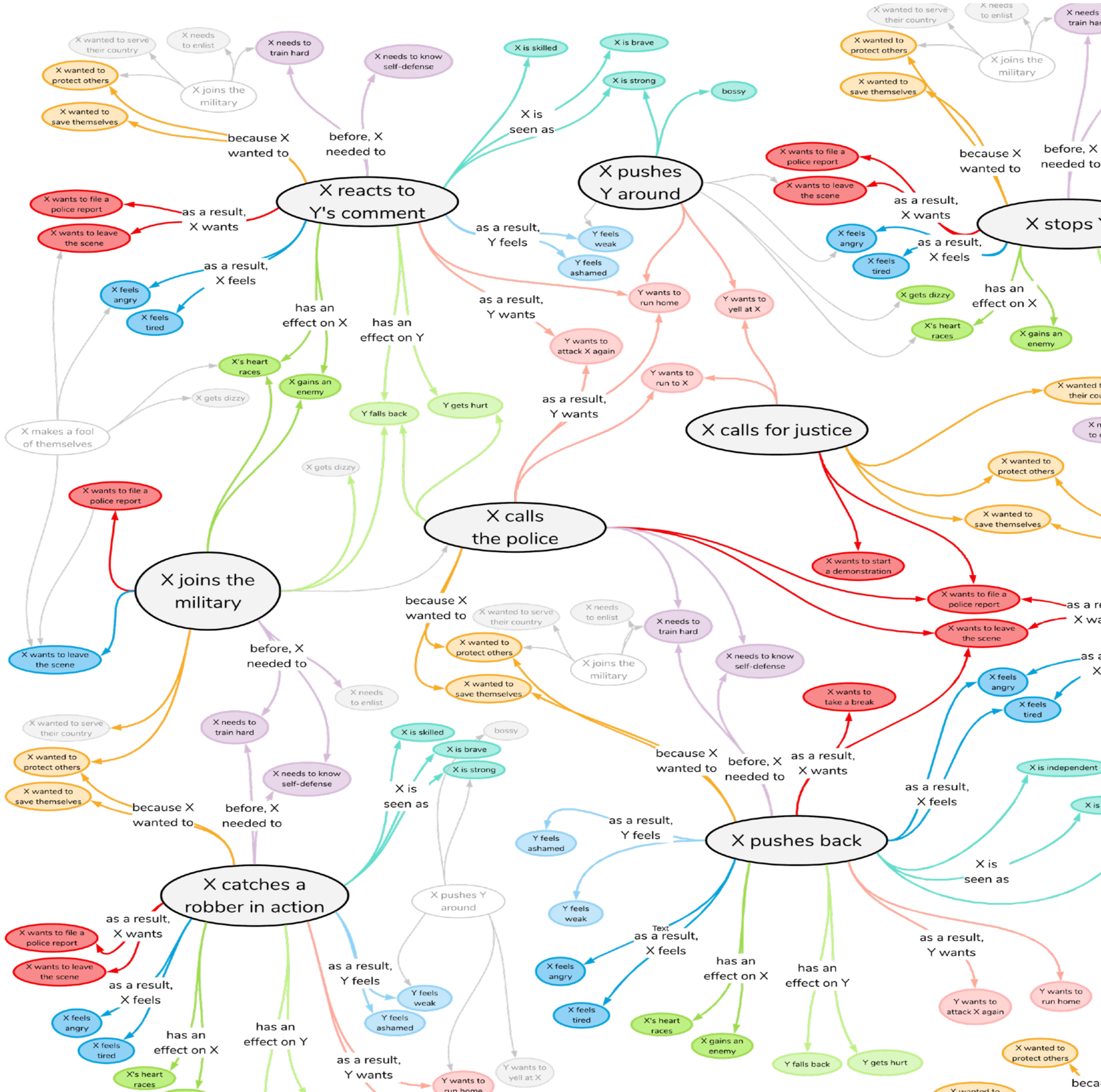
From Unstructured to Structured Knowledge

Transformer Language Model

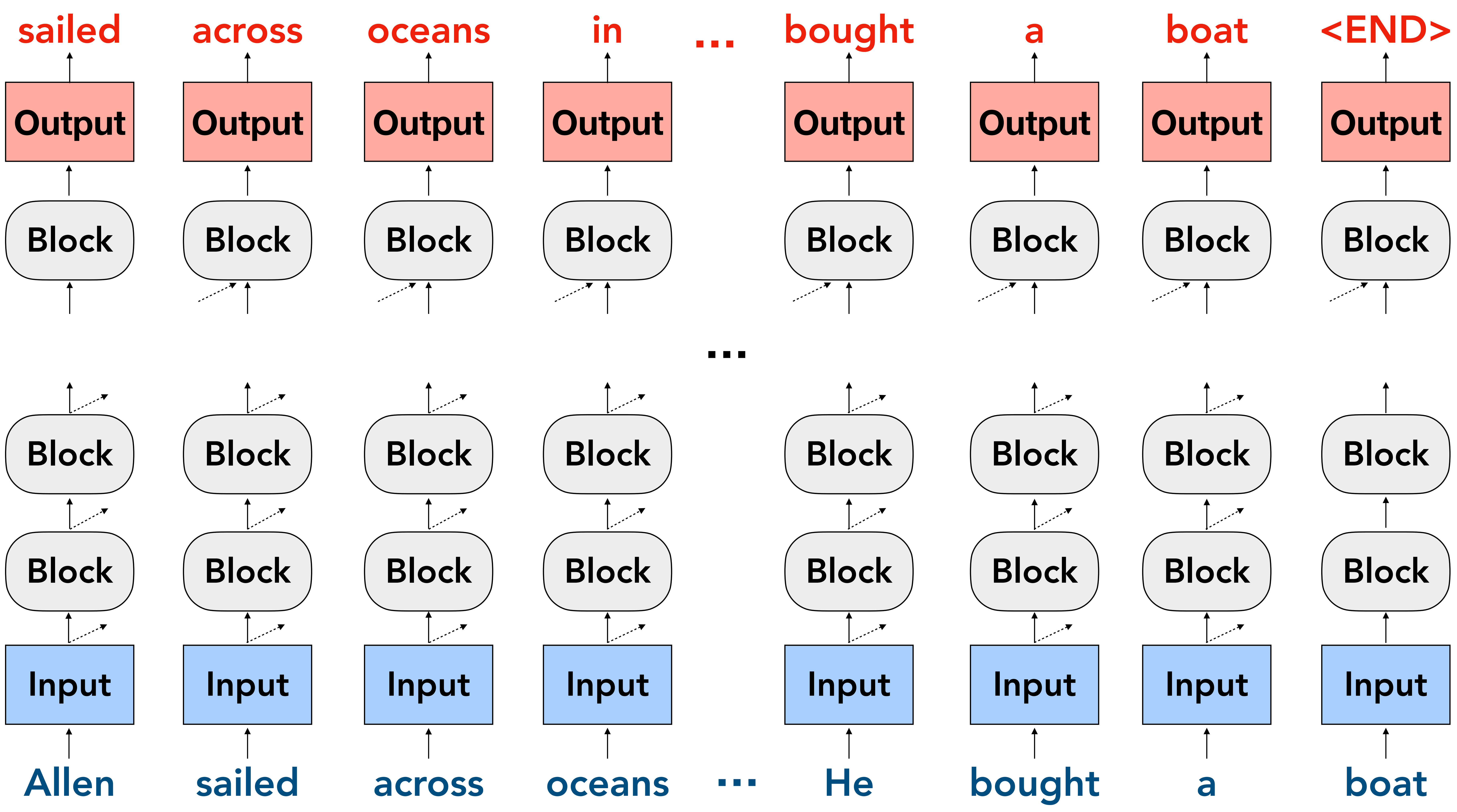


Used to

Generate

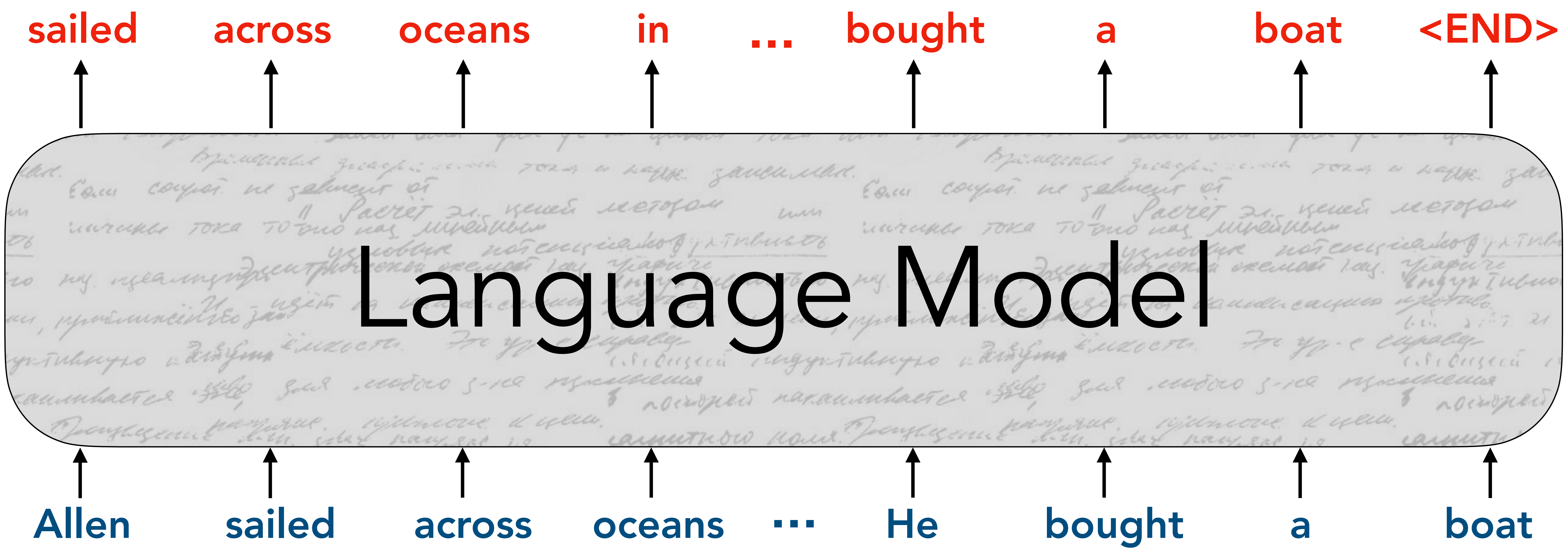


Transformer Language Models



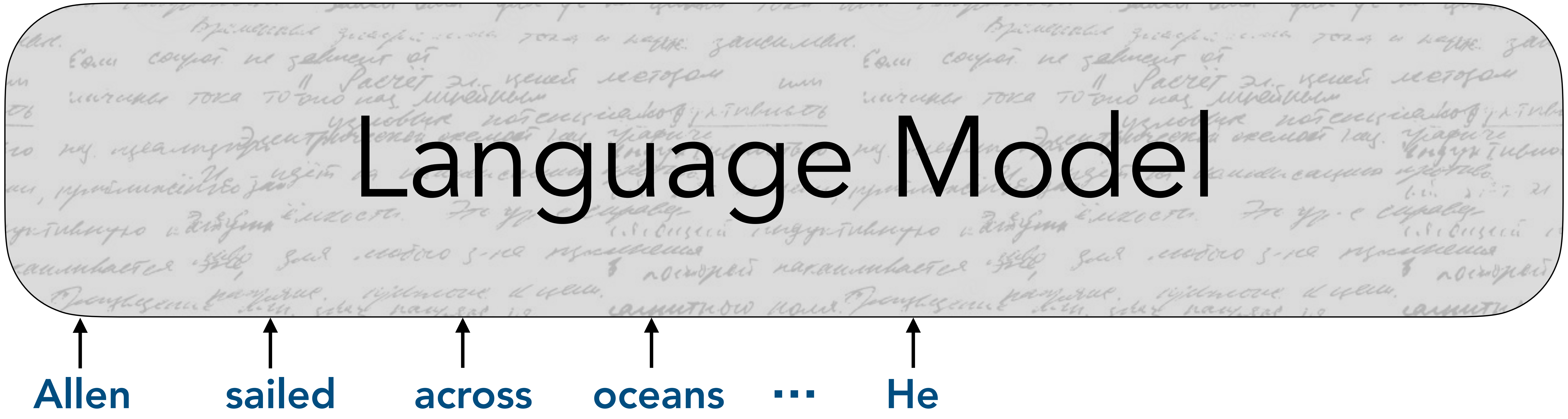
Transformer Language Models

- Trained to generate the next word given a set of preceding words



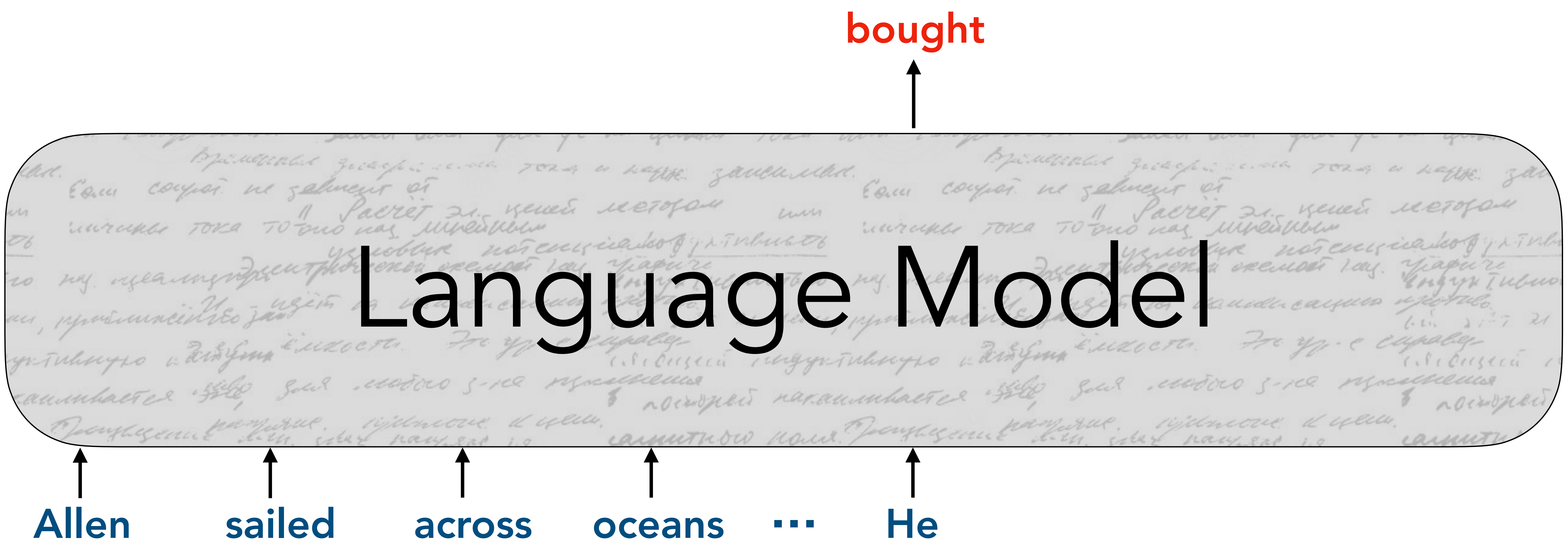
Transformer Language Models

- Trained to generate the next word given a set of preceding words
- Follow-up tokens can be generated using generated tokens as input



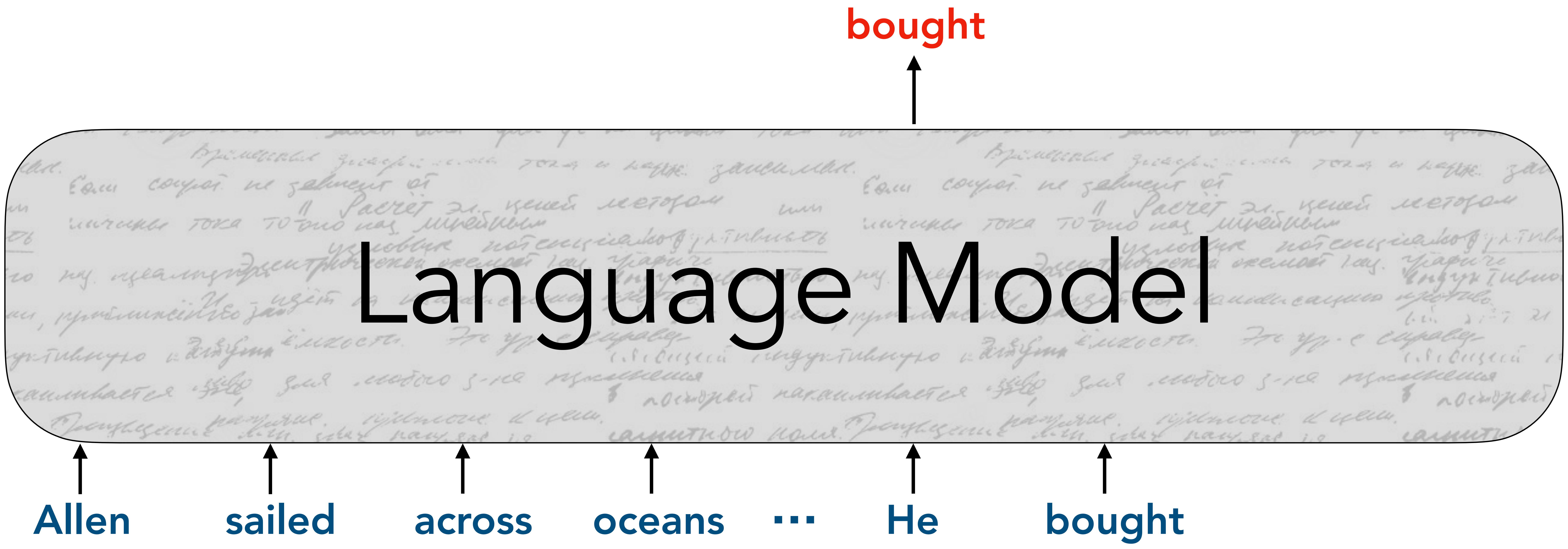
Transformer Language Models

- Trained to generate the next word given a set of preceding words
- Follow-up tokens can be generated using generated tokens as input



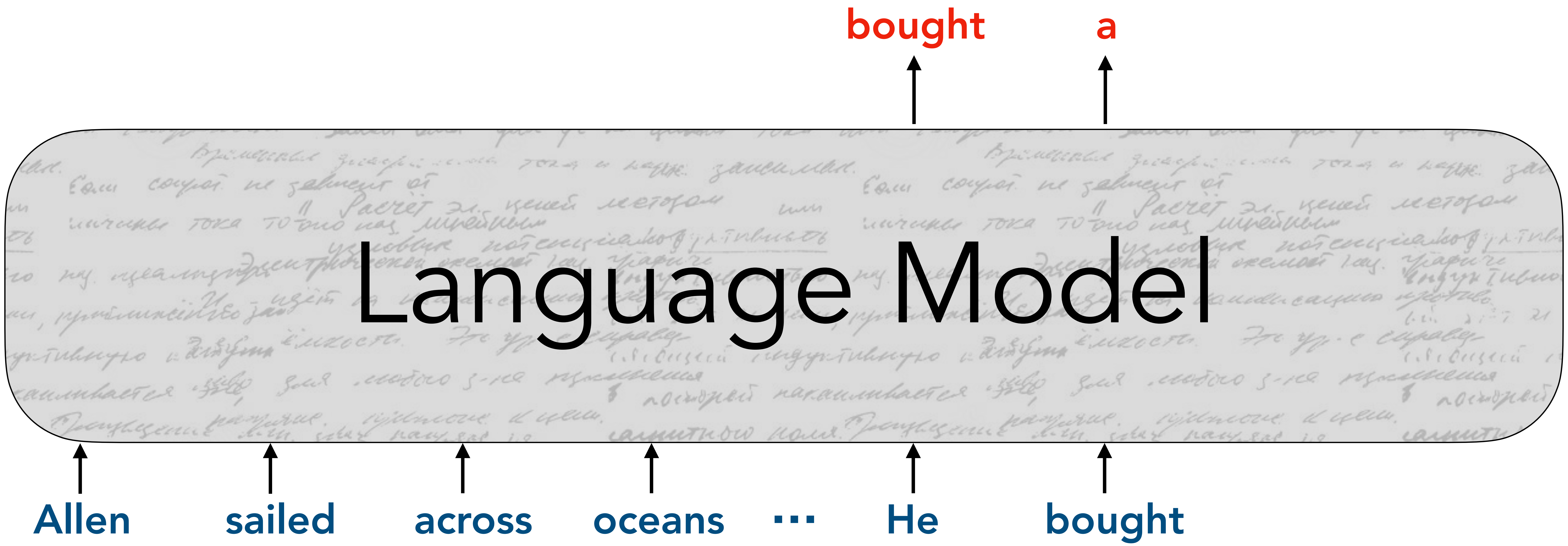
Transformer Language Models

- Trained to generate the next word given a set of preceding words
- Follow-up tokens can be generated using generated tokens as input



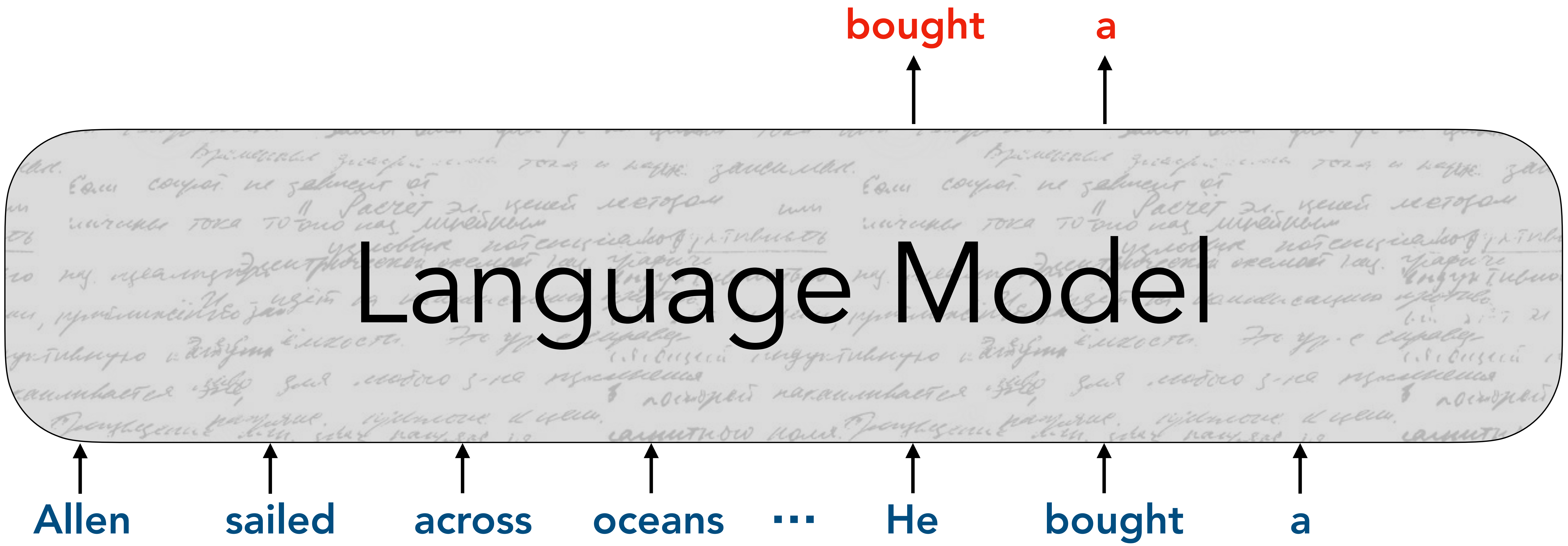
Transformer Language Models

- Trained to generate the next word given a set of preceding words
- Follow-up tokens can be generated using generated tokens as input



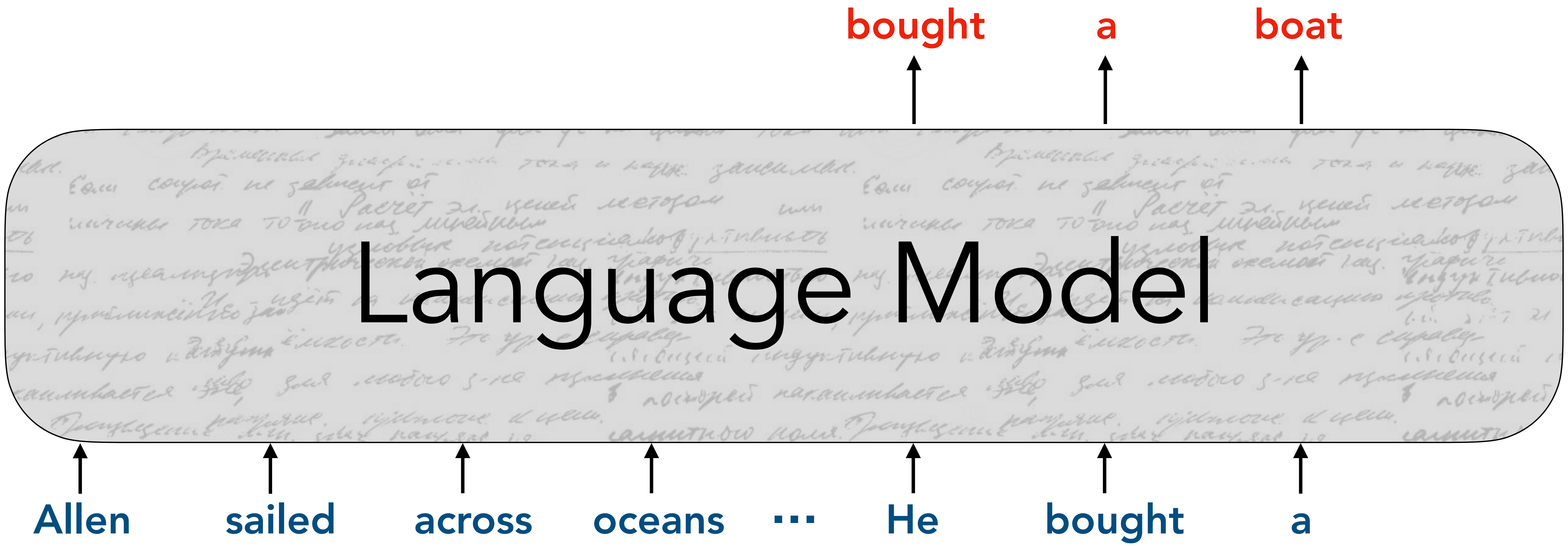
Transformer Language Models

- Trained to generate the next word given a set of preceding words
- Follow-up tokens can be generated using generated tokens as input



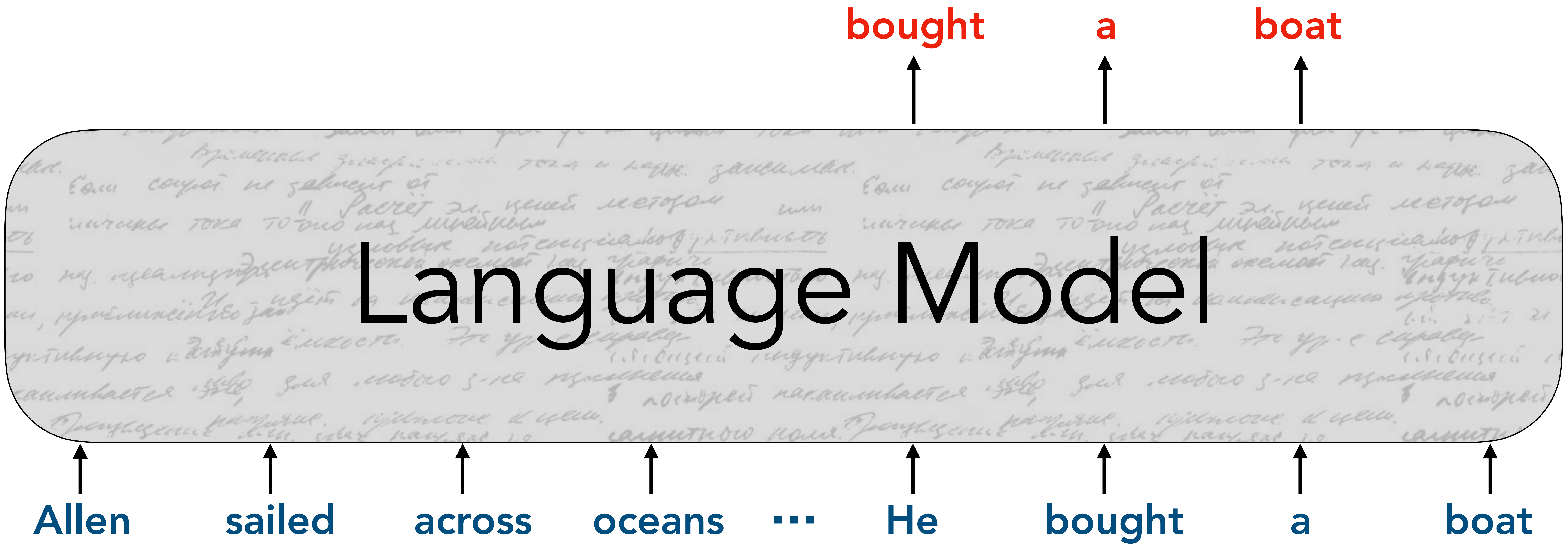
Transformer Language Models

- Trained to generate the next word given a set of preceding words
- Follow-up tokens can be generated using generated tokens as input



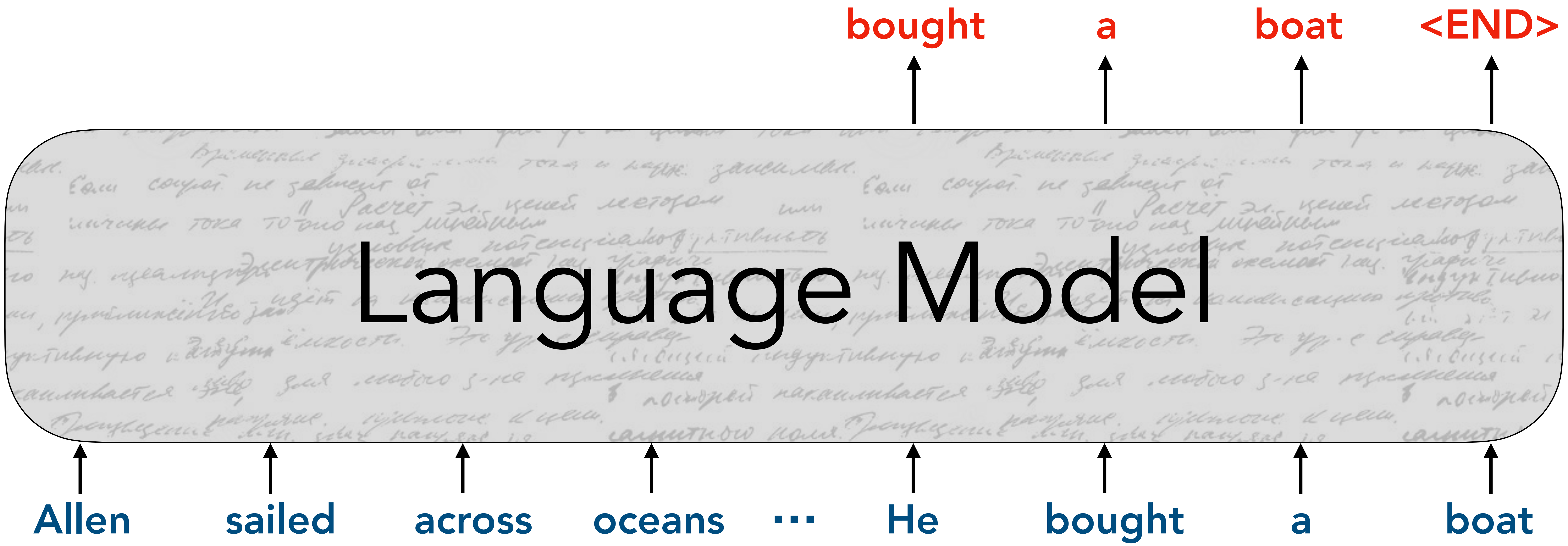
Transformer Language Models

- Trained to generate the next word given a set of preceding words
- Follow-up tokens can be generated using generated tokens as input



Transformer Language Models

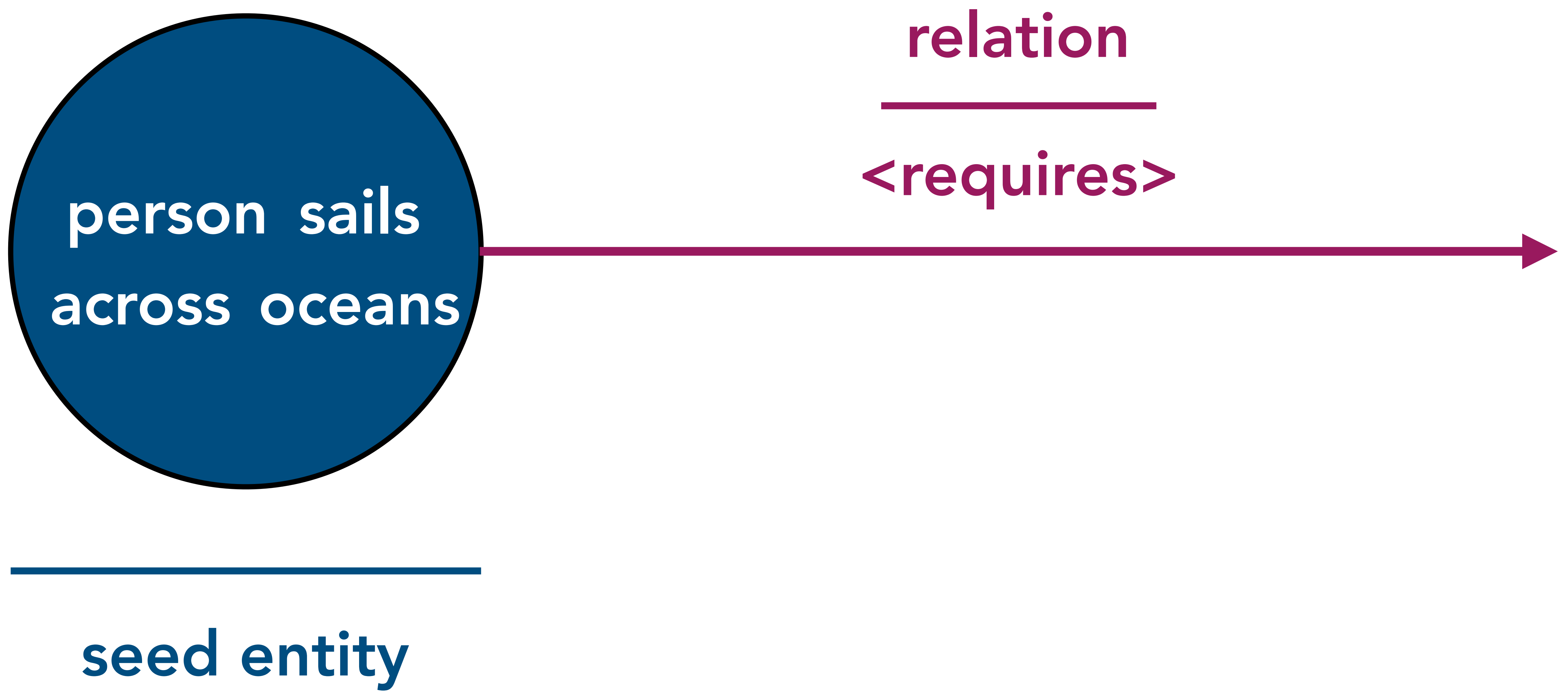
- Trained to generate the next word given a set of preceding words
- Follow-up tokens can be generated using generated tokens as input



Structure of Knowledge Tuple

Structure of Knowledge Tuple

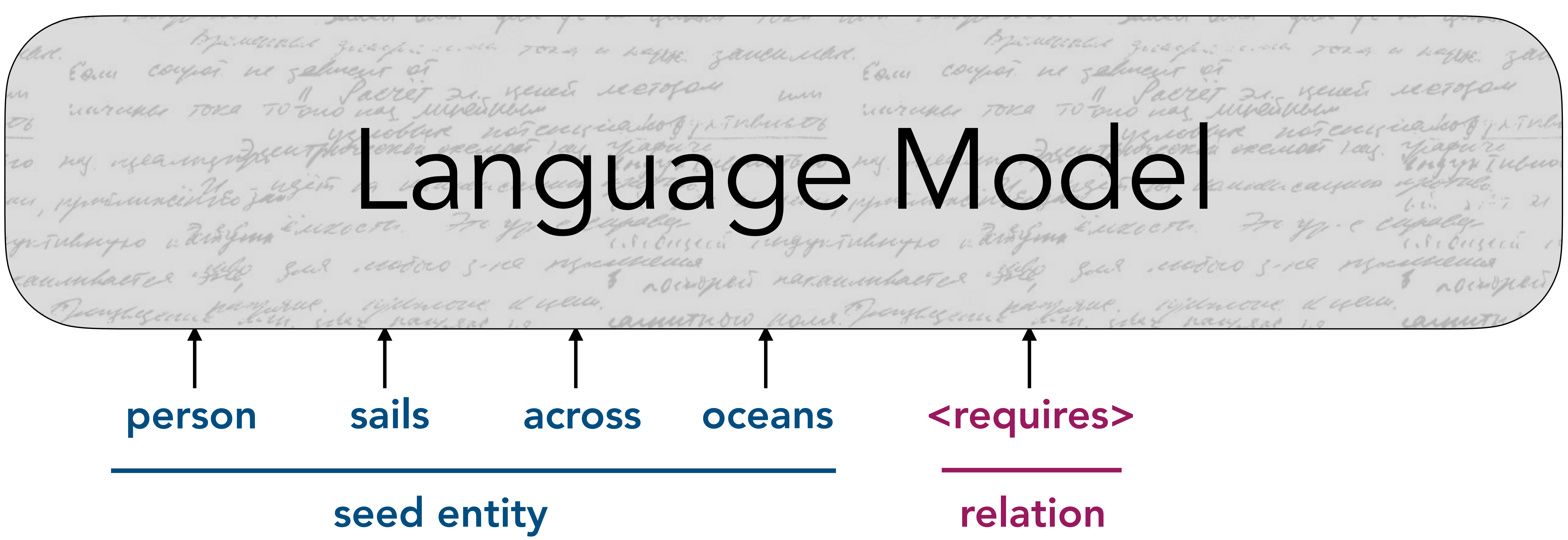
Structure of Knowledge Tuple



Structure of Knowledge Tuple

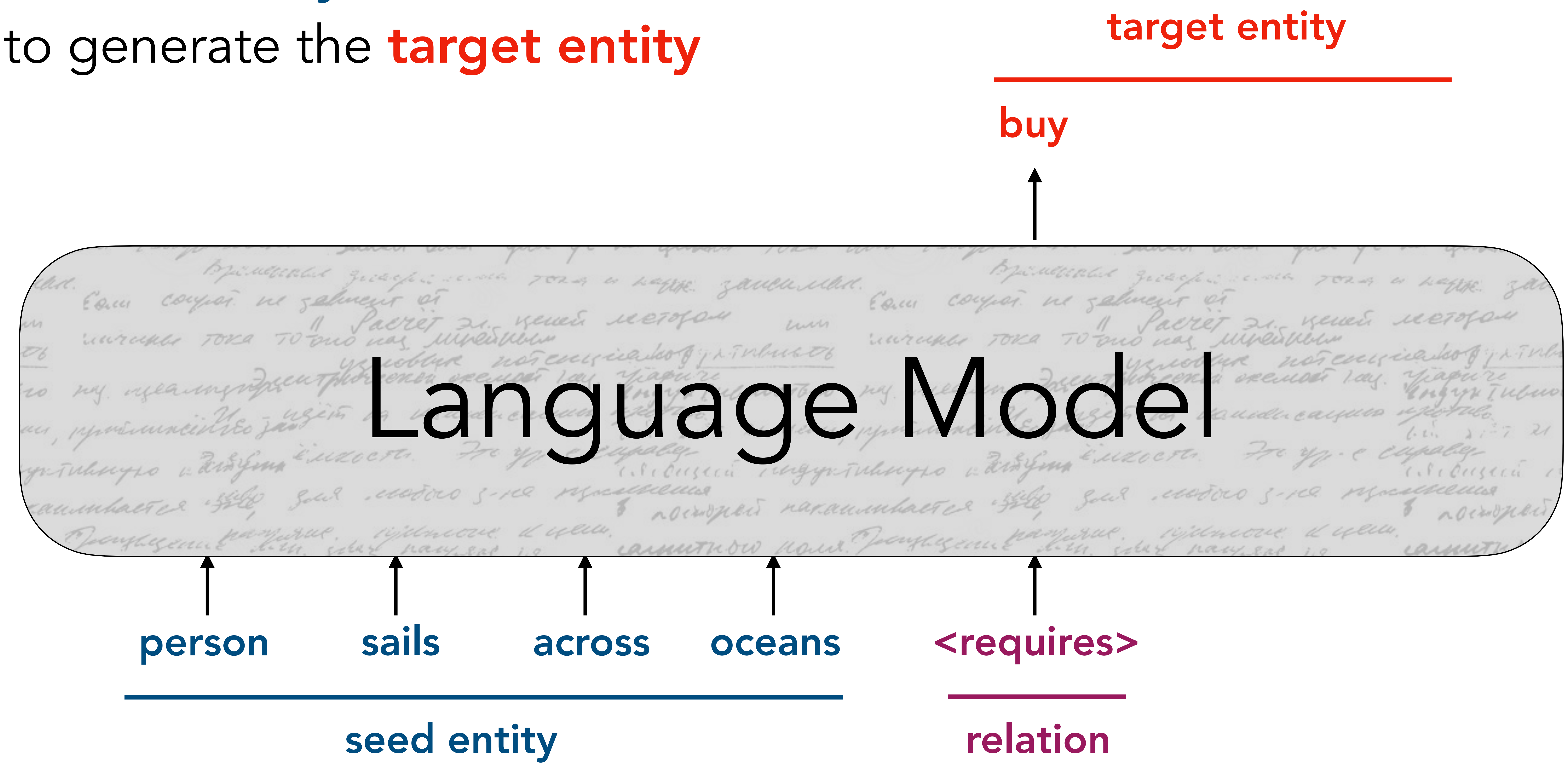
Learning Structure of Knowledge

Given a **seed entity** and a **relation**,
learn to generate the **target entity**



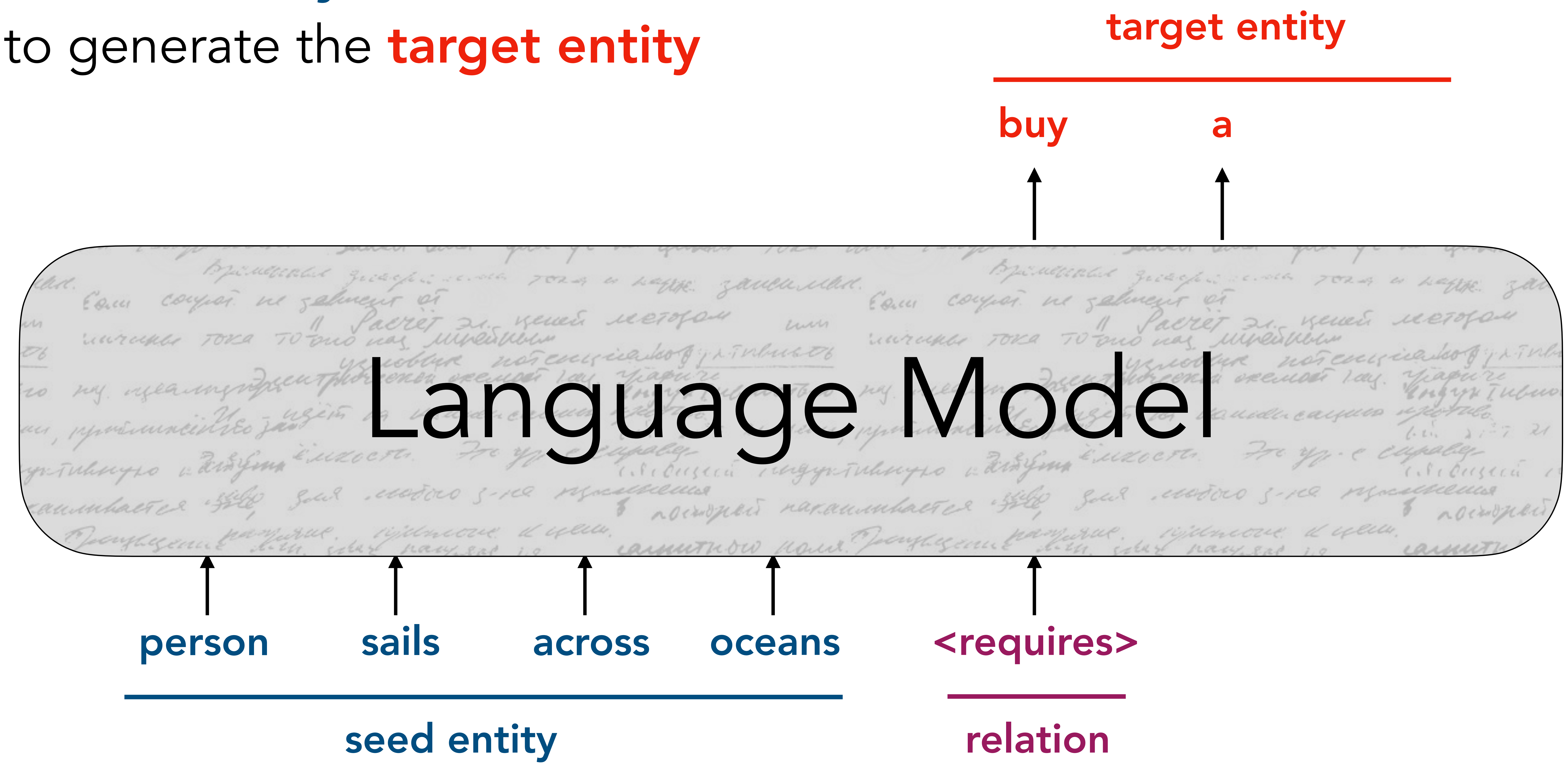
Learning Structure of Knowledge

Given a **seed entity** and a **relation**,
learn to generate the **target entity**



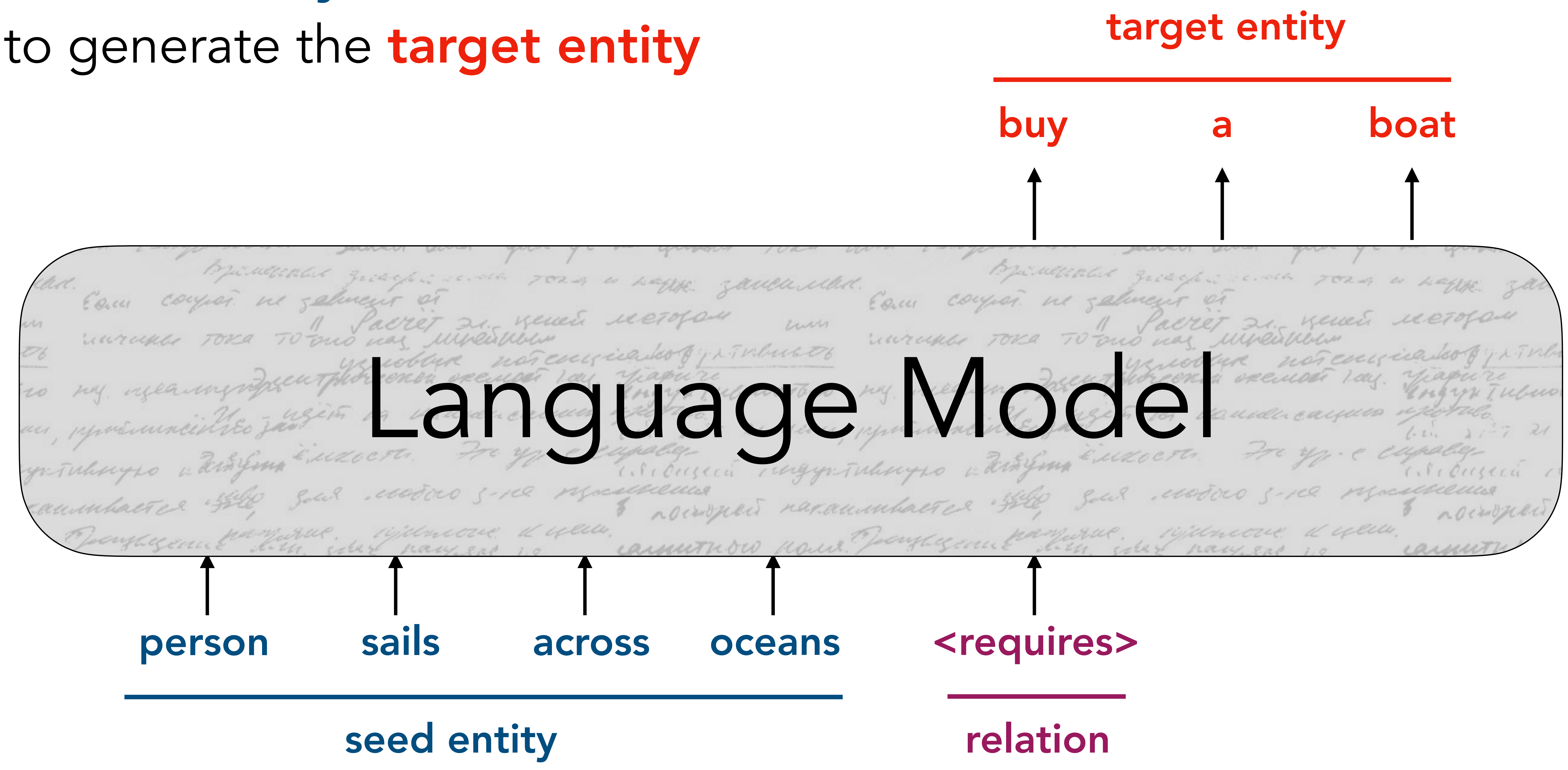
Learning Structure of Knowledge

Given a **seed entity** and a **relation**,
learn to generate the **target entity**



Learning Structure of Knowledge

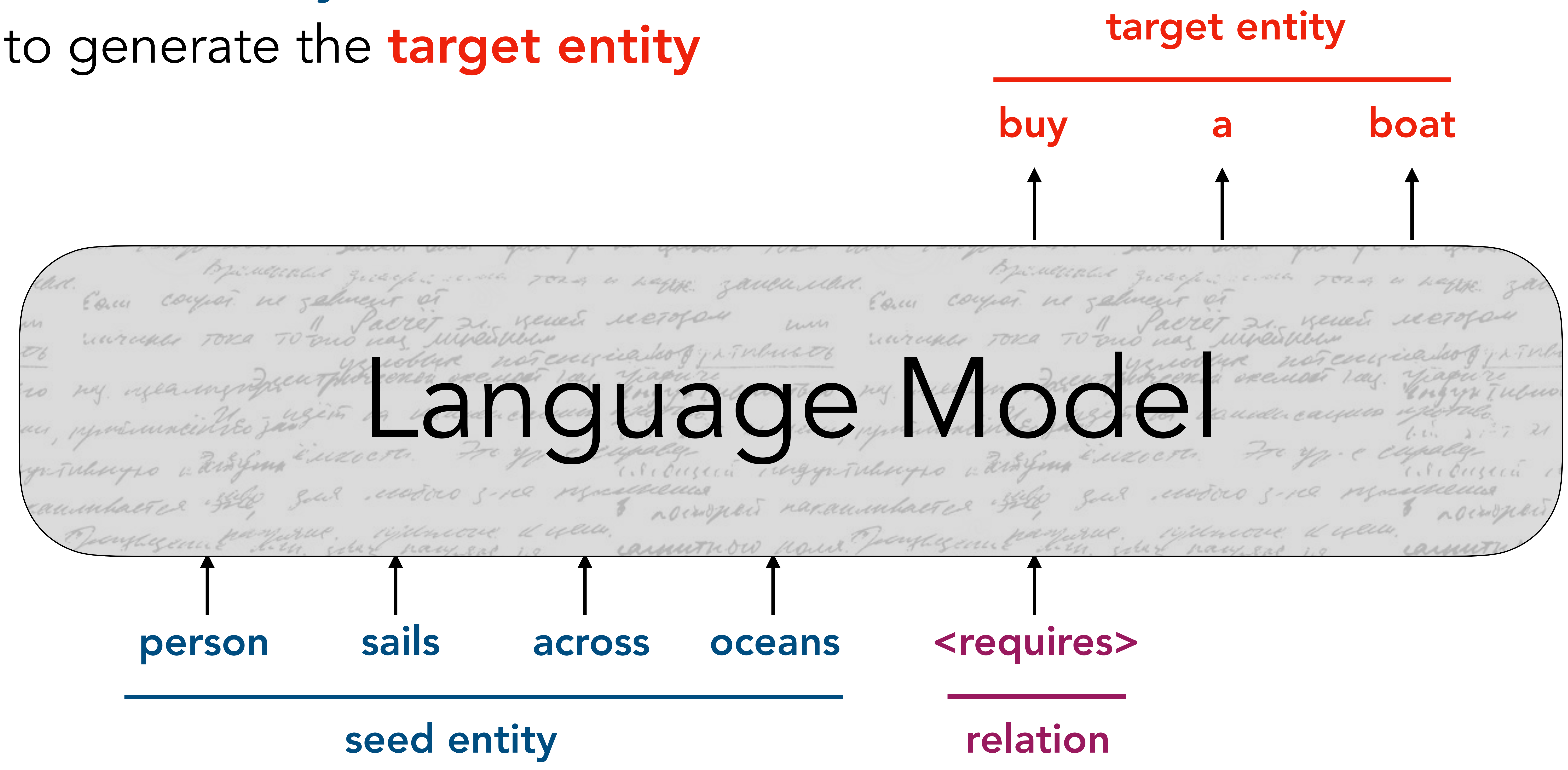
Given a **seed entity** and a **relation**,
learn to generate the **target entity**



Learning Structure of Knowledge

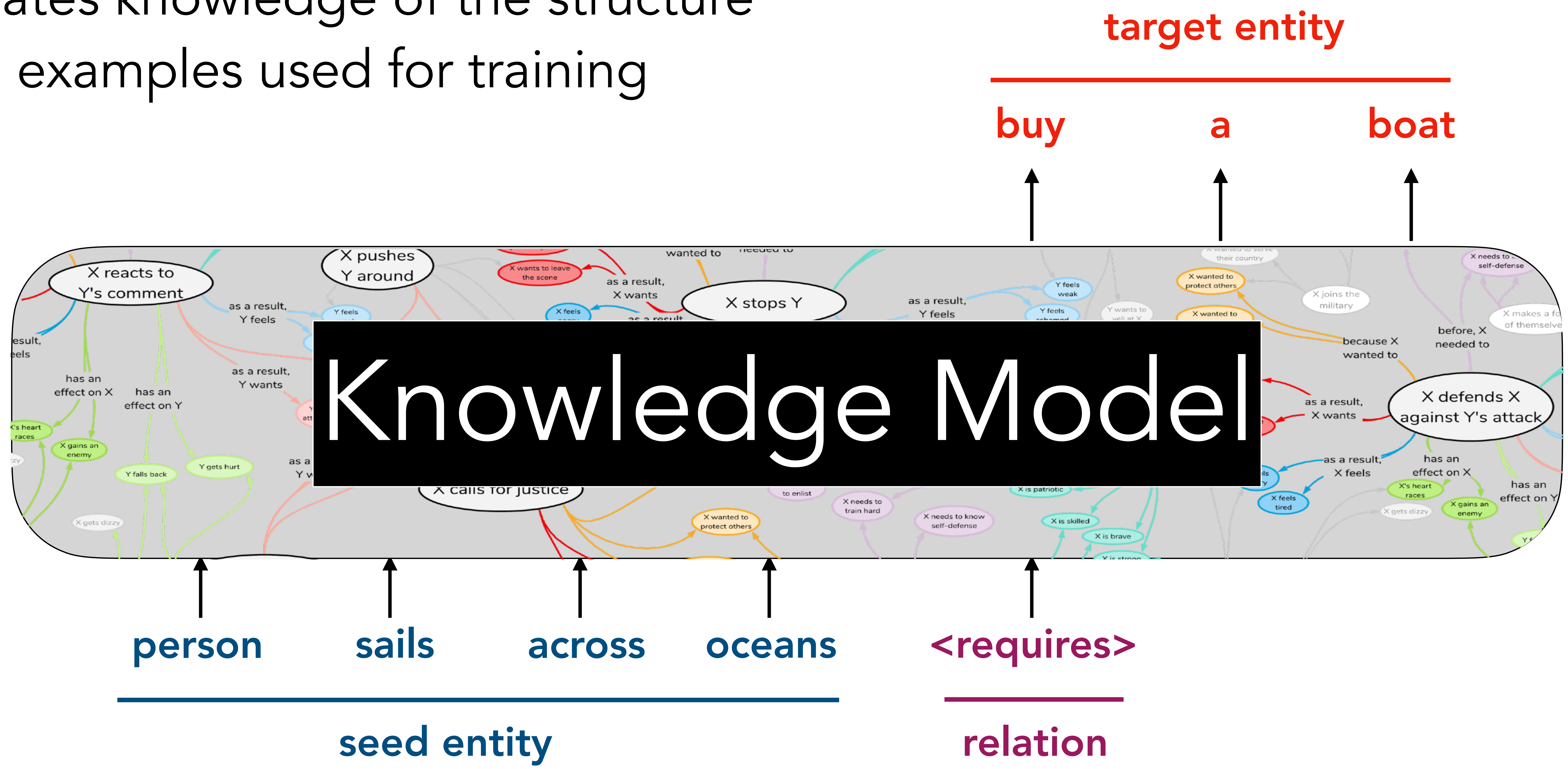
Given a **seed entity** and a **relation**, learn to generate the **target entity**

$$\mathscr{L} = -\sum \log P(\textbf{target words} \,|\, \textbf{seed words, relation})$$



Learning Structure of Knowledge

Language Model ⟶ **Knowledge Model:**
generates knowledge of the structure
of the examples used for training

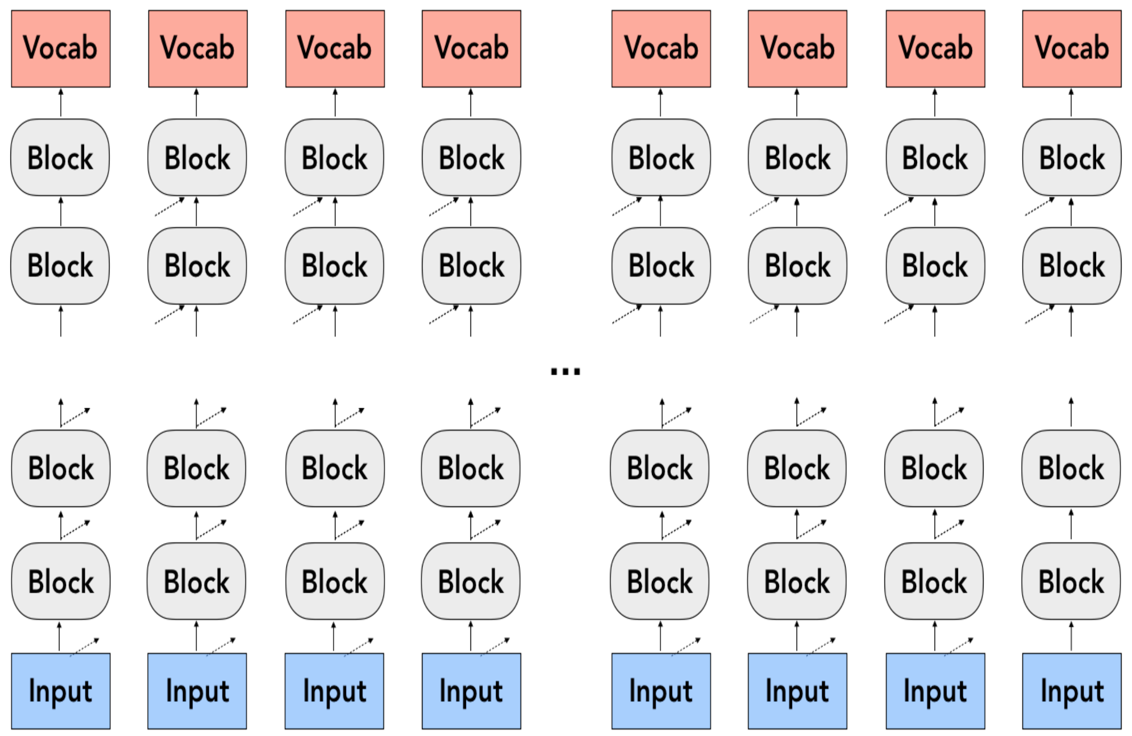


Generate commonsense knowledge for **any** input concept

COMmonsE**nse** **T**ransformers

Commonsense Transformers

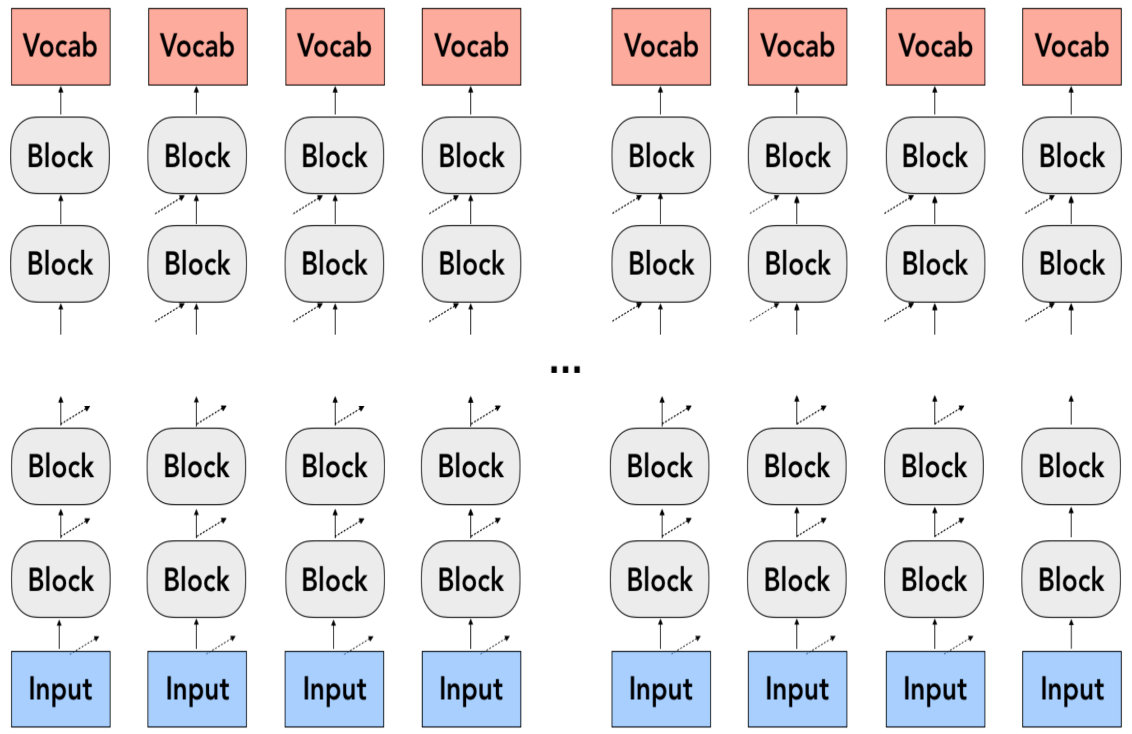
- Language models **implicitly represent knowledge**



Transformer
Language Model

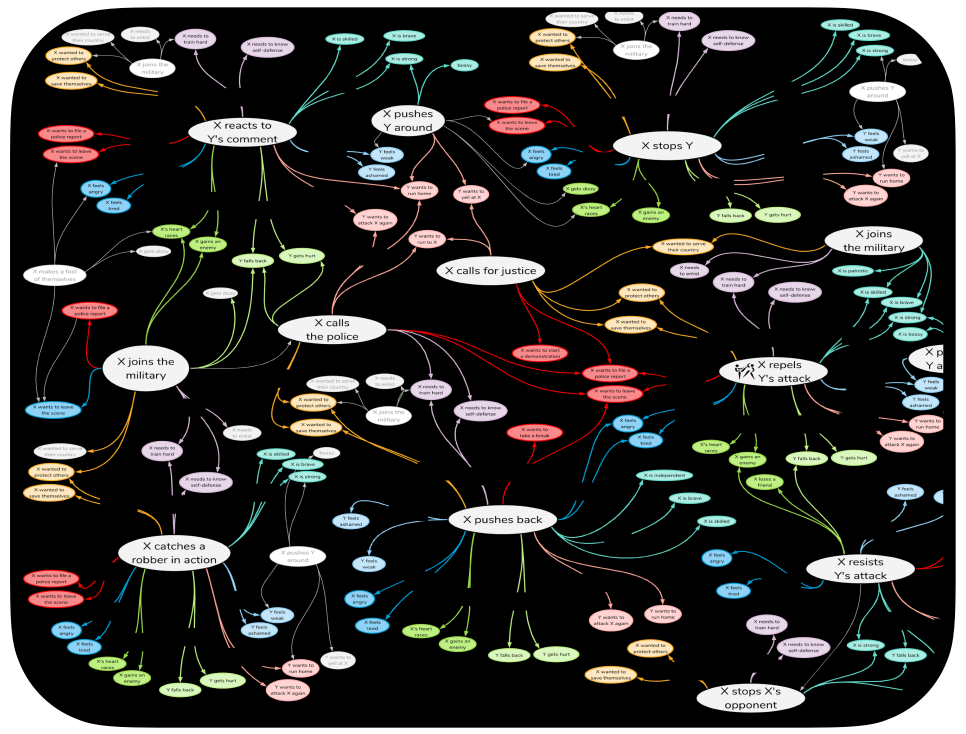
Commonsense Transformers

- Language models **implicitly represent knowledge**
- Re-train them on knowledge graphs to **learn structure of knowledge**



Transformer
Language Model

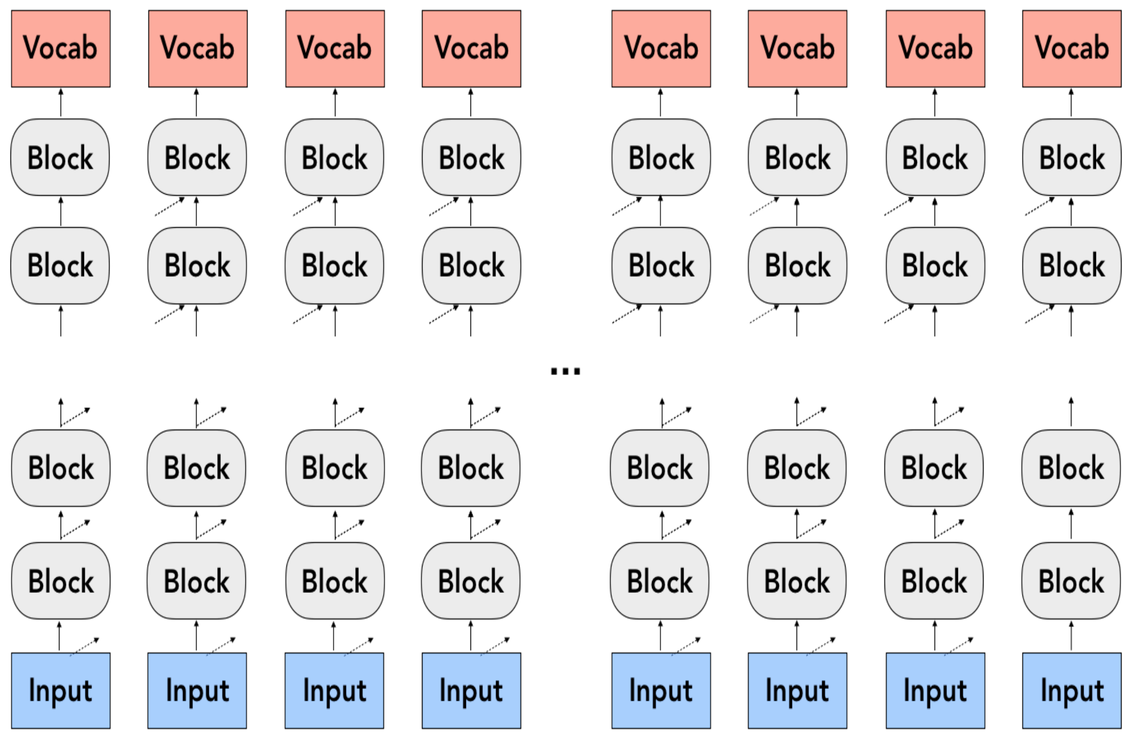
+



Seed Knowledge
Graph

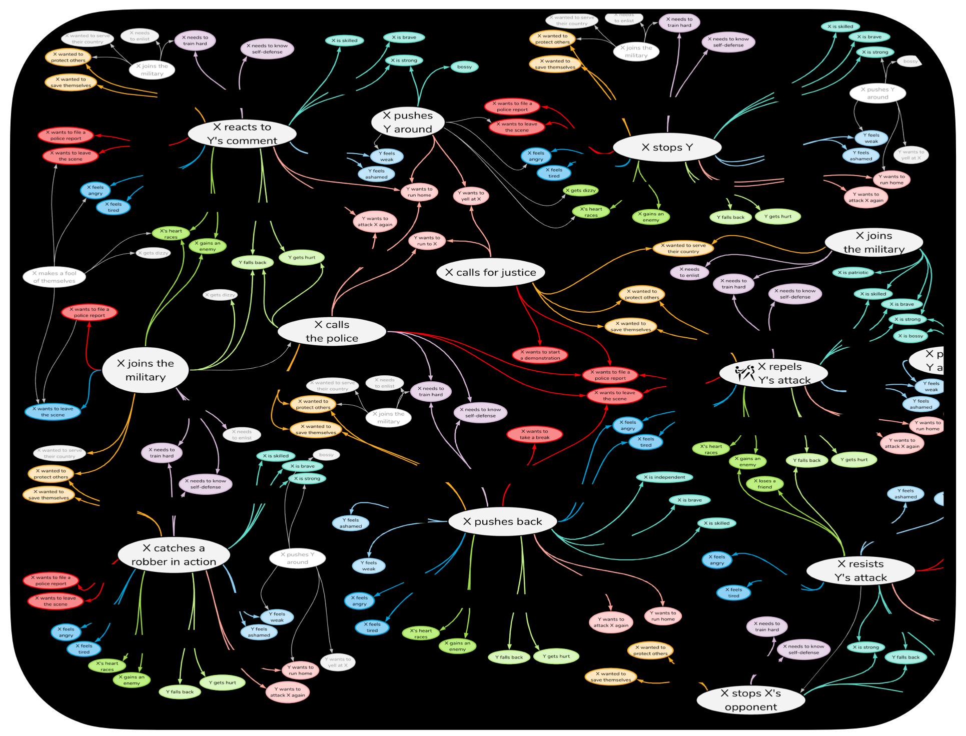
Commonsense Transformers

- Language models **implicitly represent knowledge**
- Re-train them on knowledge graphs to **learn structure of knowledge**
- Resulting knowledge model **generalizes structure** to other concepts



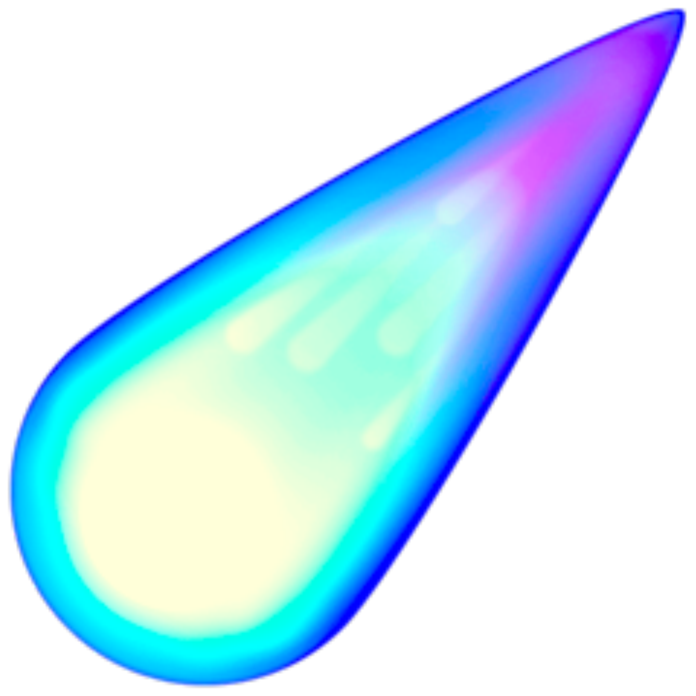
Transformer Language Model

+



Seed Knowledge Graph

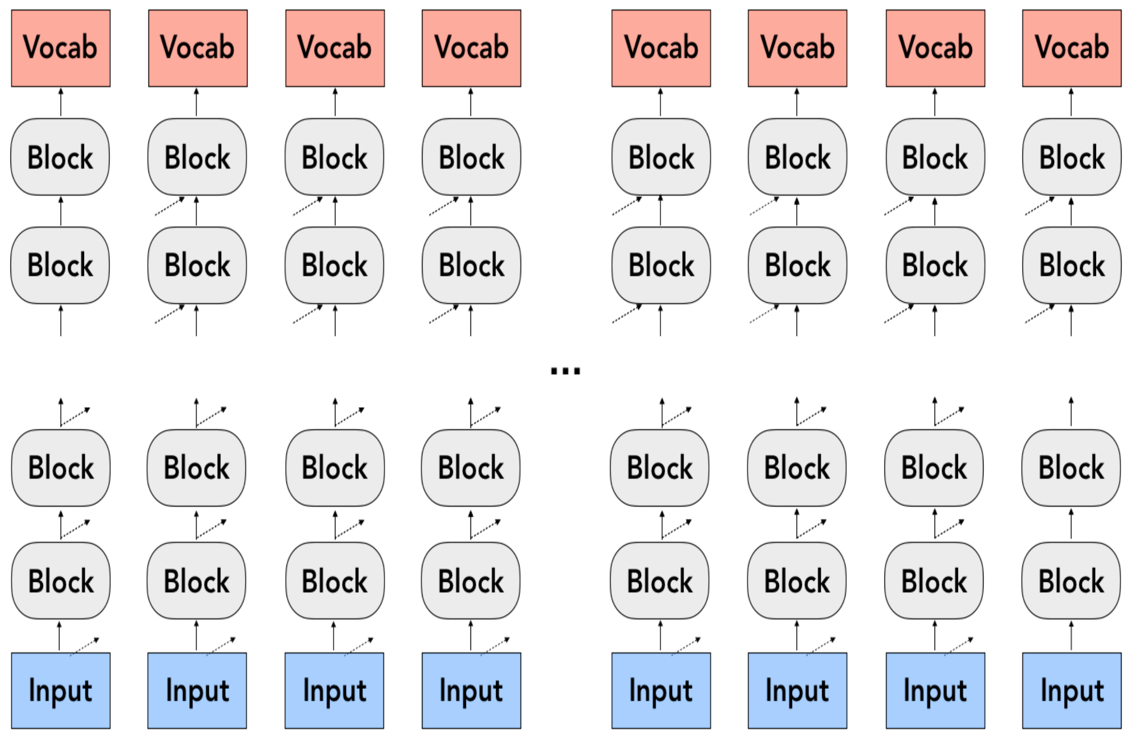
=



COMET

Generating Commonsense Knowledge

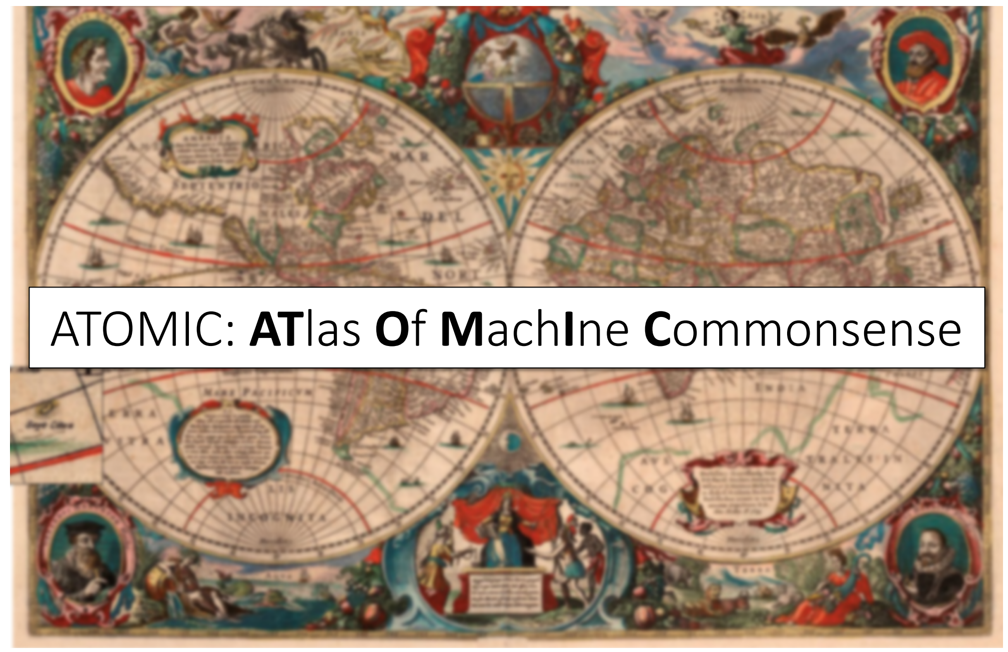
- Evaluate approach on two knowledge graphs: **ConceptNet** and **ATOMIC**



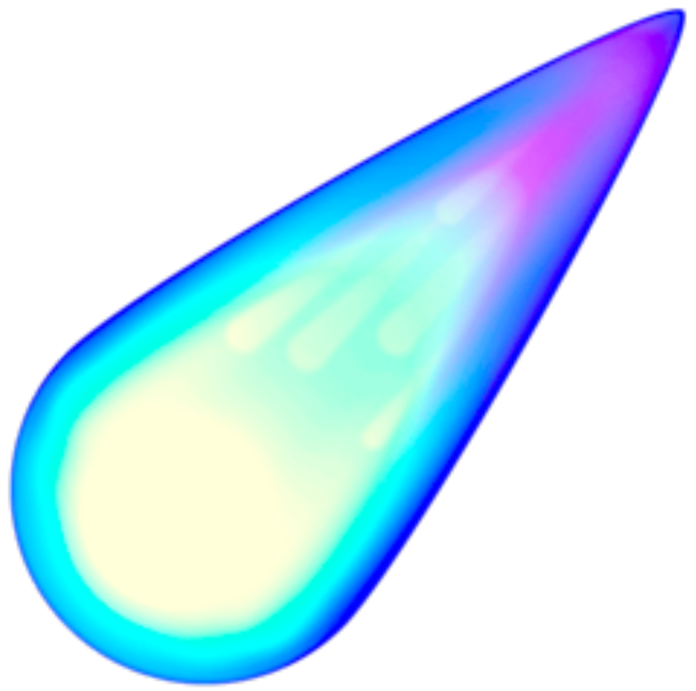
Transformer
Language Model

+

ConceptNet
An open, multilingual knowledge graph



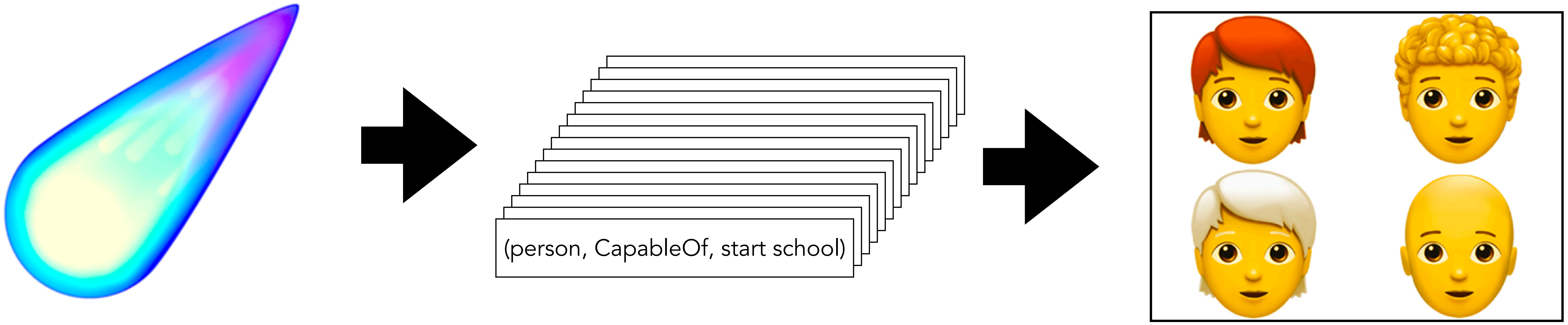
=



COMET

Generating Commonsense Knowledge

- Evaluate approach on two knowledge graphs: **ConceptNet** and **ATOMIC**
- Evaluate correctness of generated knowledge with human workers



COMET | Generate Knowledge | Evaluate by Humans

How well does COMET generate correct commonsense knowledge?

(Does it work?)

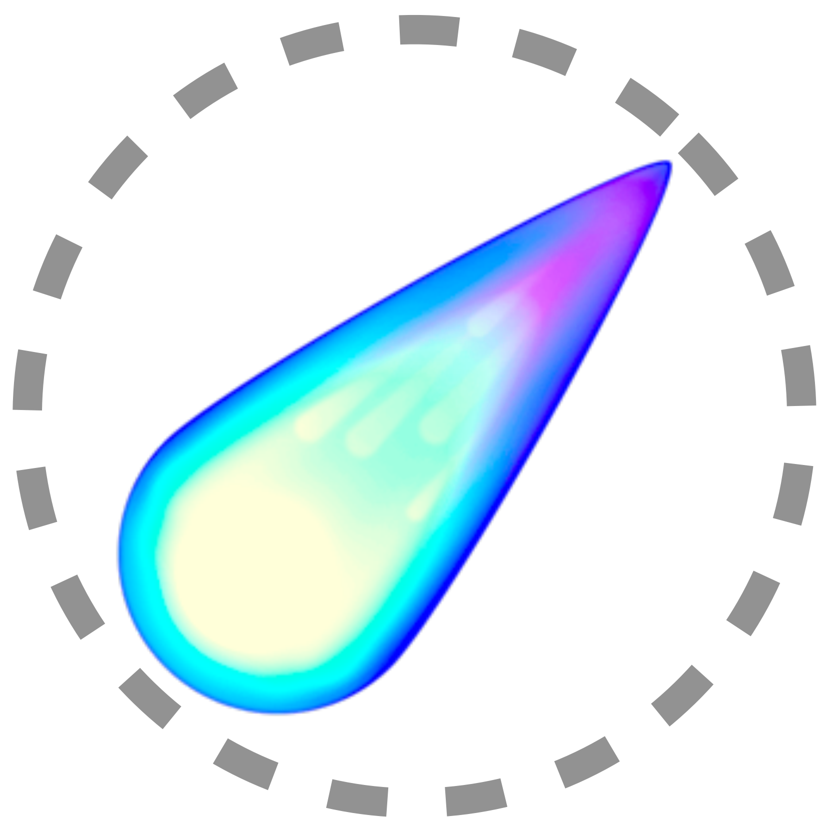
How well does COMET generate correct commonsense knowledge?

(Does it work?)

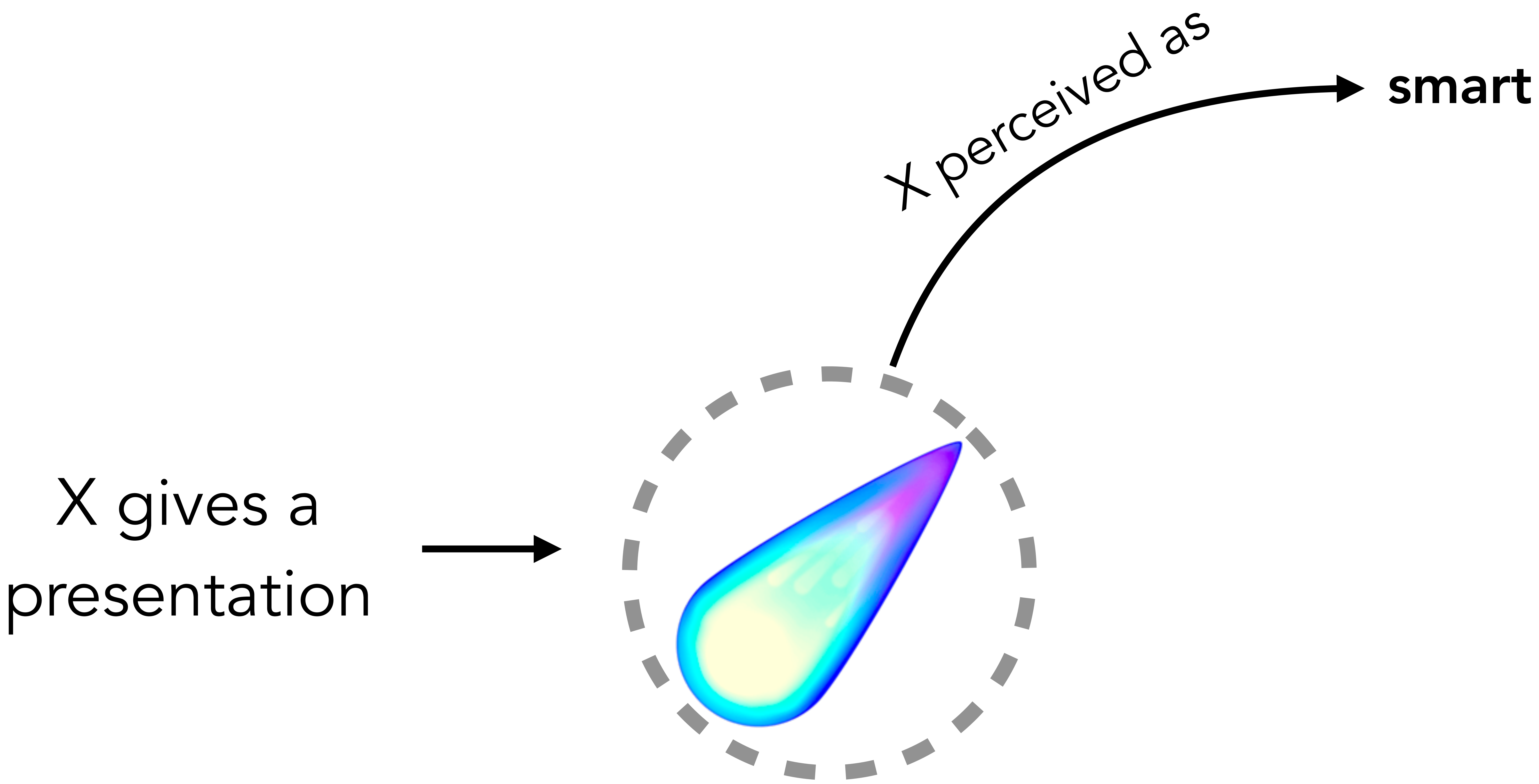
A: pretty well

ATOMIC Example

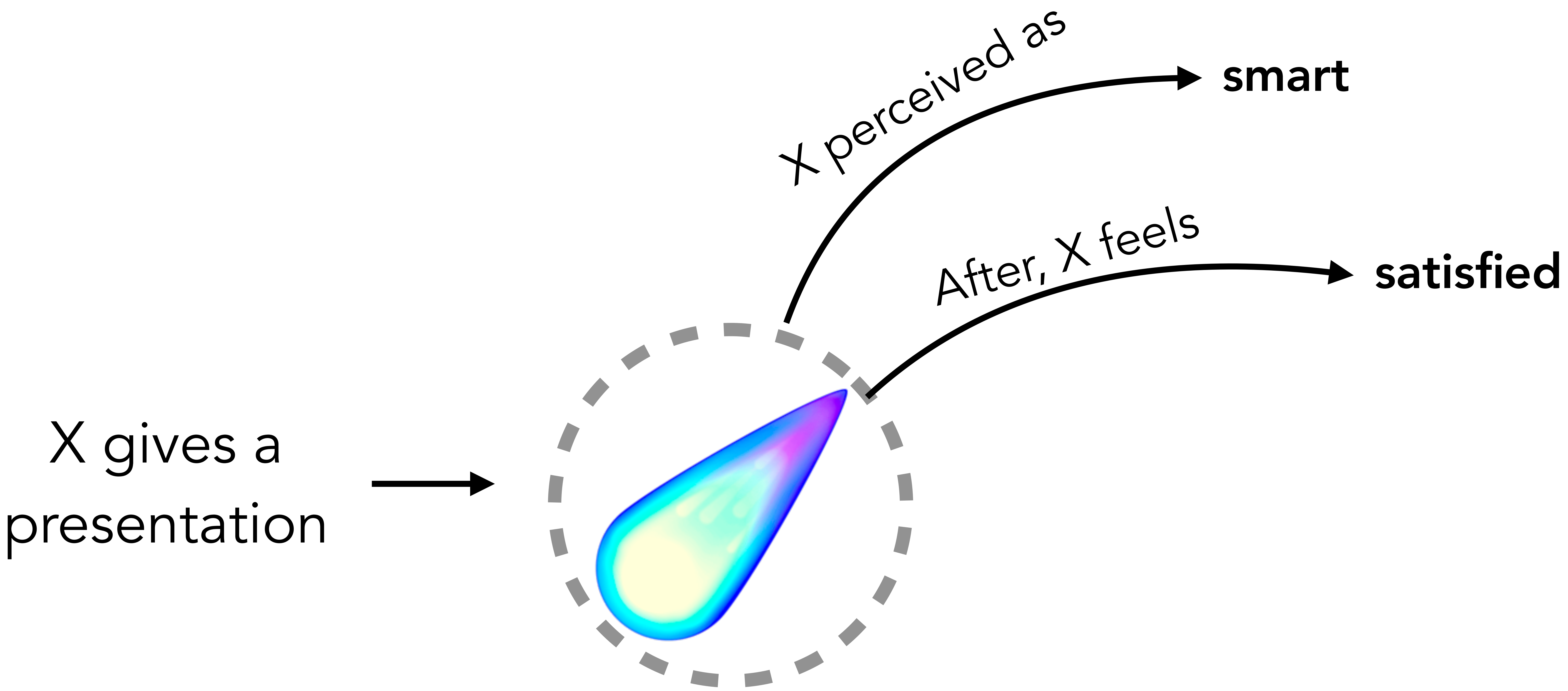
X gives a presentation



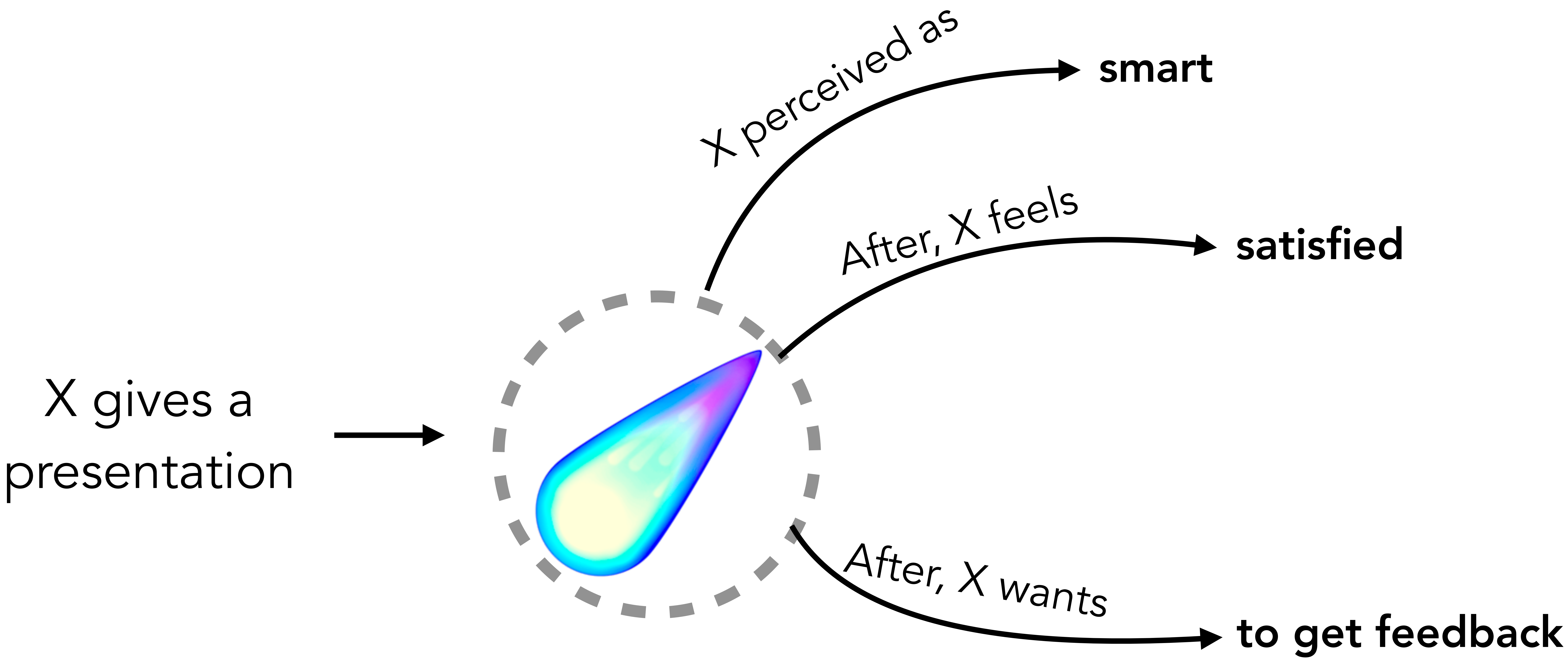
ATOMIC Example



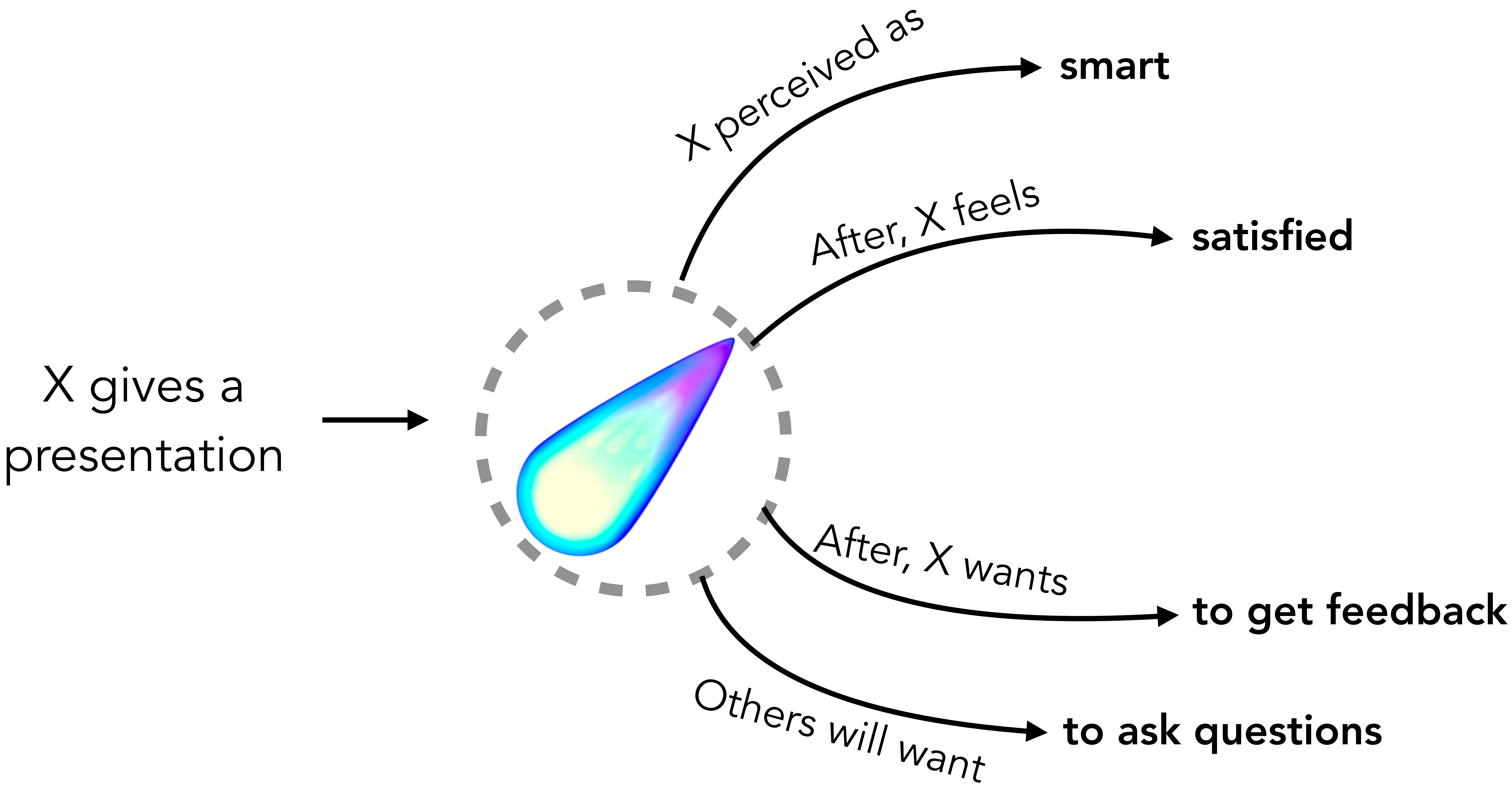
ATOMIC Example



ATOMIC Example

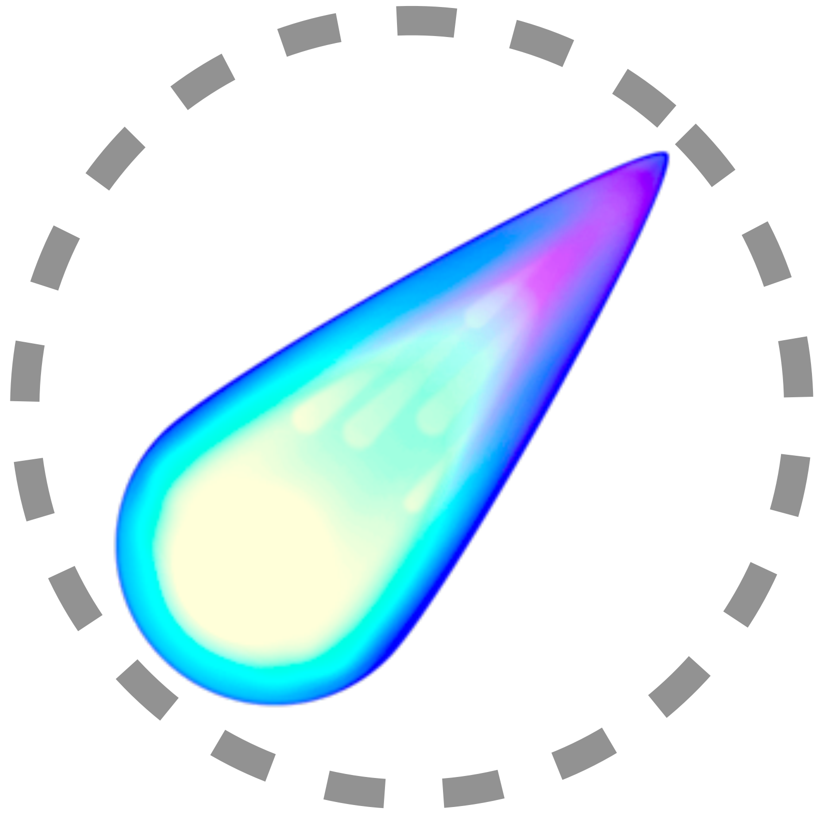


ATOMIC Example

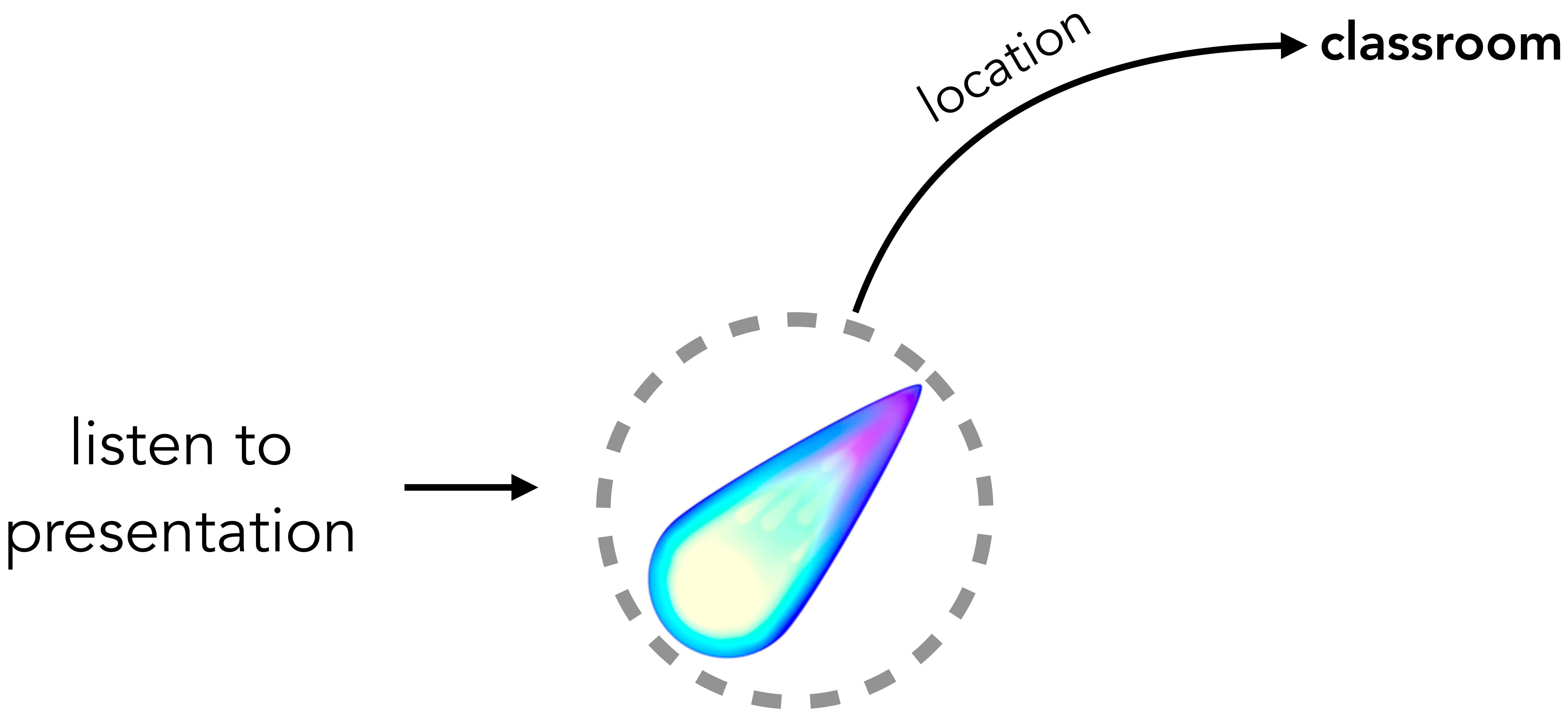


ConceptNet Example

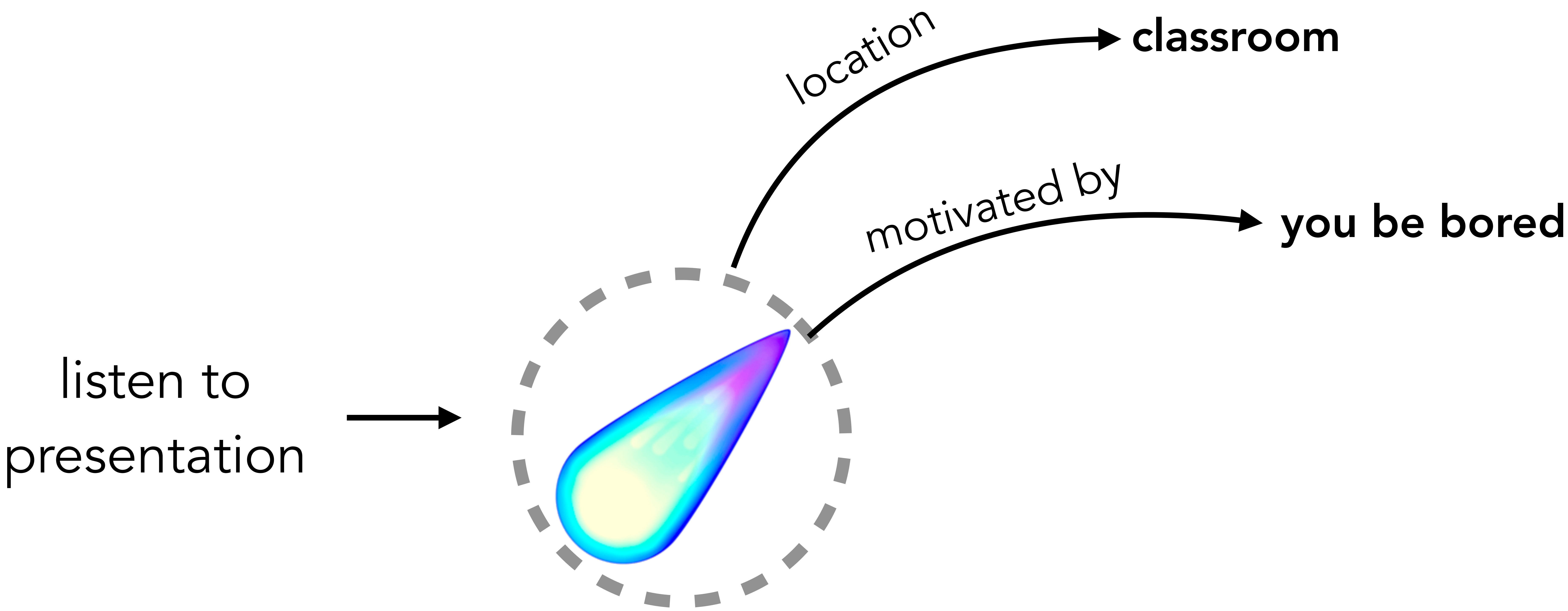
listen to presentation



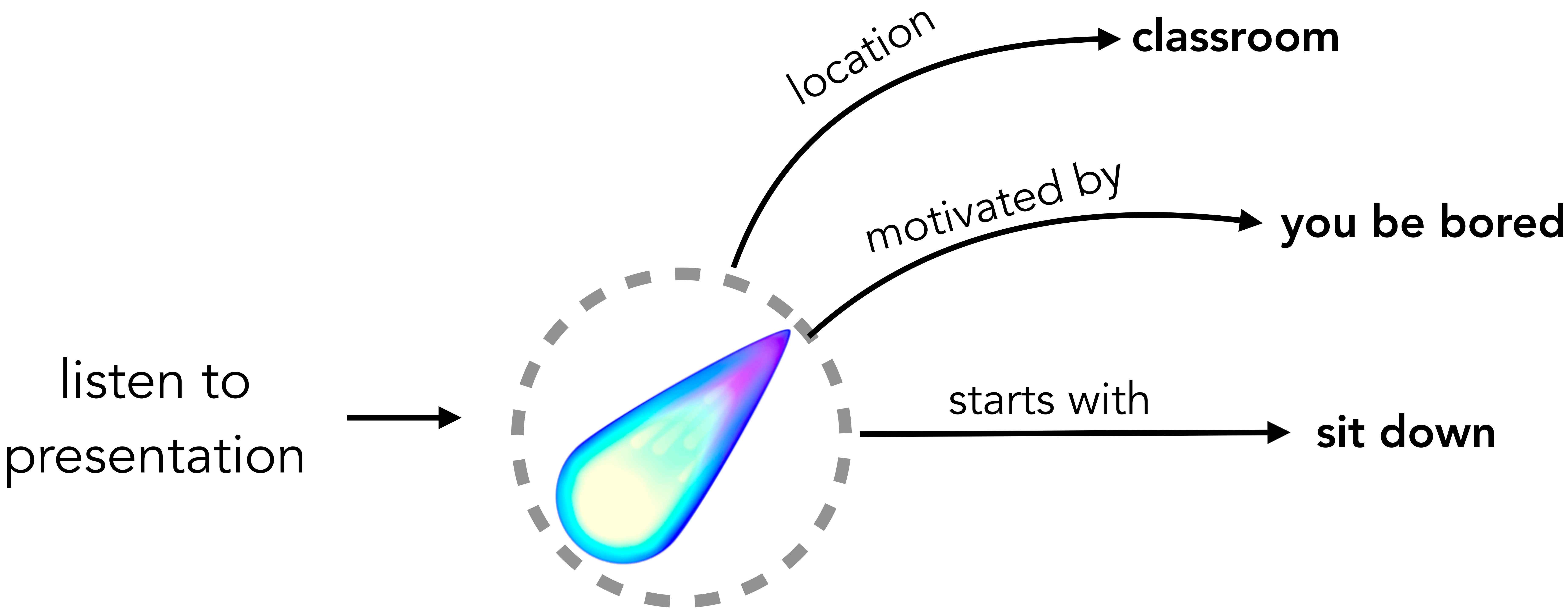
ConceptNet Example



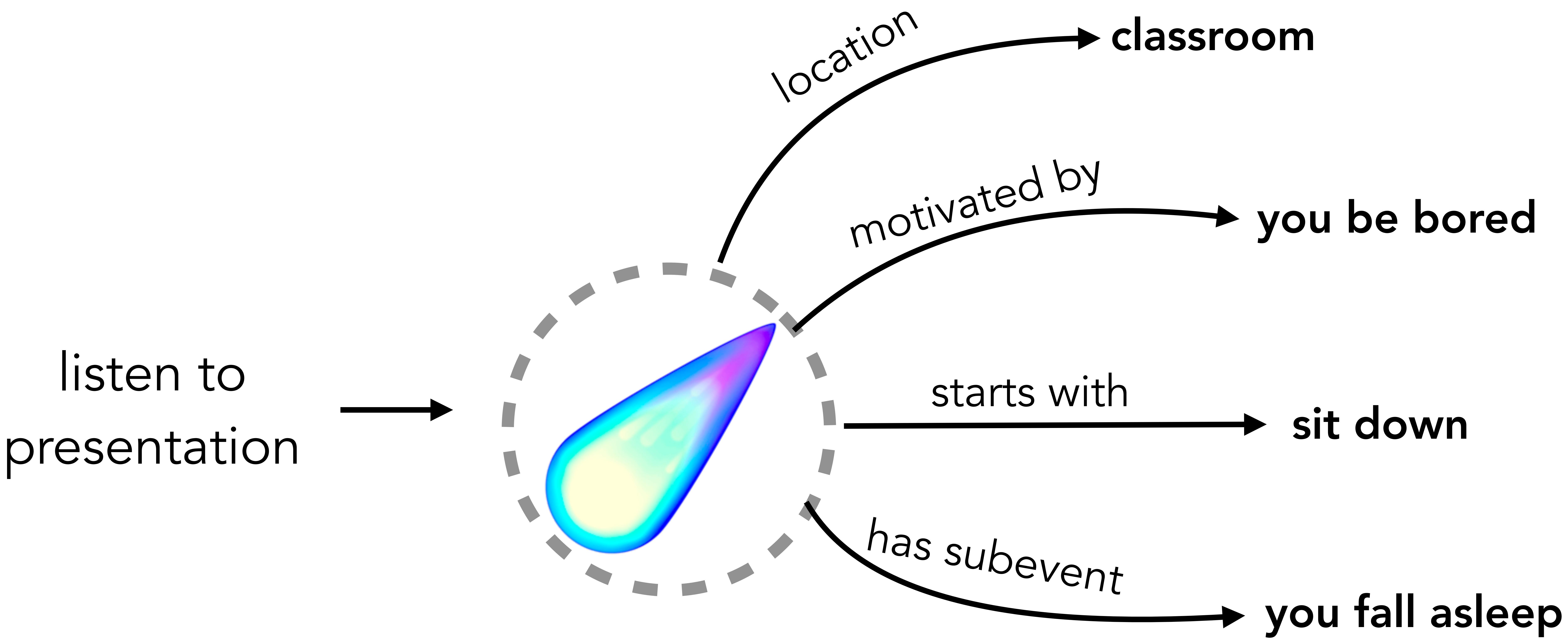
ConceptNet Example



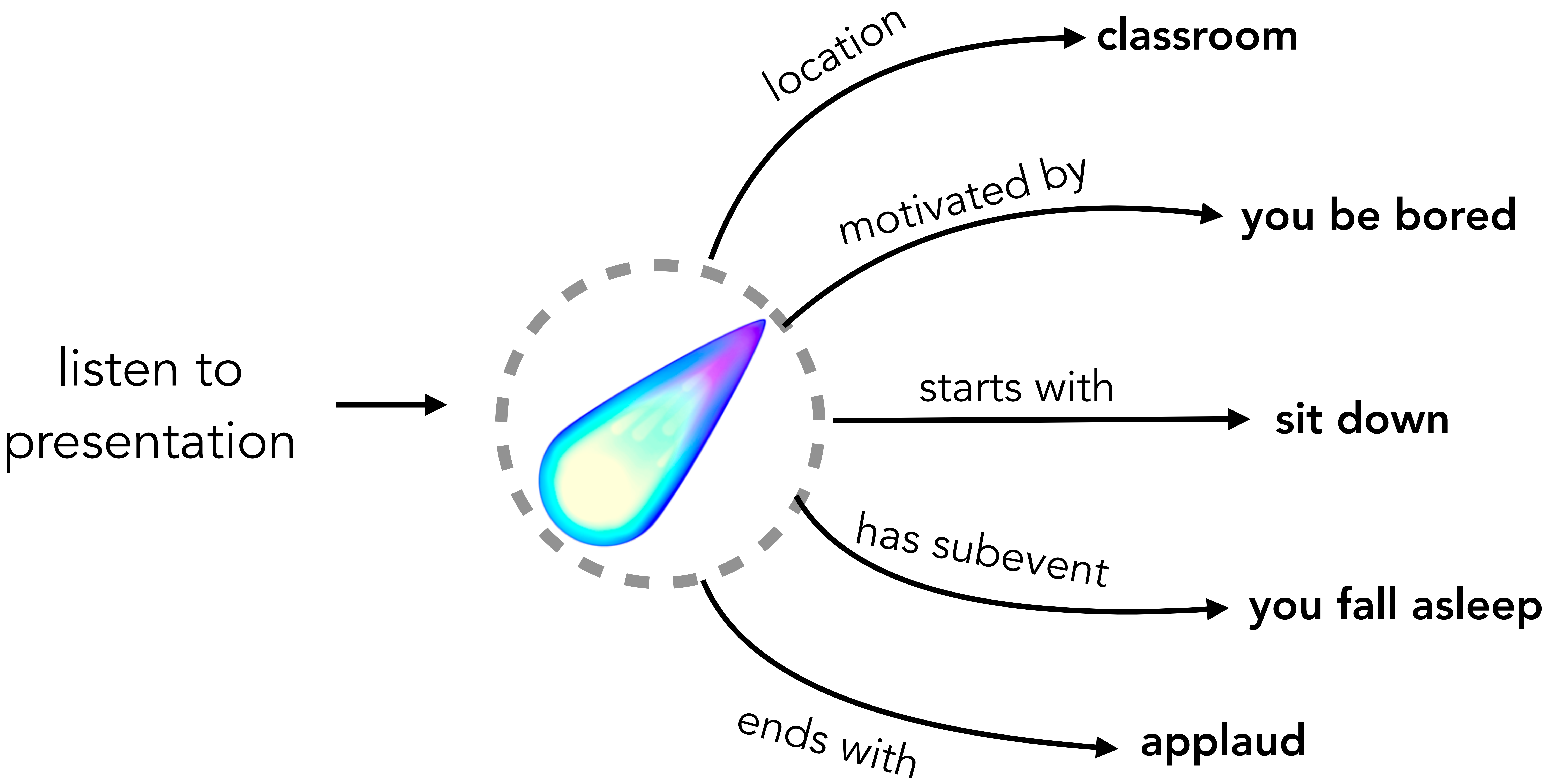
ConceptNet Example



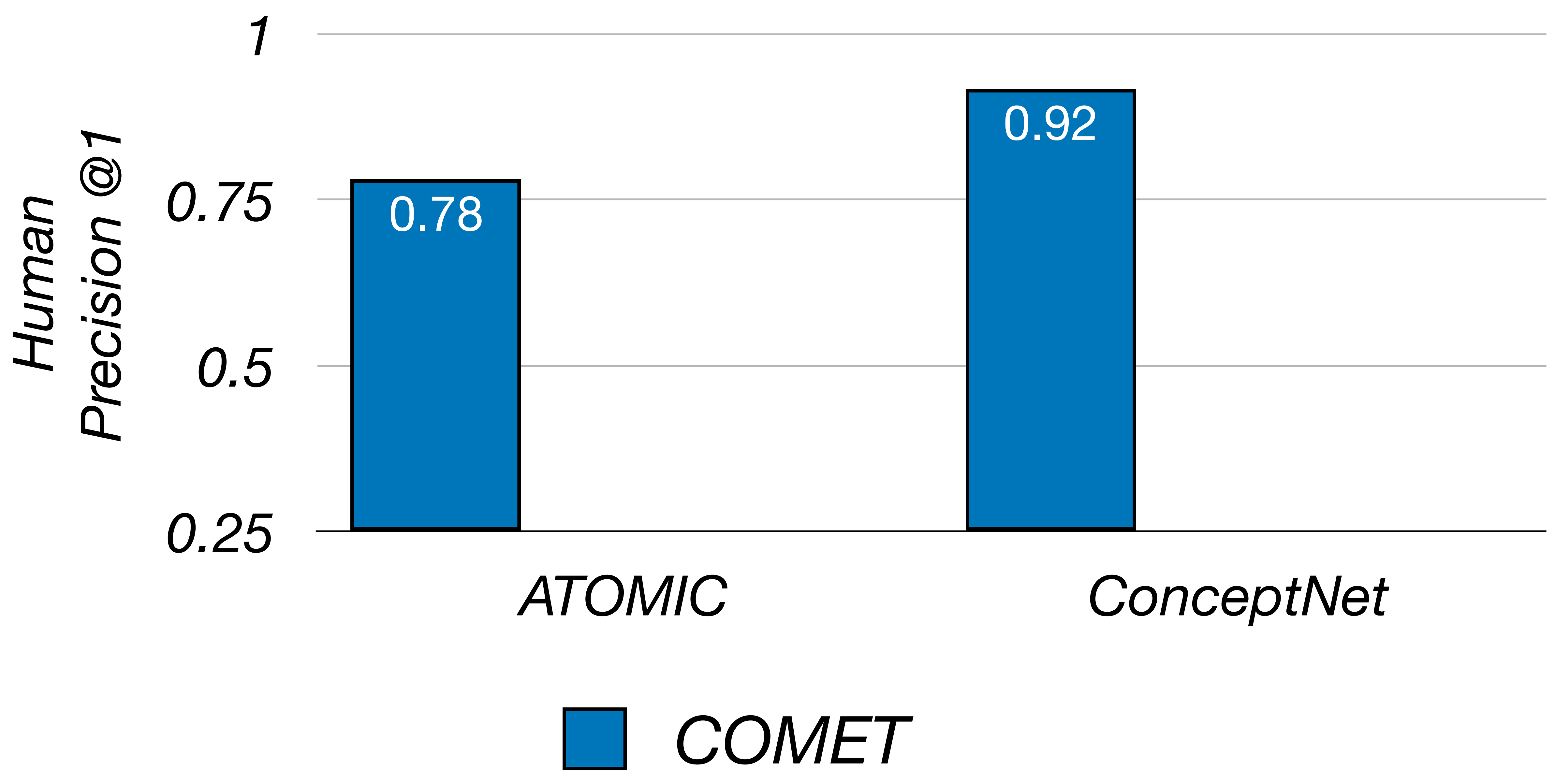
ConceptNet Example



ConceptNet Example

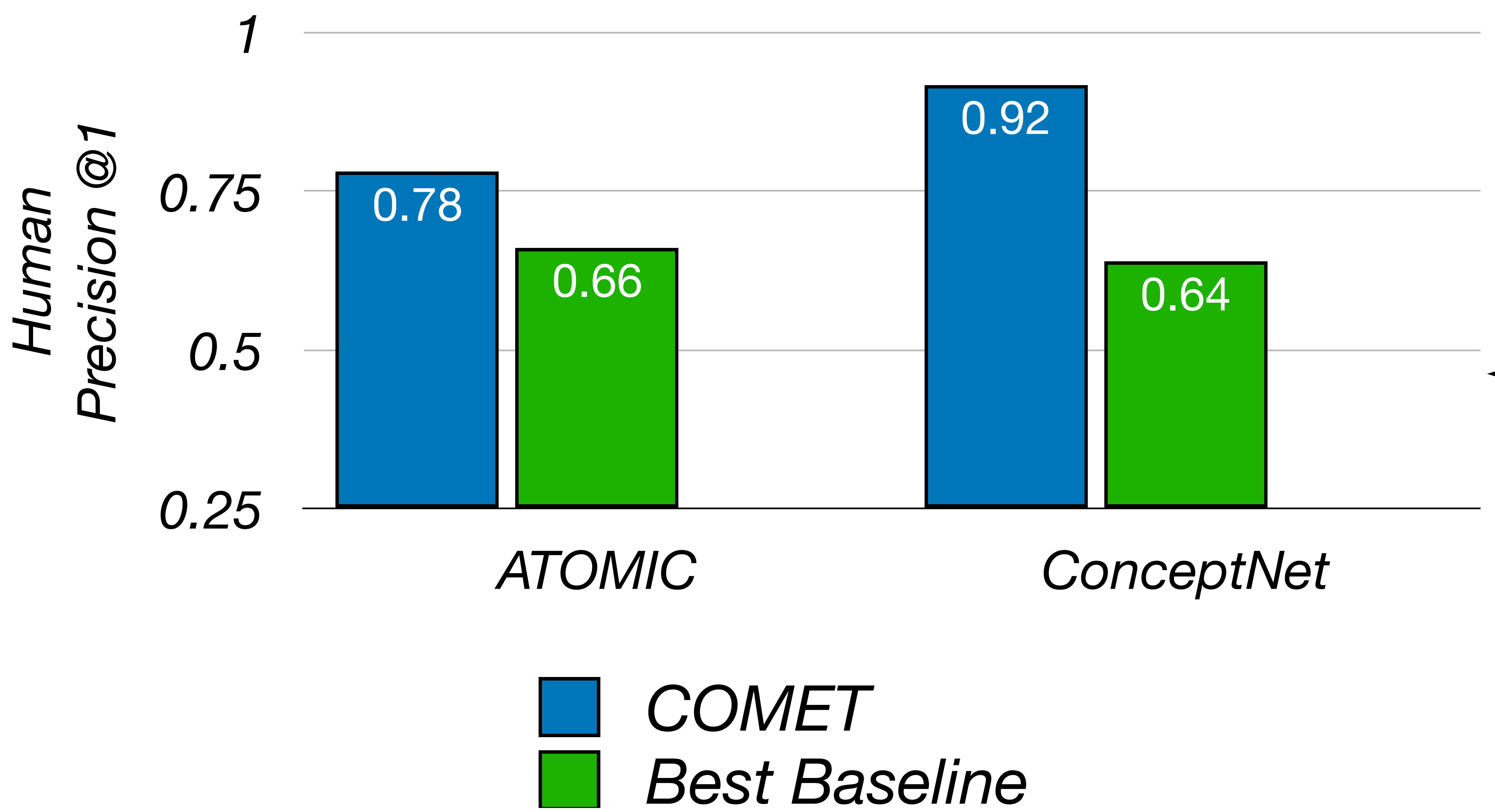


Generated Knowledge Quality



Good quality of knowledge

Generated Knowledge Quality



Good quality of knowledge

Significant improvements over baselines

Does COMET transfer unstructured knowledge learned implicitly from raw text?

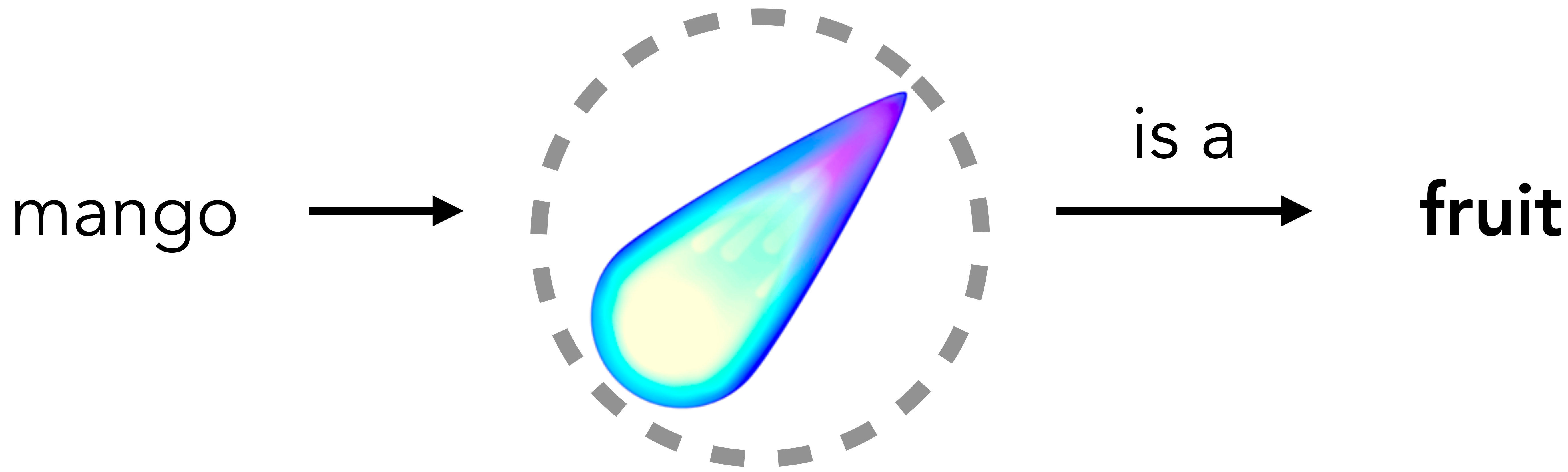
(Why does it work?)

Does COMET transfer unstructured knowledge learned implicitly from raw text?

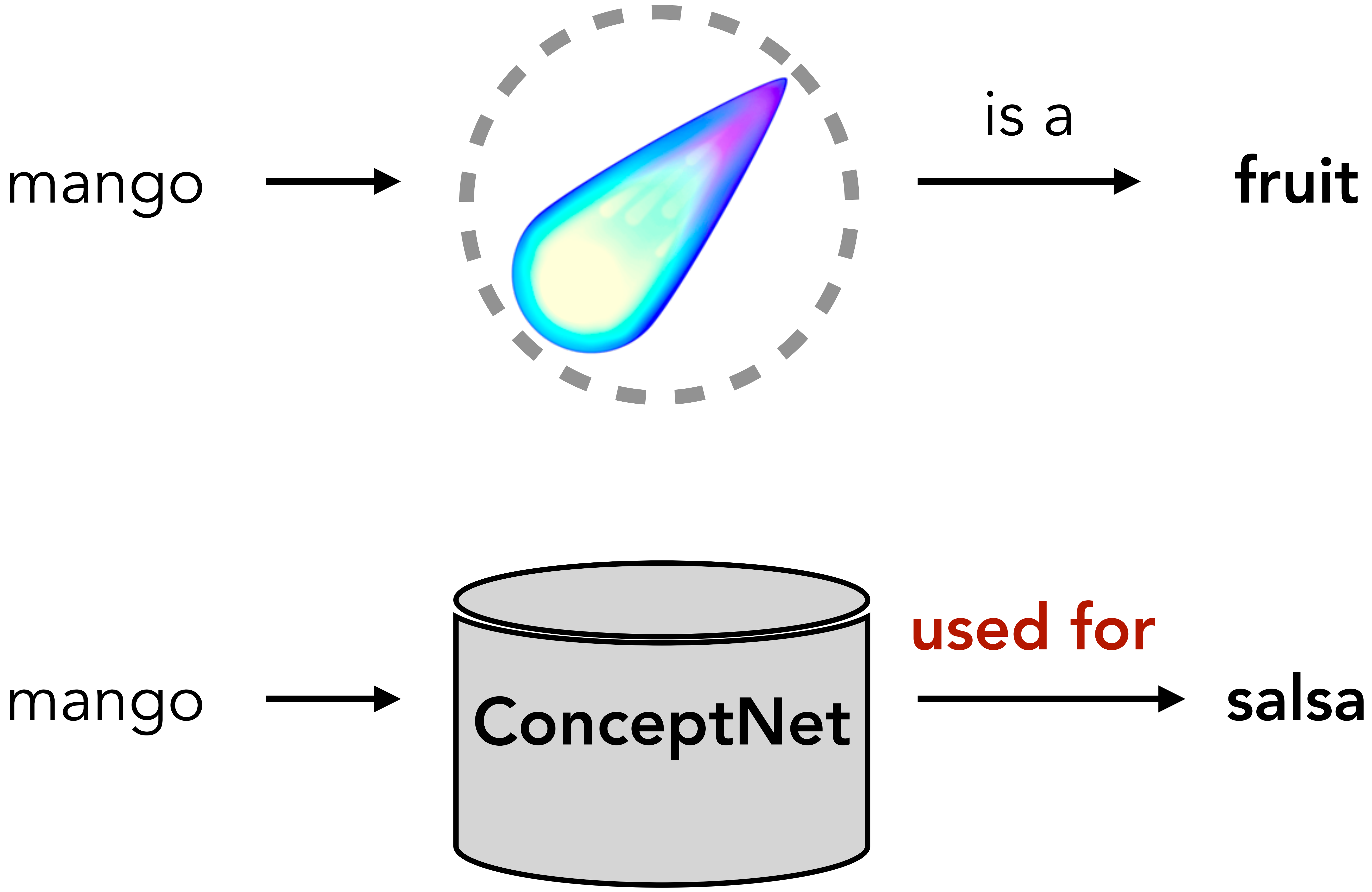
(Why does it work?)

A: language understanding transfers to knowledge understanding

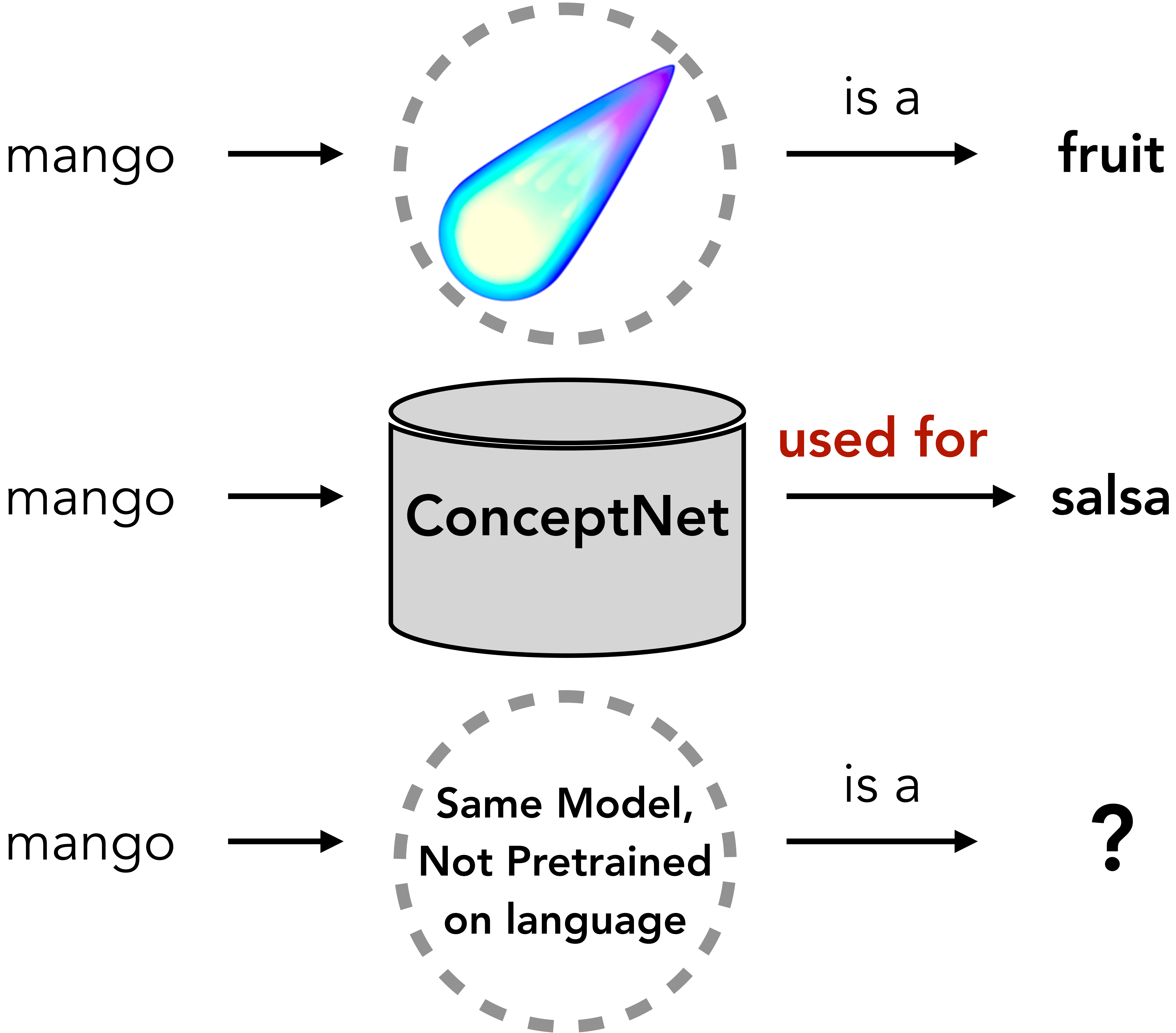
Transfer Learning from Language



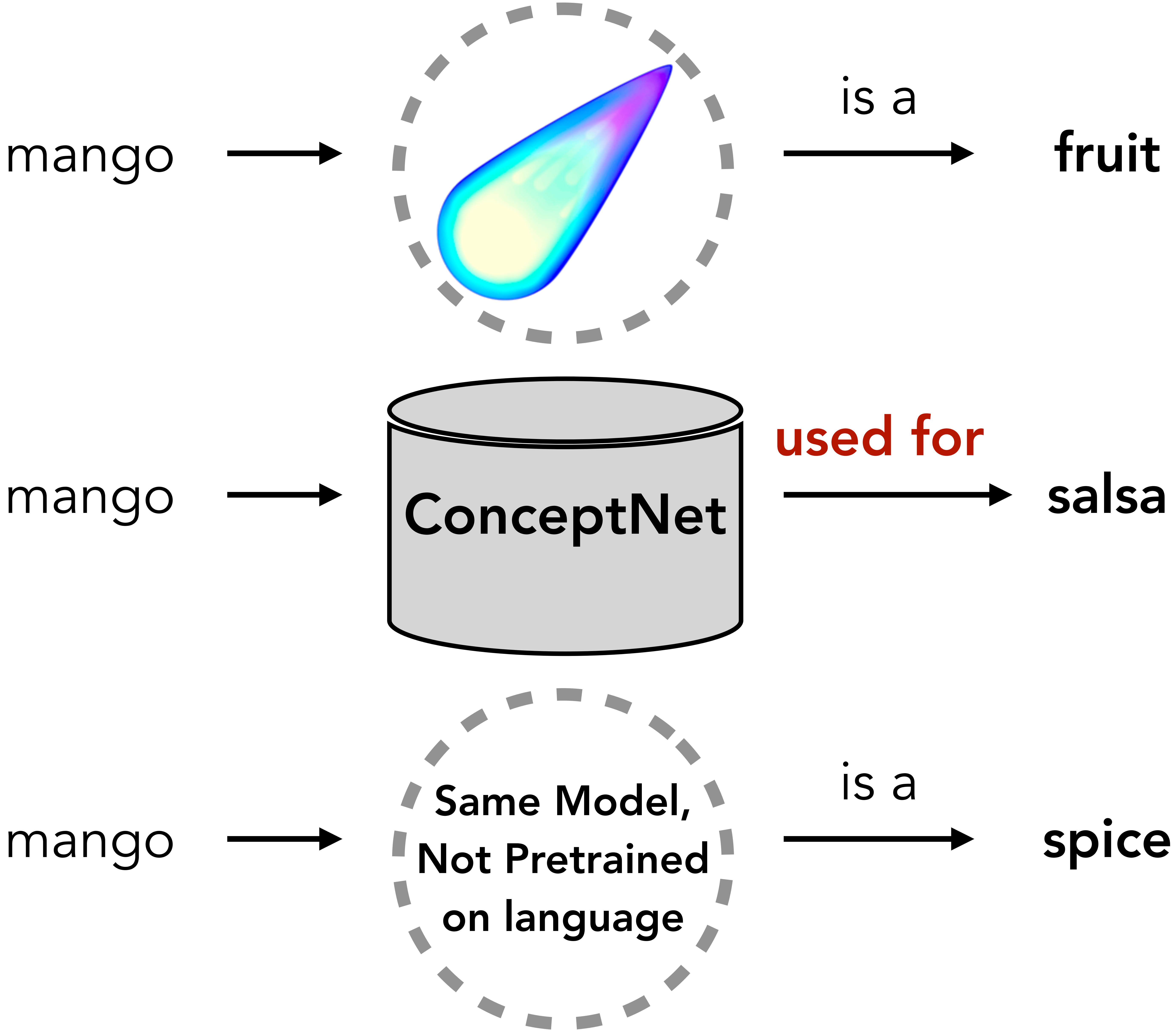
Transfer Learning from Language



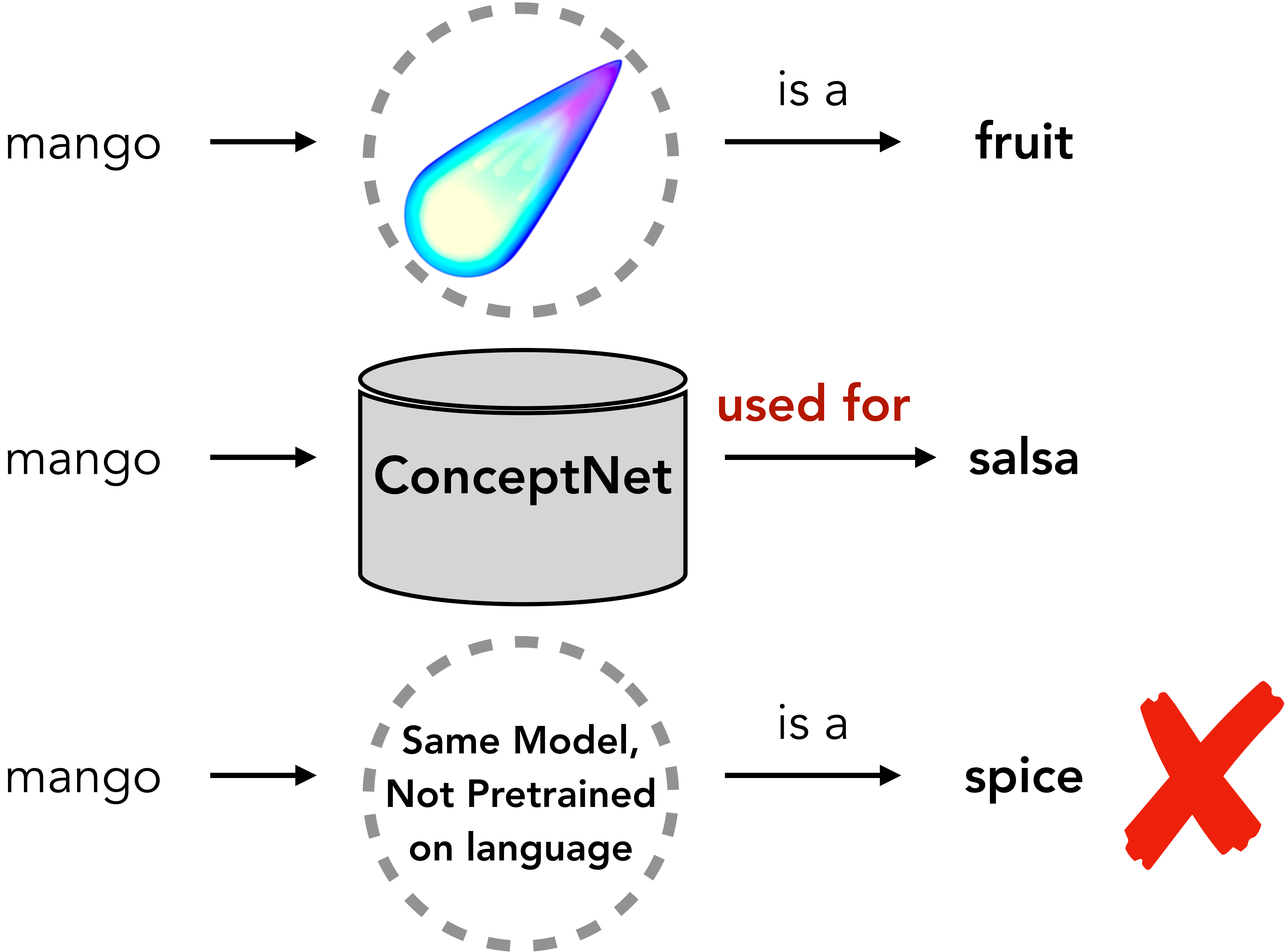
Transfer Learning from Language



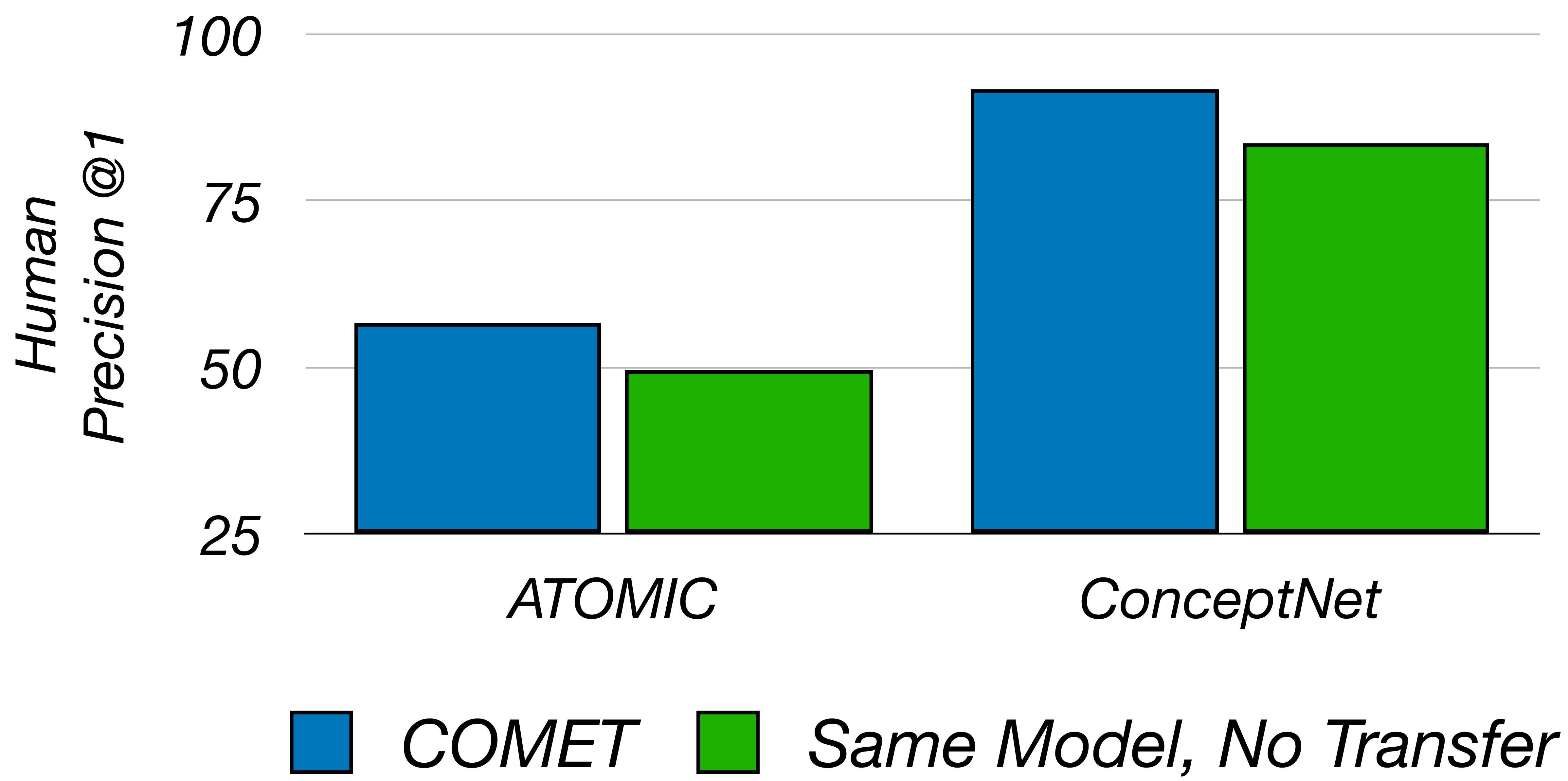
Transfer Learning from Language



Transfer Learning from Language



Transfer Learning from Language



Significant improvement over **same neural architecture** with **no transfer learning from language**

Does COMET generalize to examples that deviate from the format of the seed knowledge graph?

(Why is this important?)

Does COMET generalize to examples that deviate from the format of the seed knowledge graph?

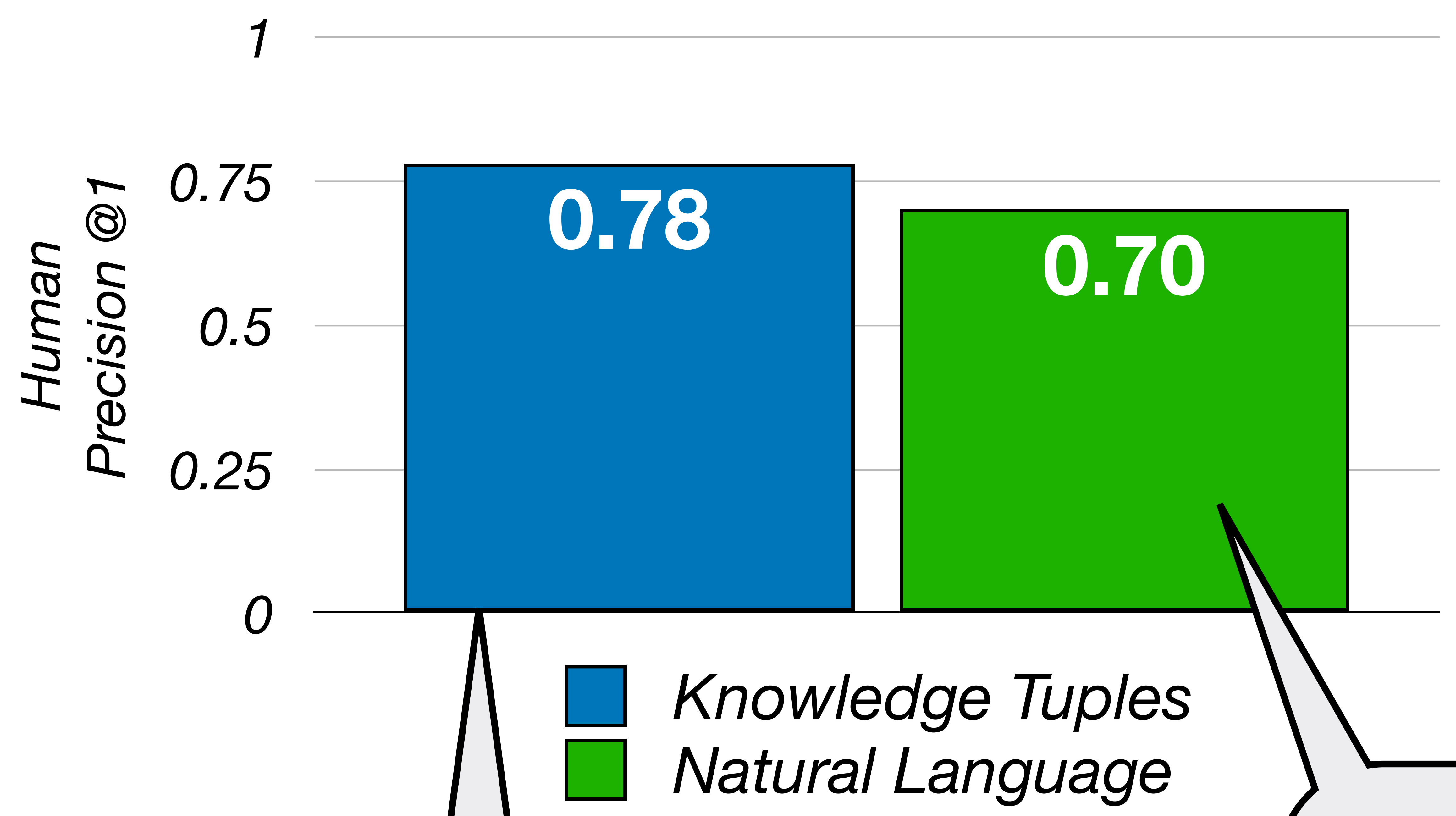
(Why is this important?)

A: COMET can generate knowledge for complex natural language inputs

Demo

mosaickg.apps.allenai.org

Common sense for any situation!



Examples that **look like knowledge graph training examples**

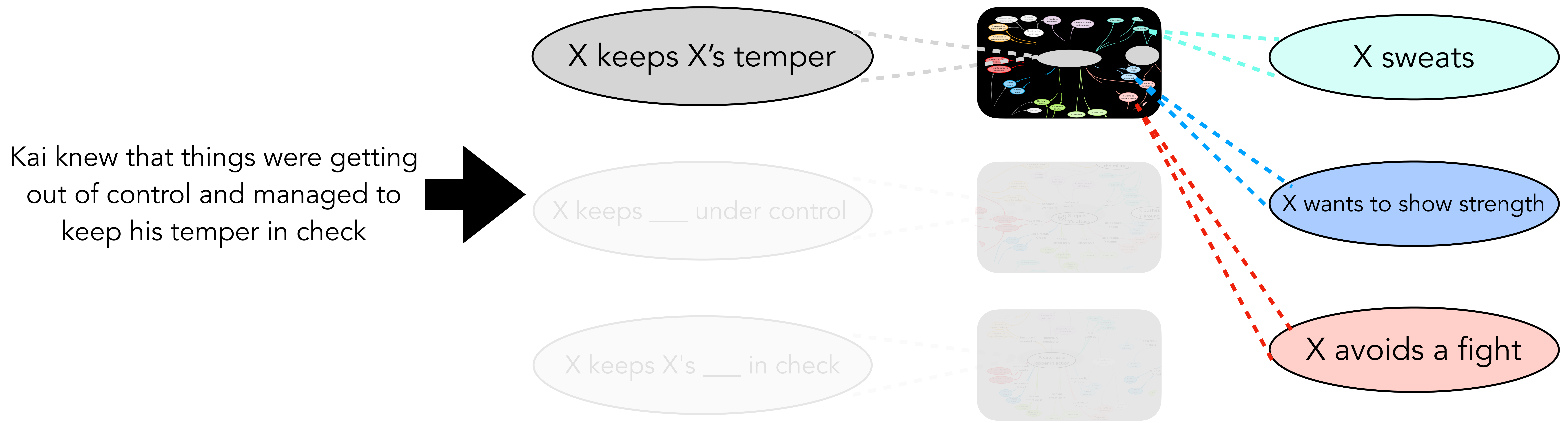
Examples that **look like natural language**

Example Context:

Kai knew that things were getting out of control and managed to keep his temper in check.

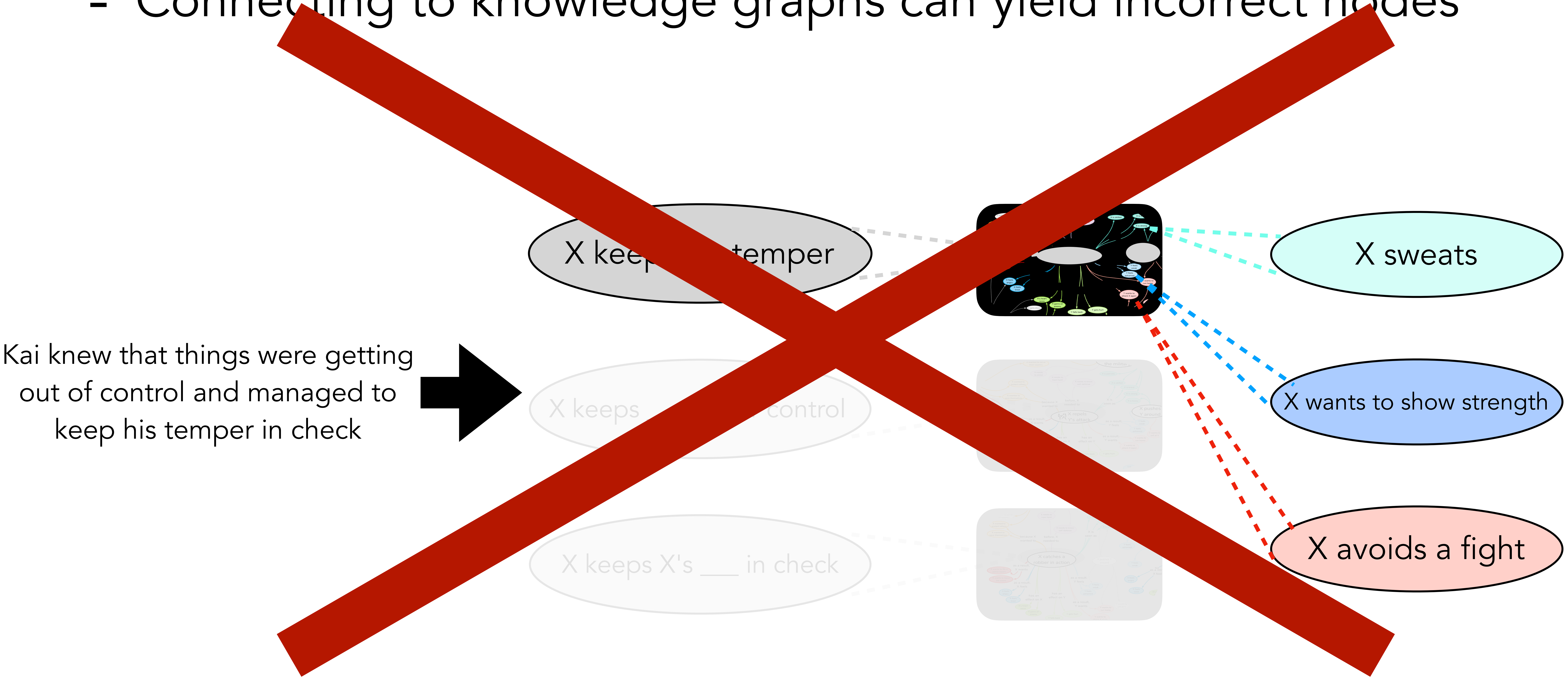
Reasoning with Knowledge Graphs

- Situations rarely found **as-is** in commonsense knowledge graphs
- Connecting to knowledge graphs can yield incorrect nodes



Reasoning with Knowledge Graphs

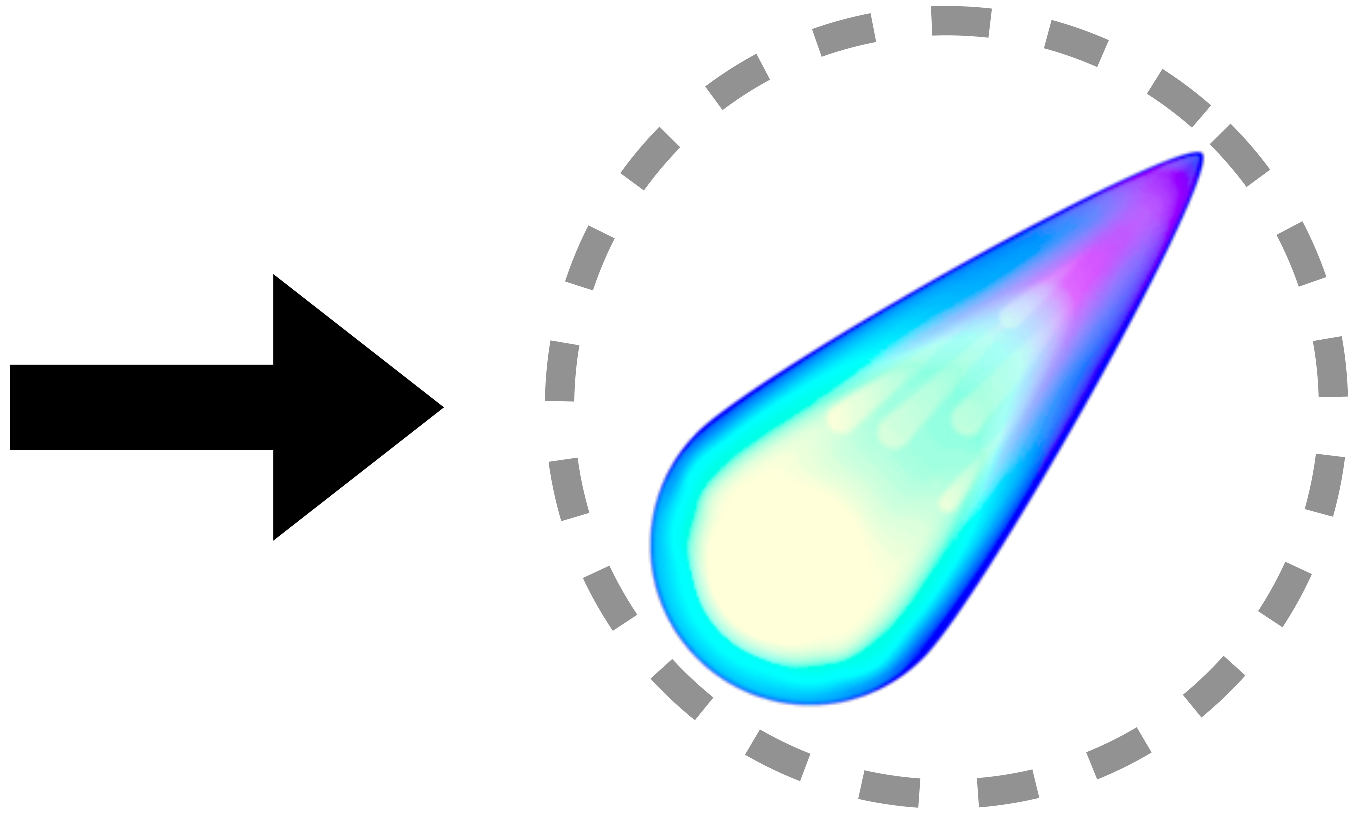
- Situations rarely found **as-is** in commonsense knowledge graphs
- Connecting to knowledge graphs can yield incorrect nodes



Common Sense for Arbitrary Situations

- transformer-style architecture — input format is natural language
  - event can be fully parsed

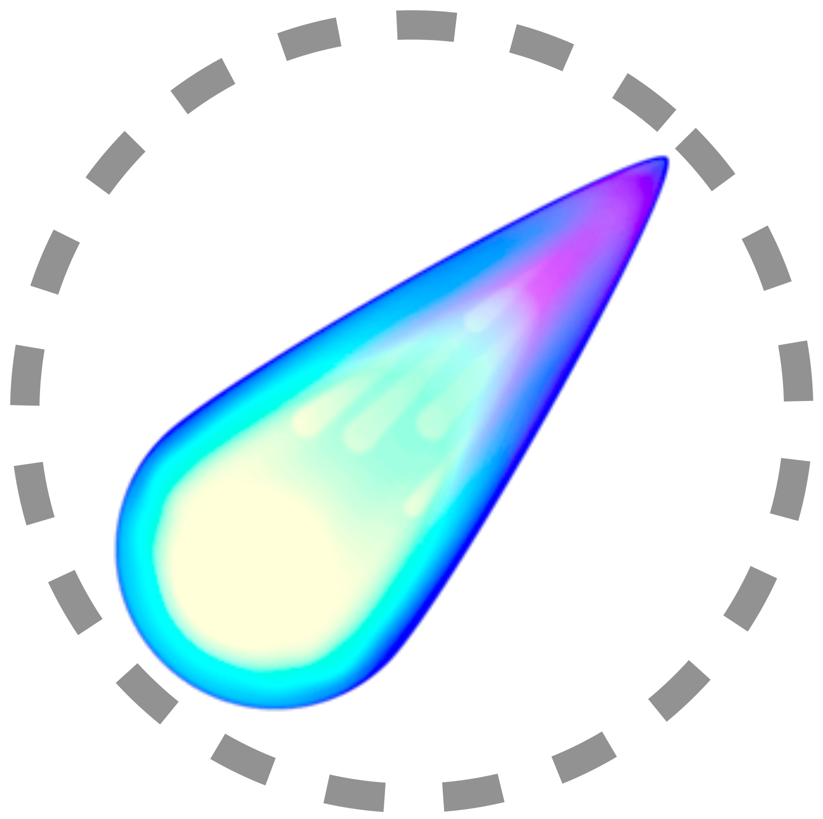
Kai knew that things were getting out of control and managed to keep his temper in check



Common Sense for Arbitrary Situations

- transformer-style architecture — input format is natural language
  - event can be fully parsed
  - knowledge generated **dynamically** from **neural** knowledge model

Kai knew that things were getting out of control and managed to keep his temper in check



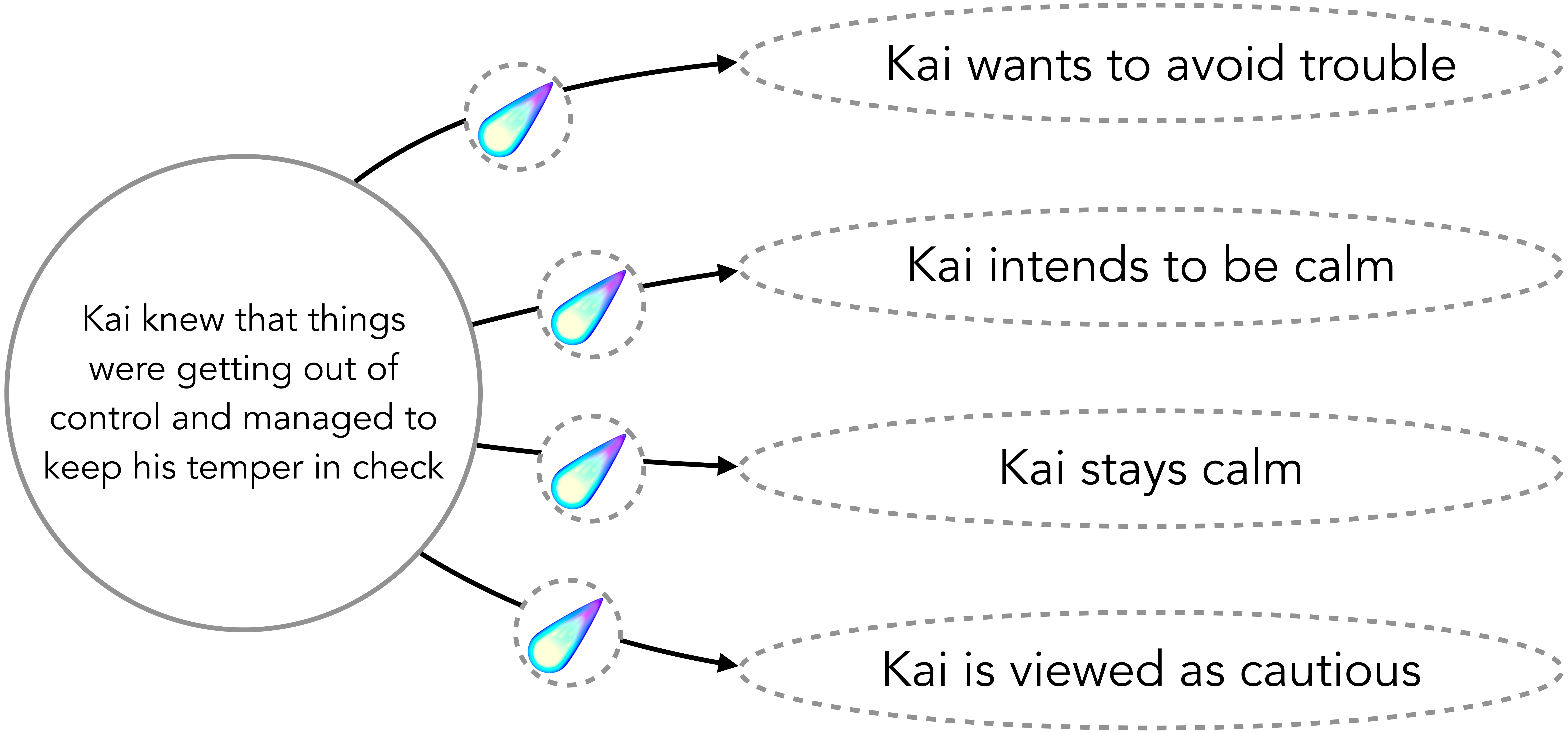
Kai wants to avoid trouble

Kai intends to be calm

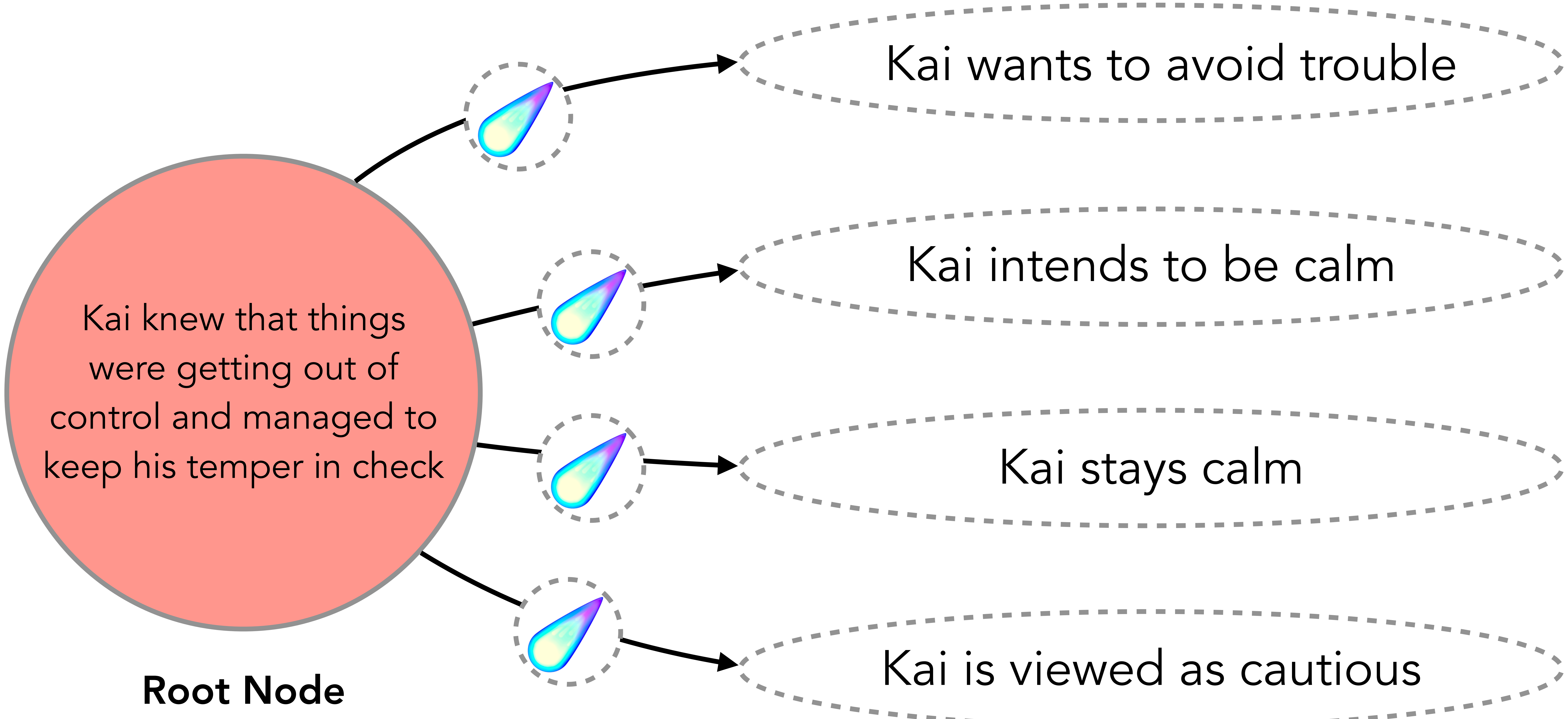
Kai stays calm

Kai is viewed as cautious

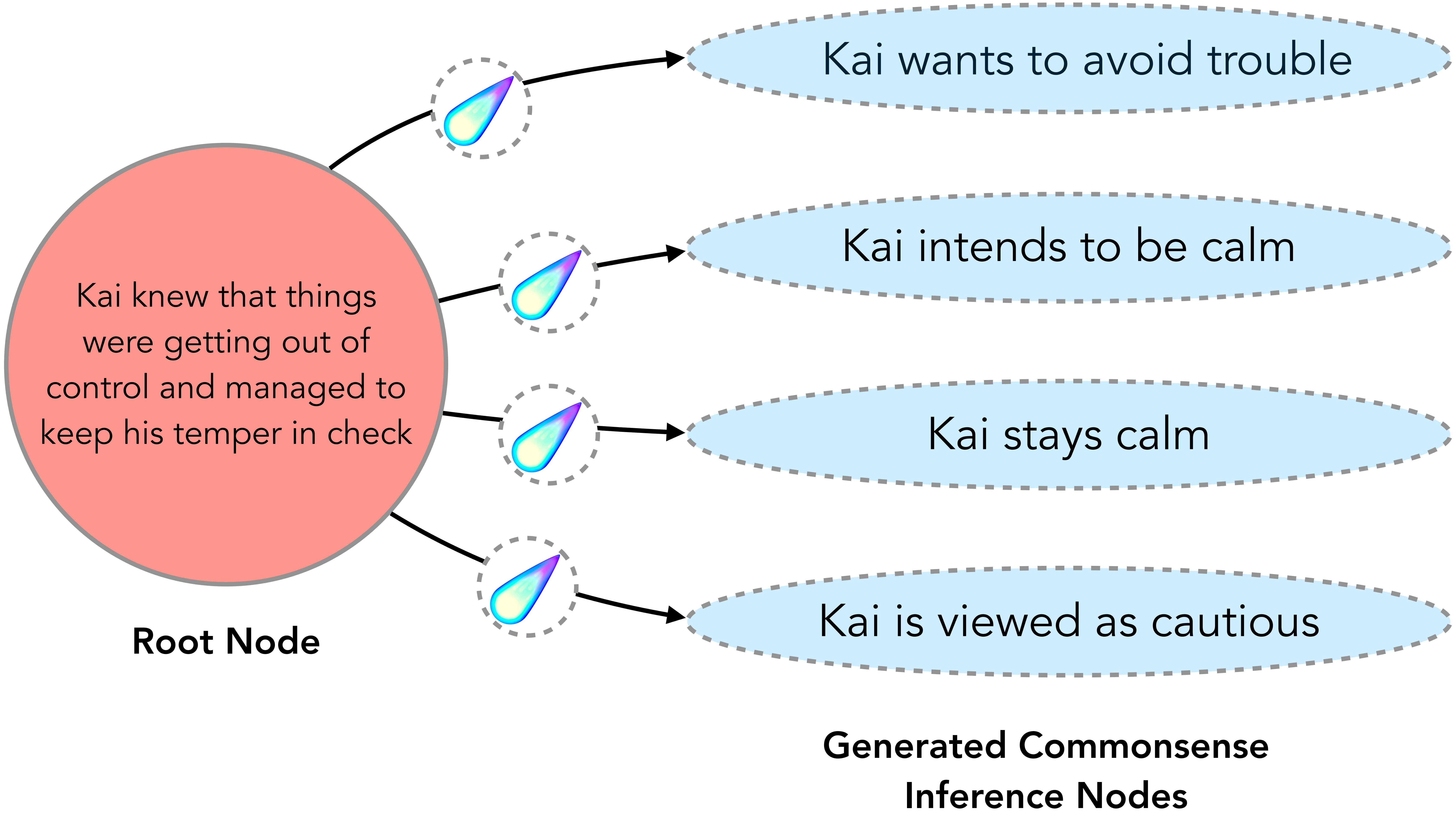
Dynamic Construction of Reasoning Graphs

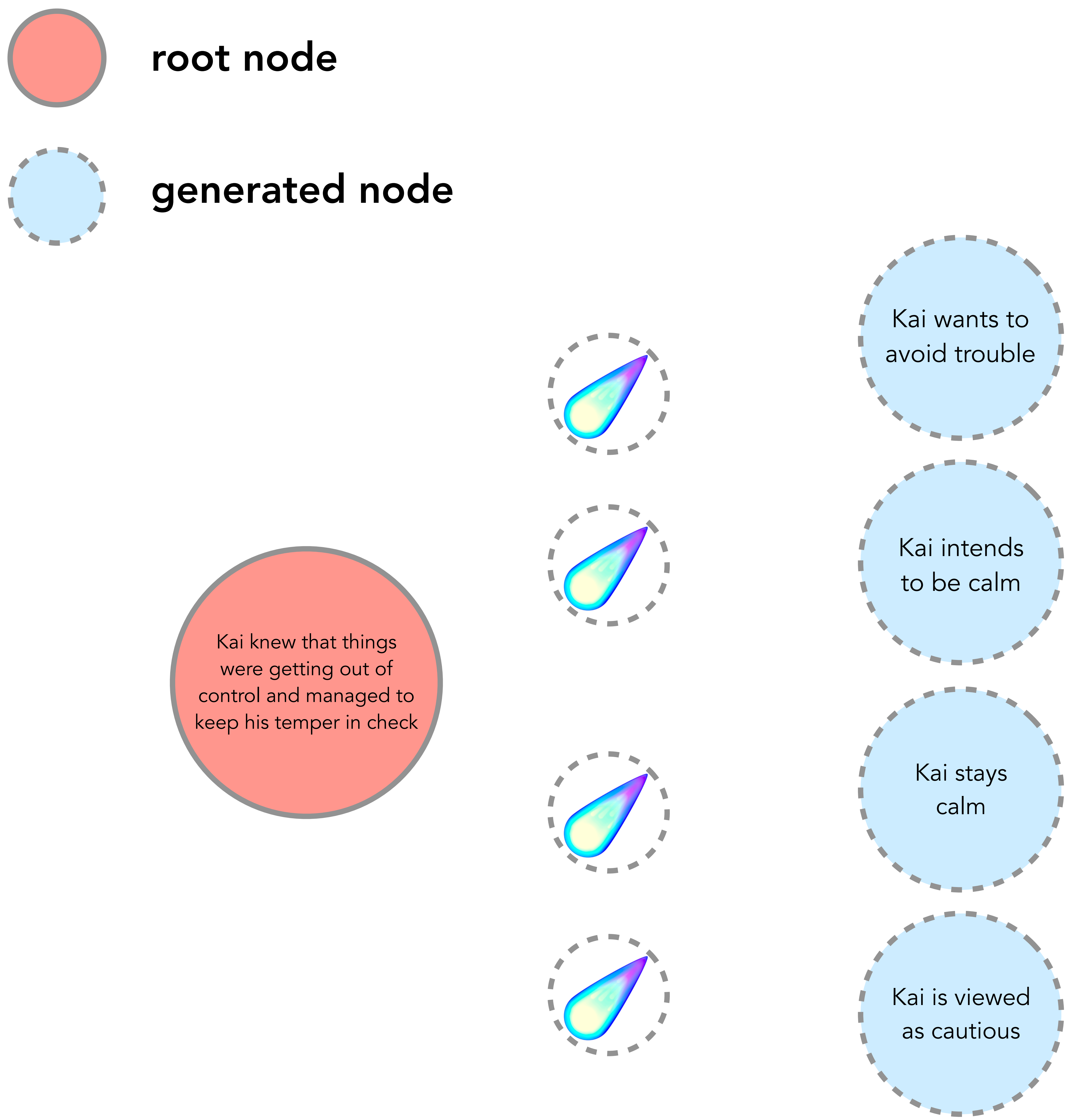


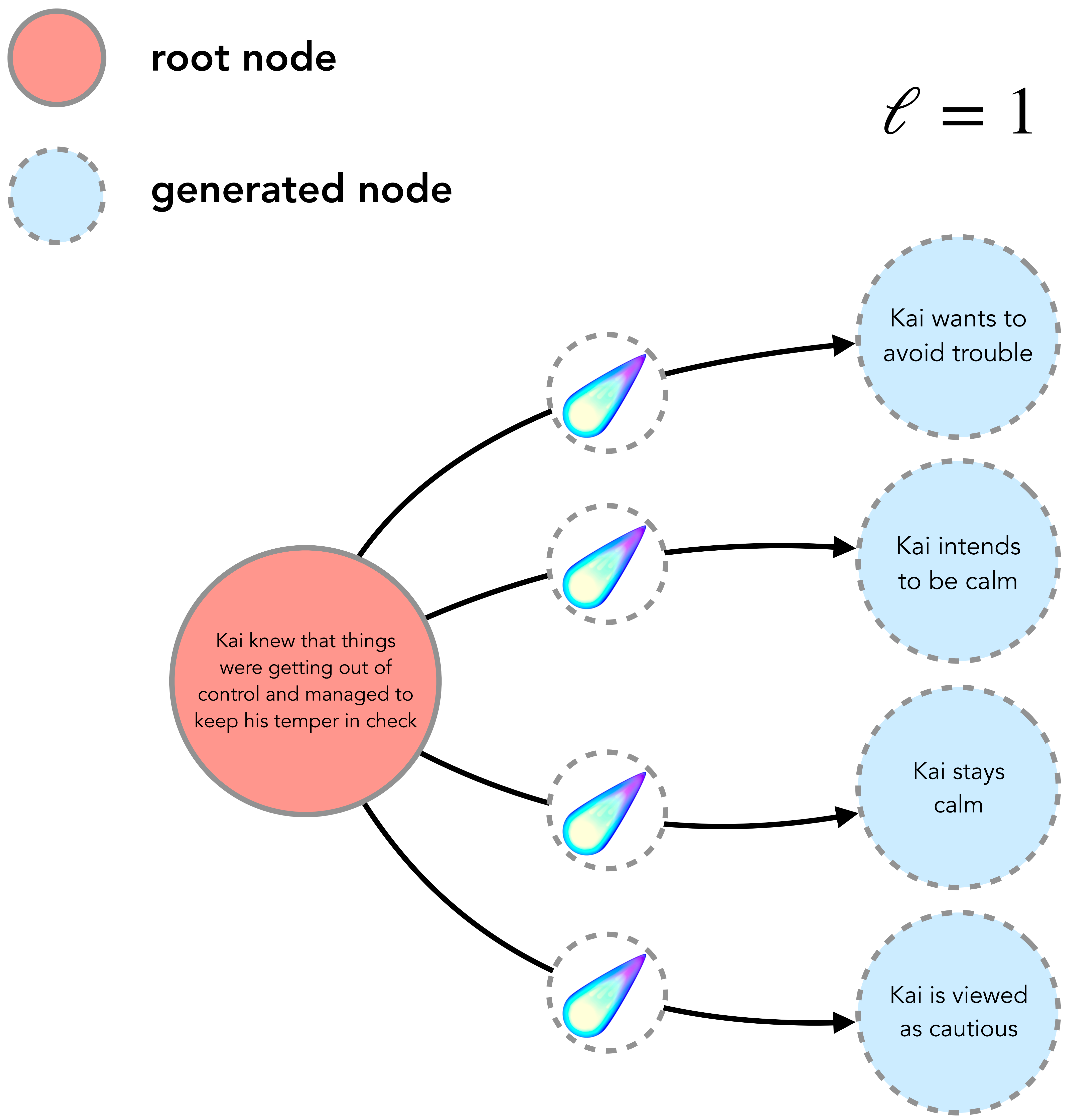
Dynamic Construction of Reasoning Graphs

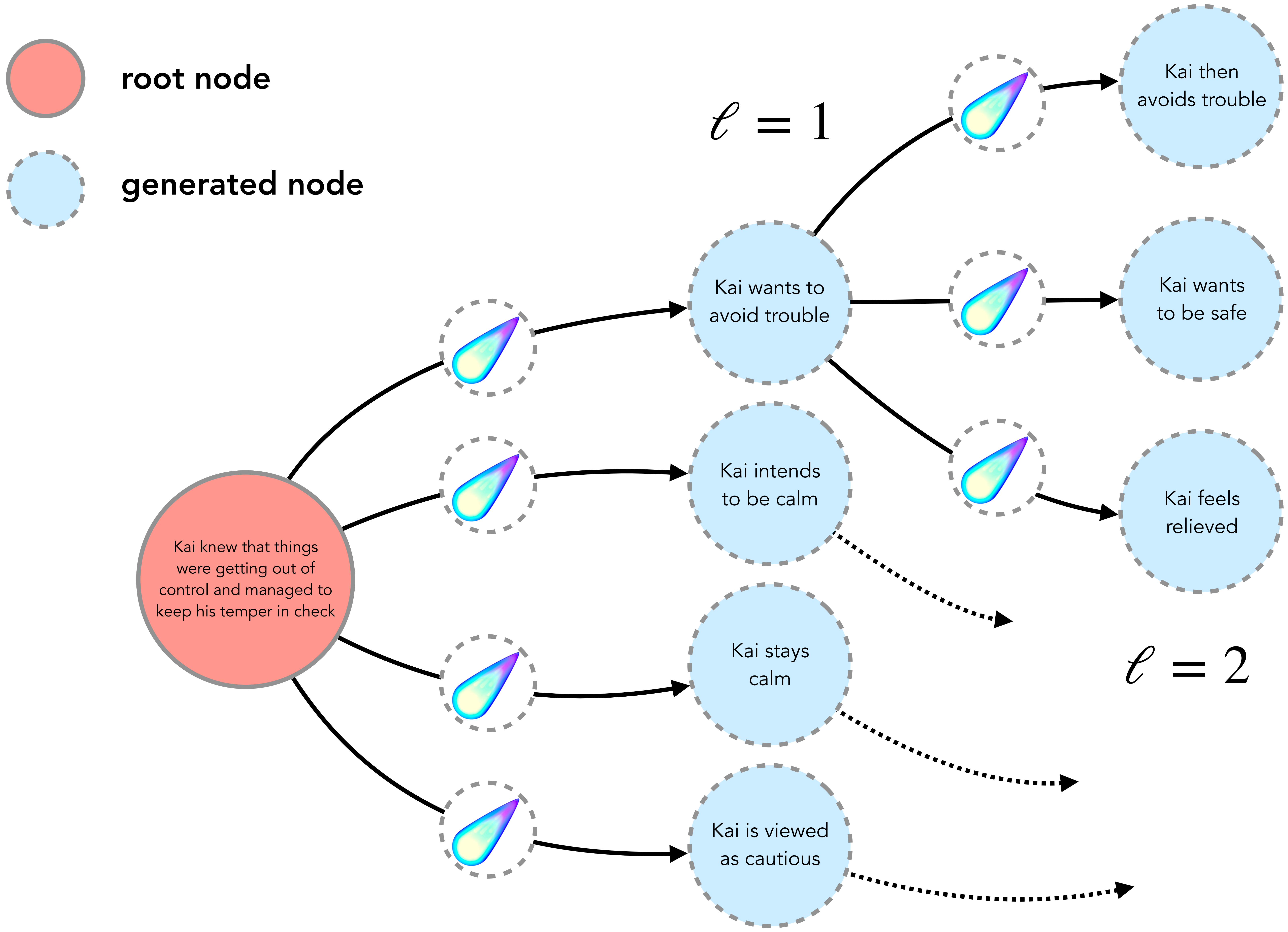


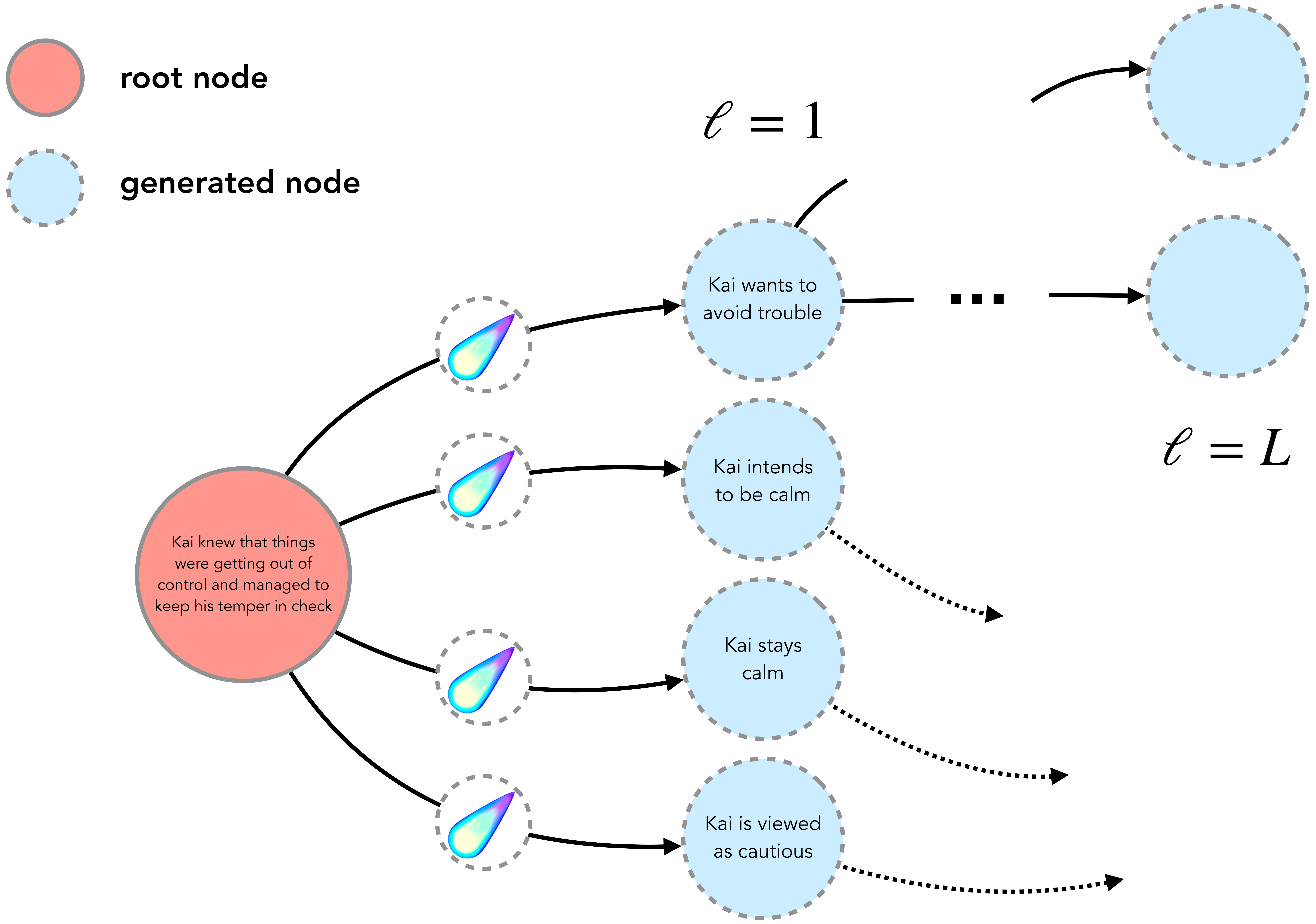
Dynamic Construction of Reasoning Graphs

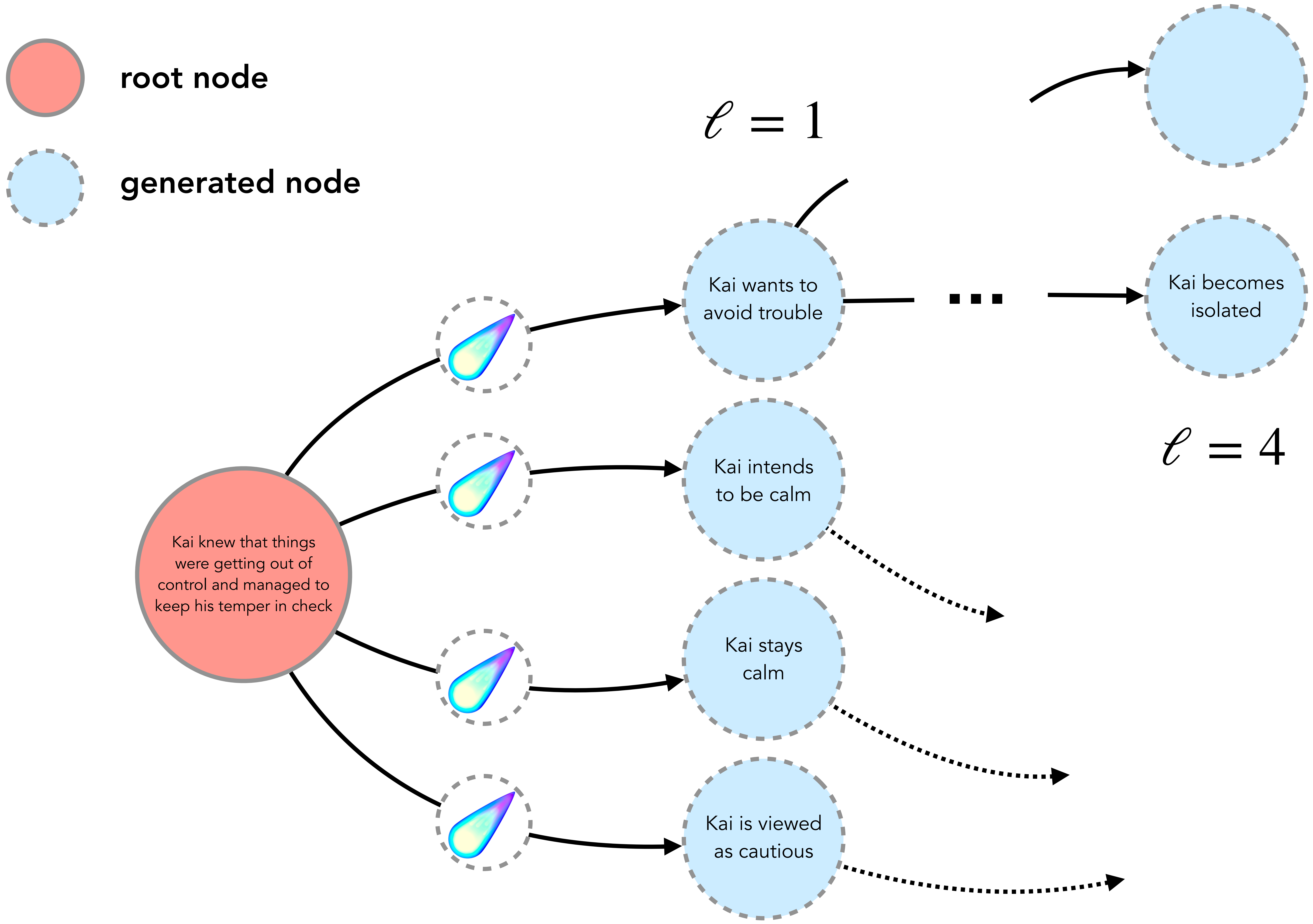



root node
generated node
Kai knew that things were getting out of control and managed to keep his temper in check
Kai wants to avoid trouble
Kai intends to be calm
Kai stays calm
Kai is viewed as cautious

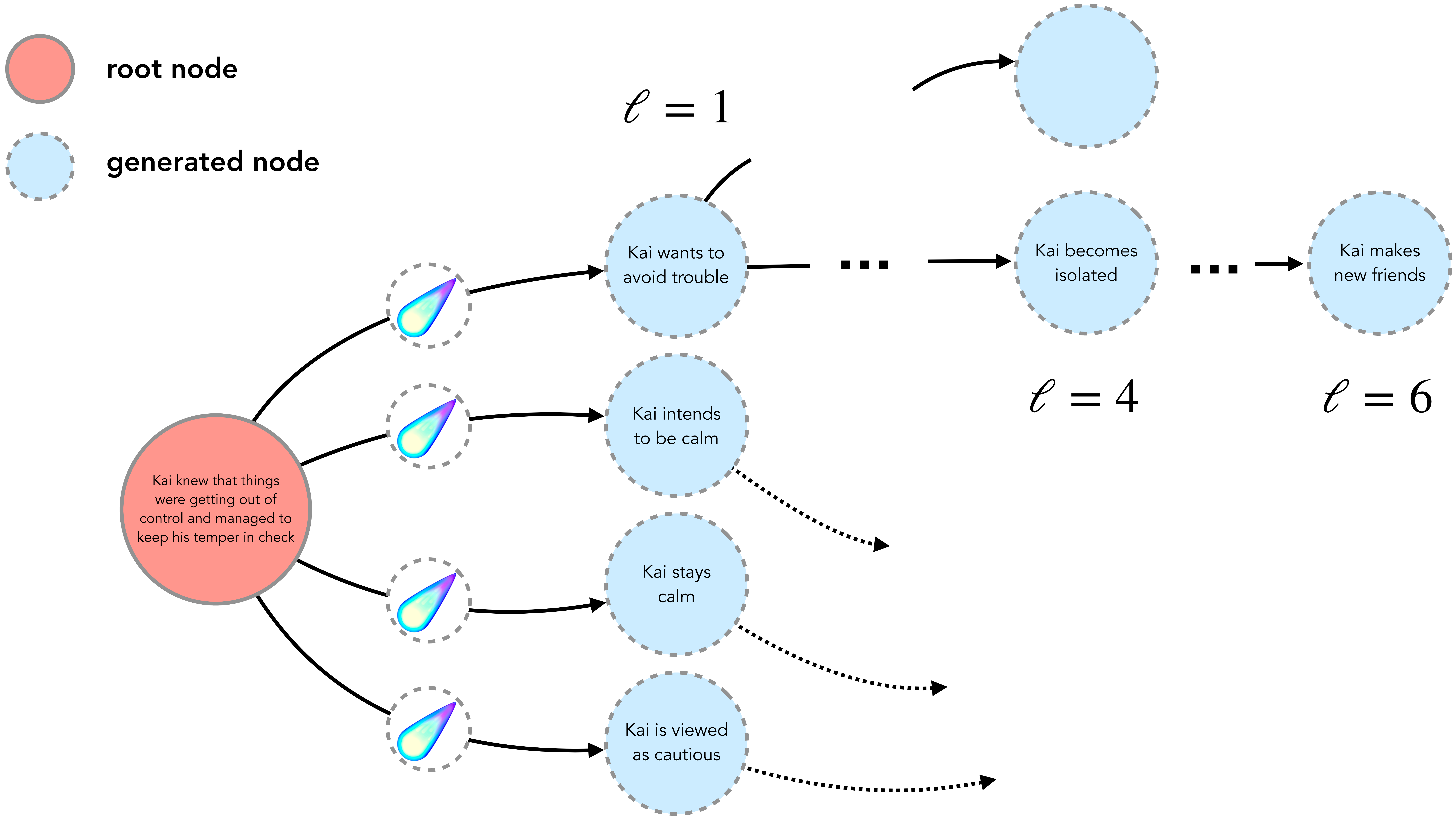

root node
generated node
$\ell = 1$
Kai knew that things were getting out of control and managed to keep his temper in check
Kai wants to avoid trouble
Kai intends to be calm
Kai stays calm
Kai is viewed as cautious

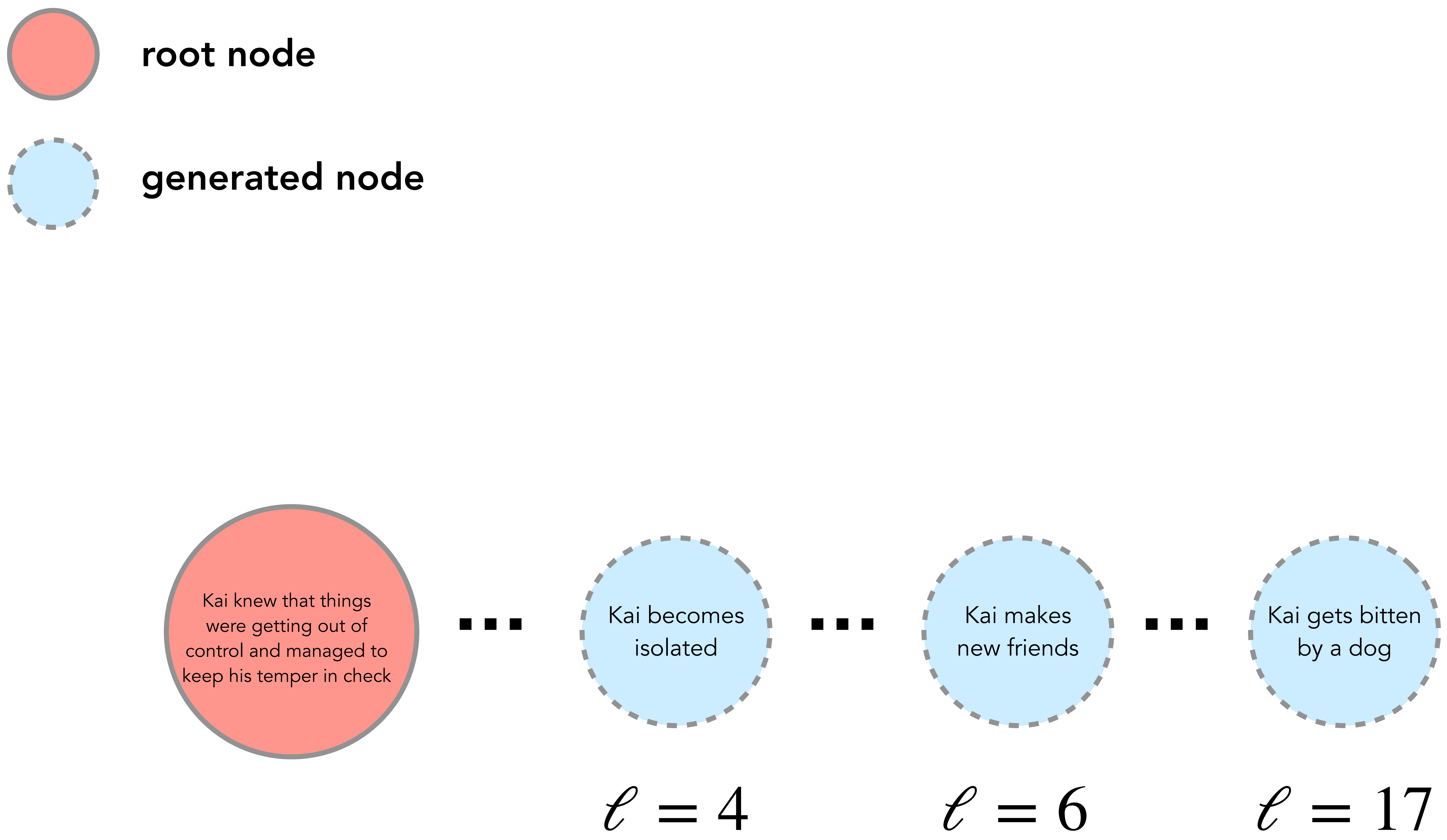

root node
generated node
$\ell = 1$
$\ell = 2$
Kai knew that things were getting out of control and managed to keep his temper in check
Kai wants to avoid trouble
Kai intends to be calm
Kai stays calm
Kai is viewed as cautious
Kai then avoids trouble
Kai wants to be safe
Kai feels relieved

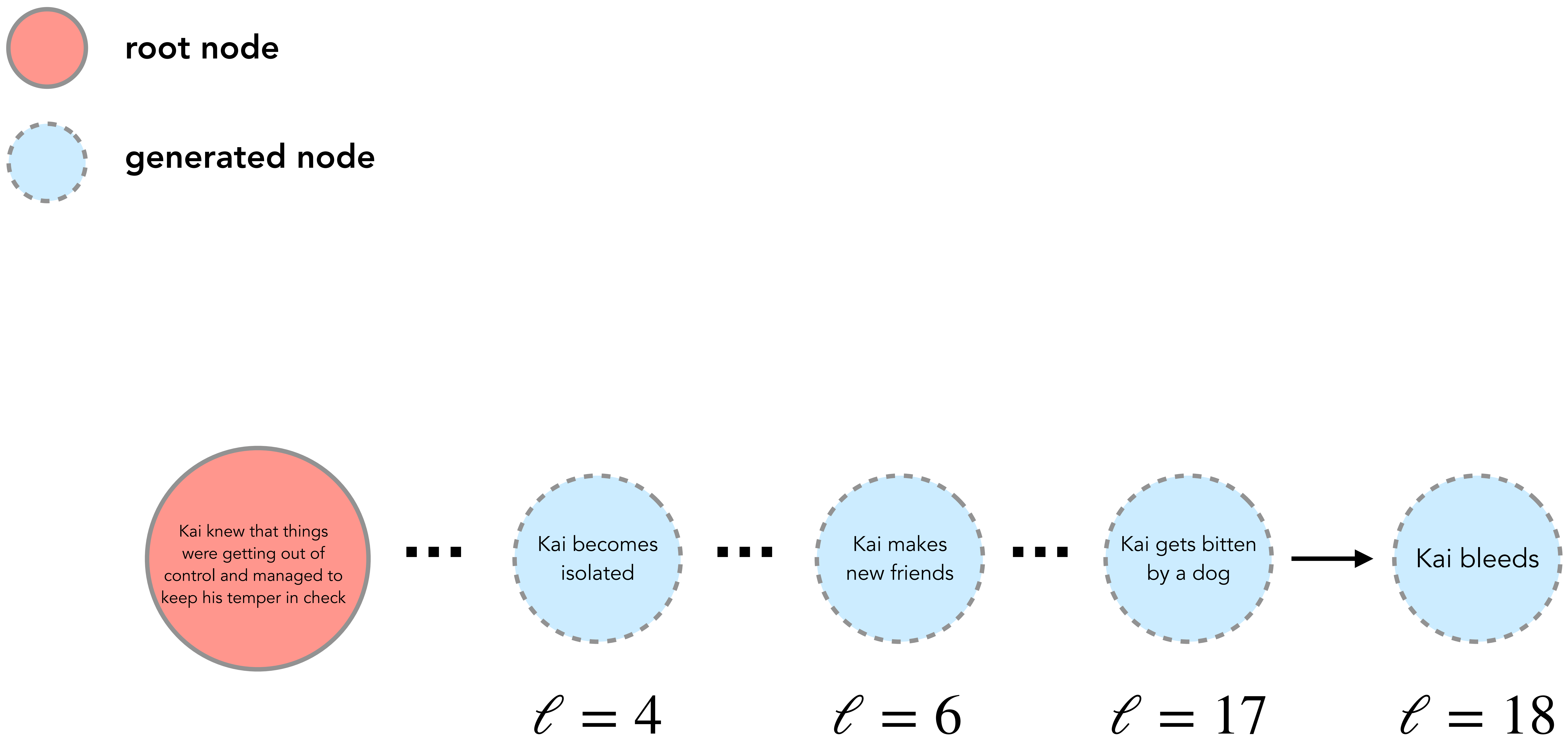

root node
generated node
$\ell = 1$
$\ell = L$
Kai knew that things were getting out of control and managed to keep his temper in check
Kai wants to avoid trouble
Kai intends to be calm
Kai stays calm
Kai is viewed as cautious
...

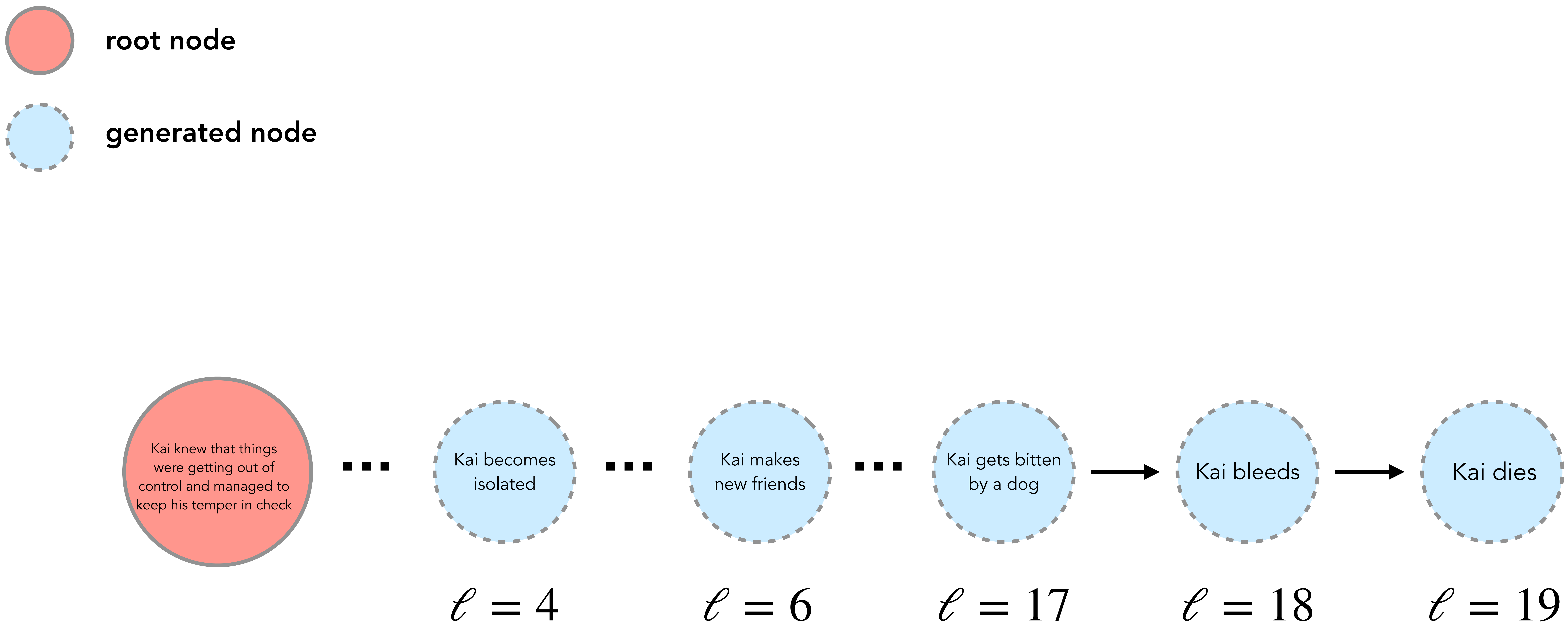


root node
generated node
$\ell = 1$
$\ell = 4$
Kai knew that things were getting out of control and managed to keep his temper in check
Kai wants to avoid trouble
Kai intends to be calm
Kai stays calm
Kai is viewed as cautious
...
Kai becomes isolated


root node
generated node
$\ell = 1$
Kai wants to avoid trouble
Kai becomes isolated
Kai makes new friends
$\ell = 4$
$\ell = 6$
Kai intends to be calm
Kai knew that things were getting out of control and managed to keep his temper in check
Kai stays calm
Kai is viewed as cautious


root node
generated node
Kai knew that things were getting out of control and managed to keep his temper in check
Kai becomes isolated
Kai makes new friends
Kai gets bitten by a dog
$\ell = 4$
$\ell = 6$
$\ell = 17$


root node
generated node
Kai knew that things were getting out of control and managed to keep his temper in check
Kai becomes isolated
Kai makes new friends
Kai gets bitten by a dog
Kai bleeds
$\ell = 4$
$\ell = 6$
$\ell = 17$
$\ell = 18$


root node
generated node
Kai knew that things were getting out of control and managed to keep his temper in check
Kai becomes isolated
Kai makes new friends
Kai gets bitten by a dog
Kai bleeds
Kai dies
$\ell = 4$
$\ell = 6$
$\ell = 17$
$\ell = 18$
$\ell = 19$

Answers as Leaf Nodes

Answer choices can be viewed as nodes in your knowledge graph

Context:

Kai knew that things were getting out of control and managed to keep his temper in check.

Question:

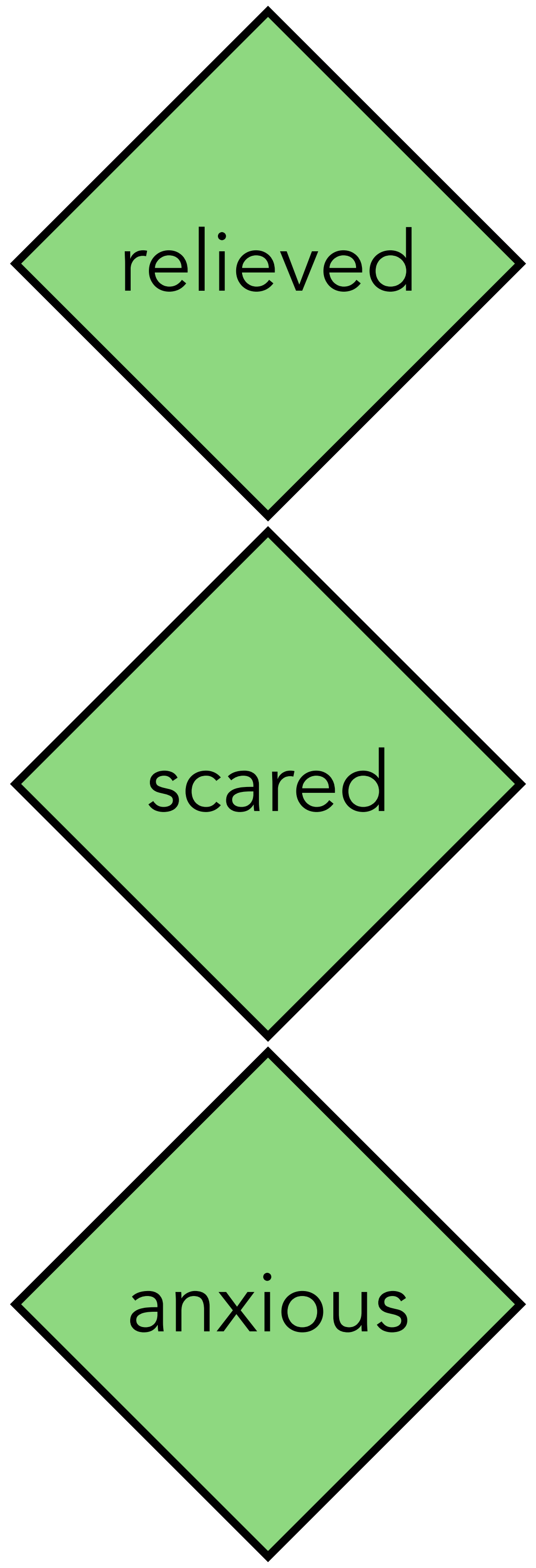
How would others feel as a result?

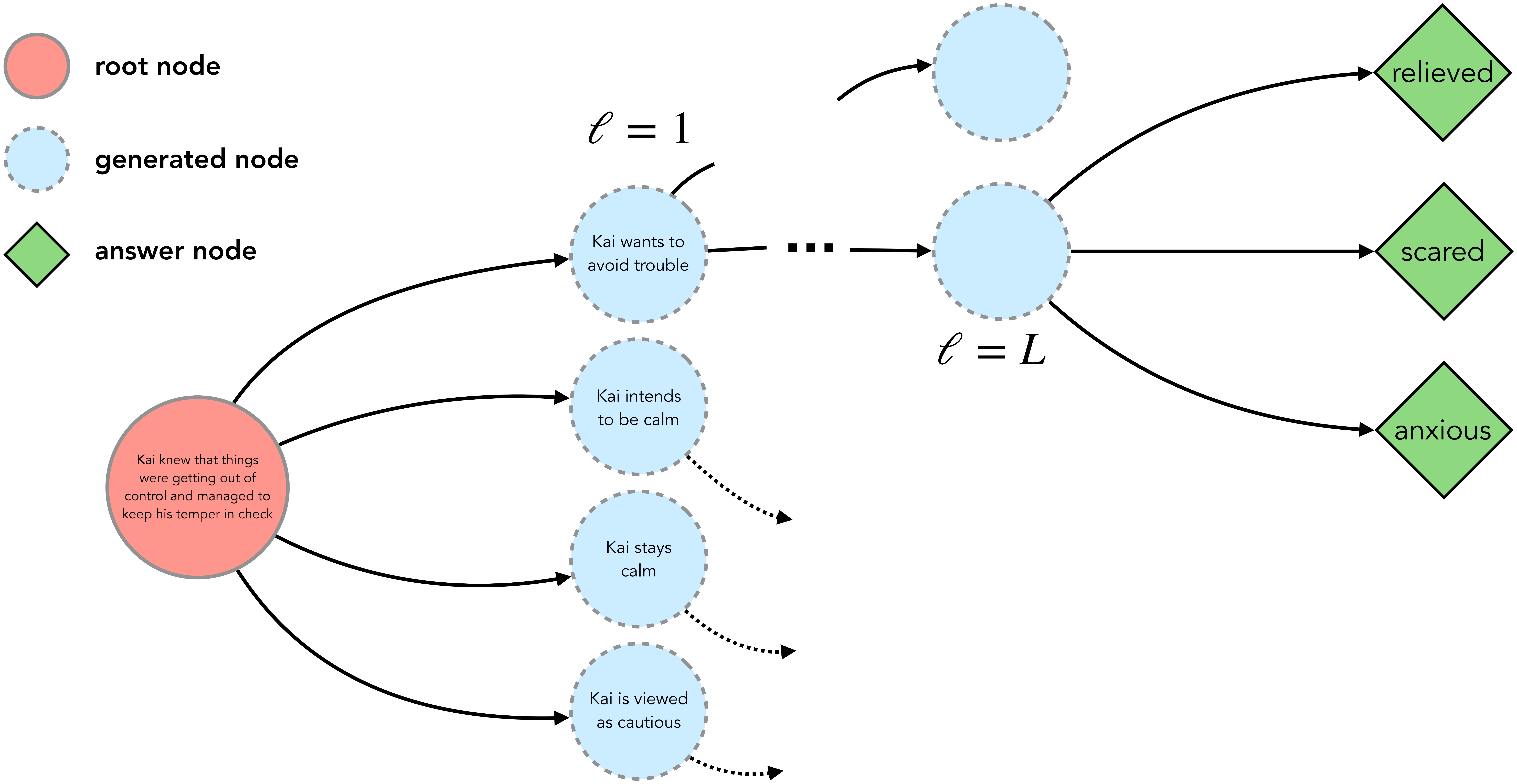
Answer Options:

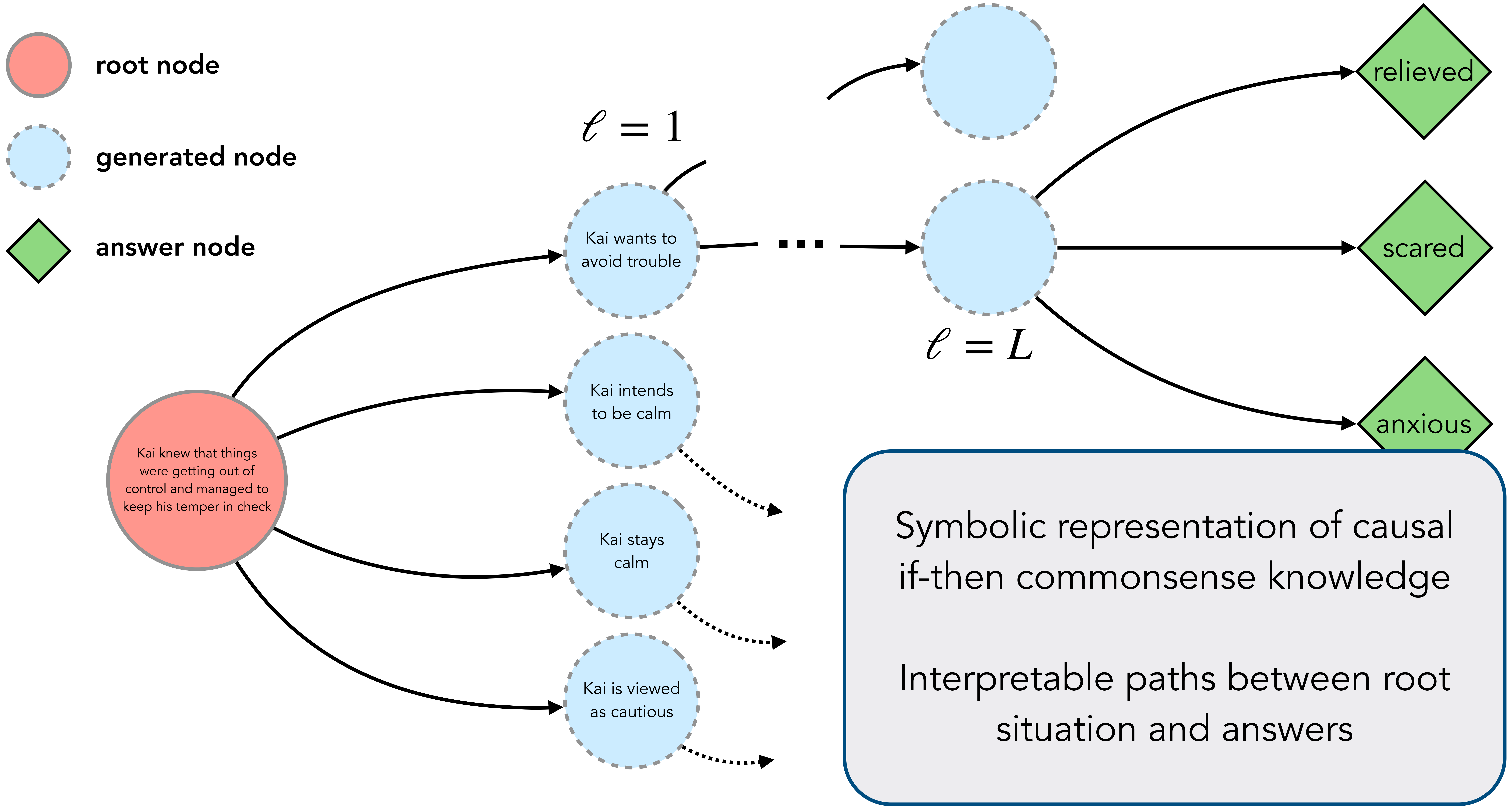
a) relieved

b) scared

c) anxious

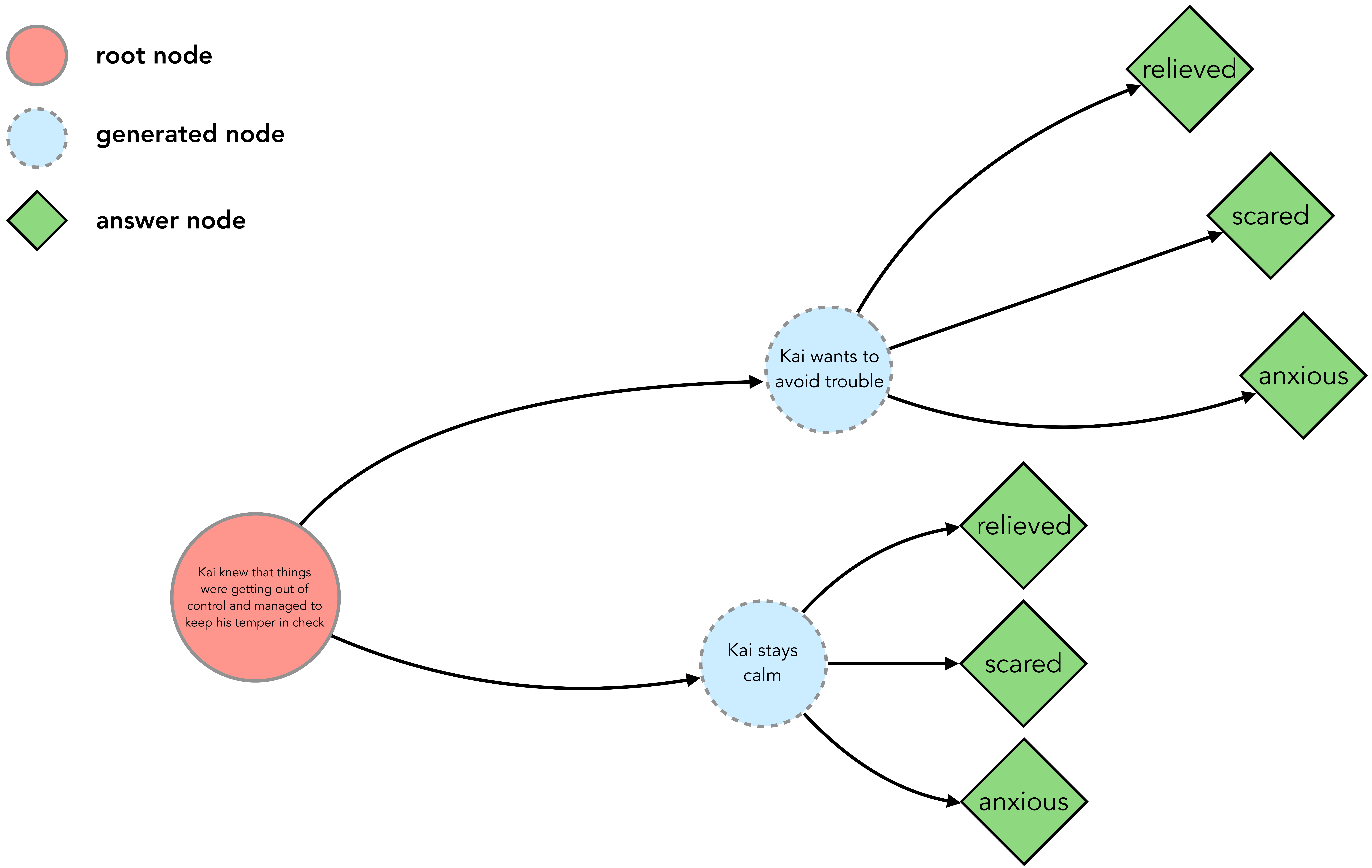


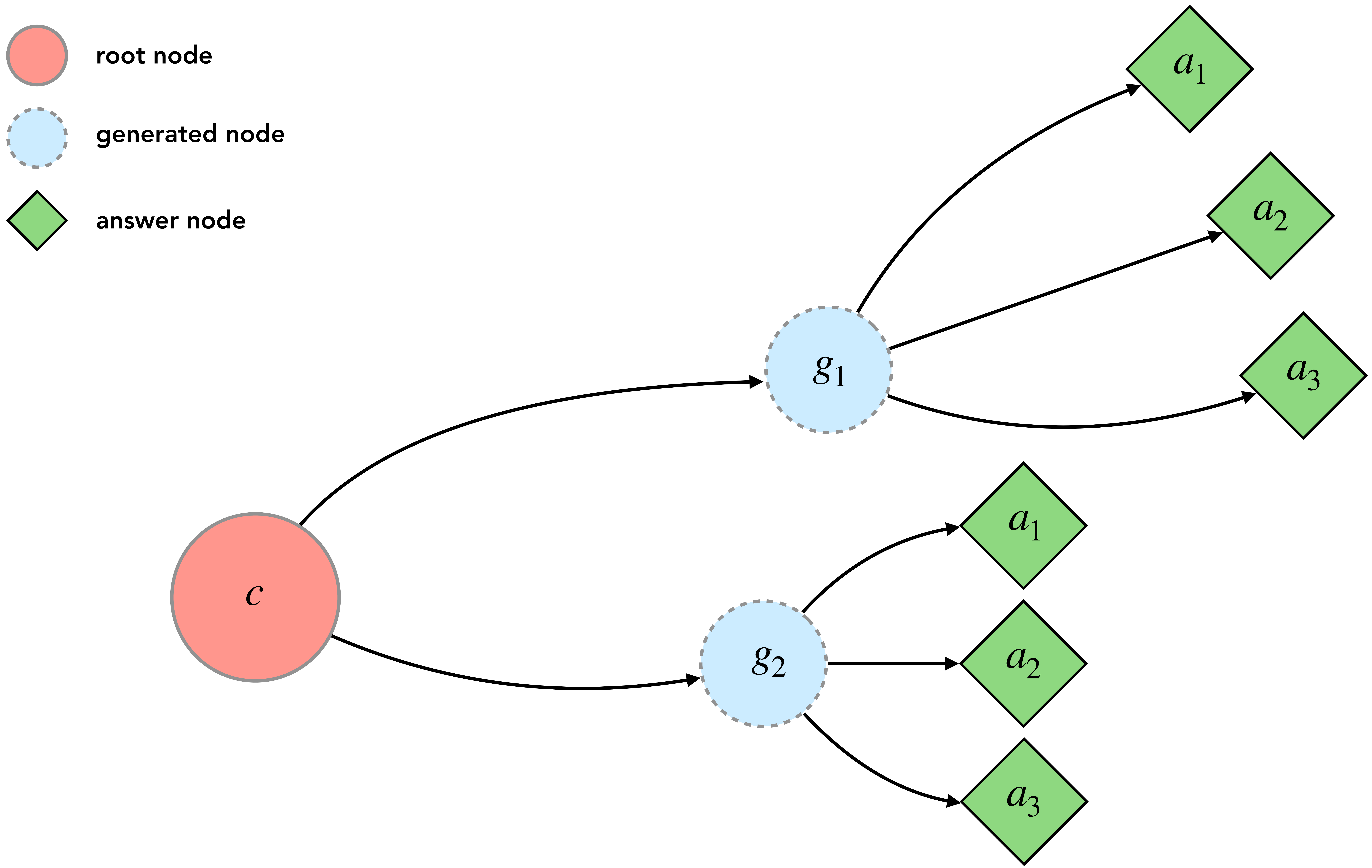

root node
generated node
answer node
Kai knew that things were getting out of control and managed to keep his temper in check
$\ell = 1$
Kai wants to avoid trouble
Kai intends to be calm
Kai stays calm
Kai is viewed as cautious
...
$\ell = L$
relieved
scared
anxious



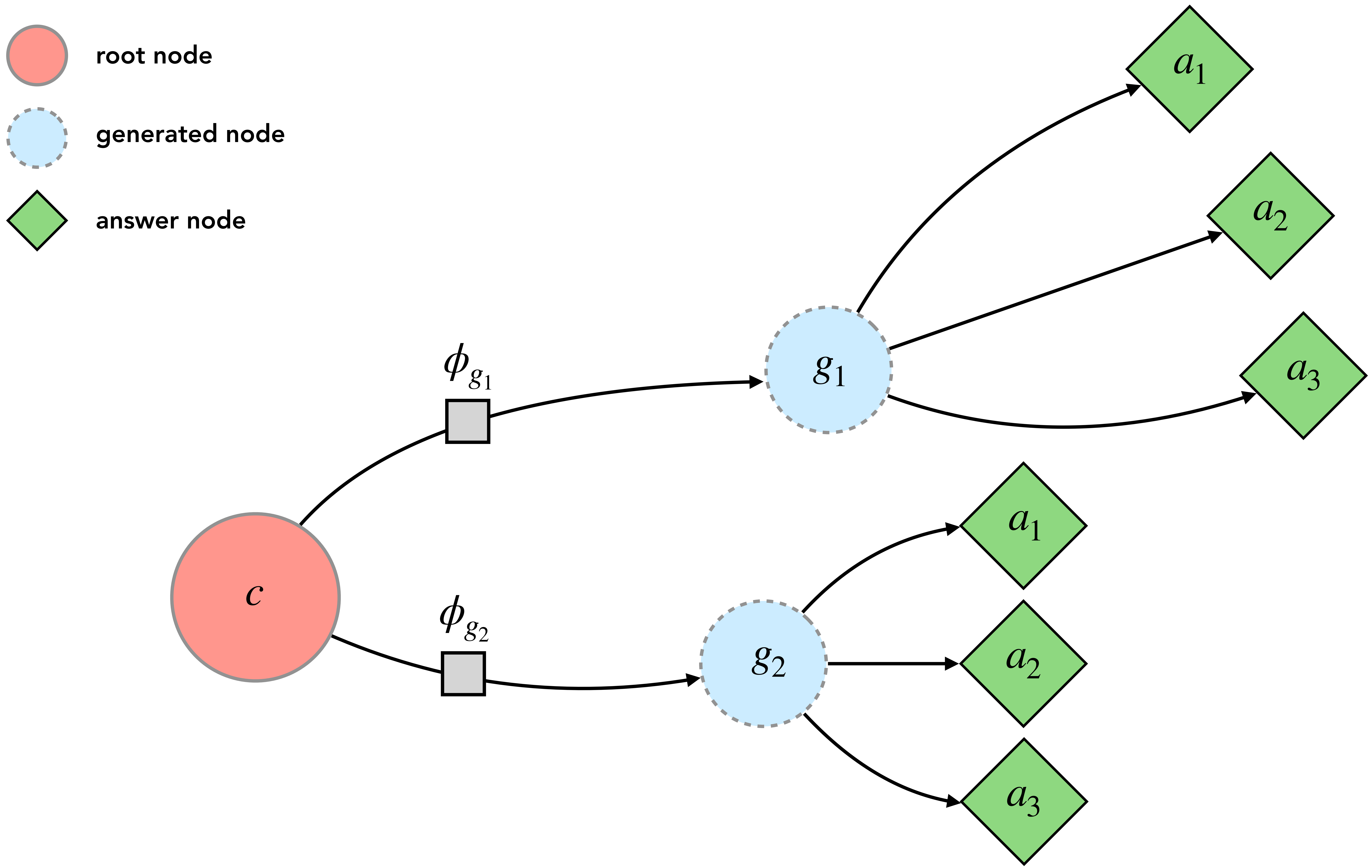
root node
generated node
answer node
Kai knew that things were getting out of control and managed to keep his temper in check
Kai wants to avoid trouble
Kai intends to be calm
Kai stays calm
Kai is viewed as cautious
$\ell = 1$
$\ell = L$
relieved
scared
anxious
Symbolic representation of causal if-then commonsense knowledge
Interpretable paths between root situation and answers

How do we reason over the commonsense knowledge in the graph?


root node
generated node
answer node
Kai knew that things were getting out of control and managed to keep his temper in check
Kai wants to avoid trouble
relieved
scared
anxious
Kai stays calm
relieved
scared
anxious

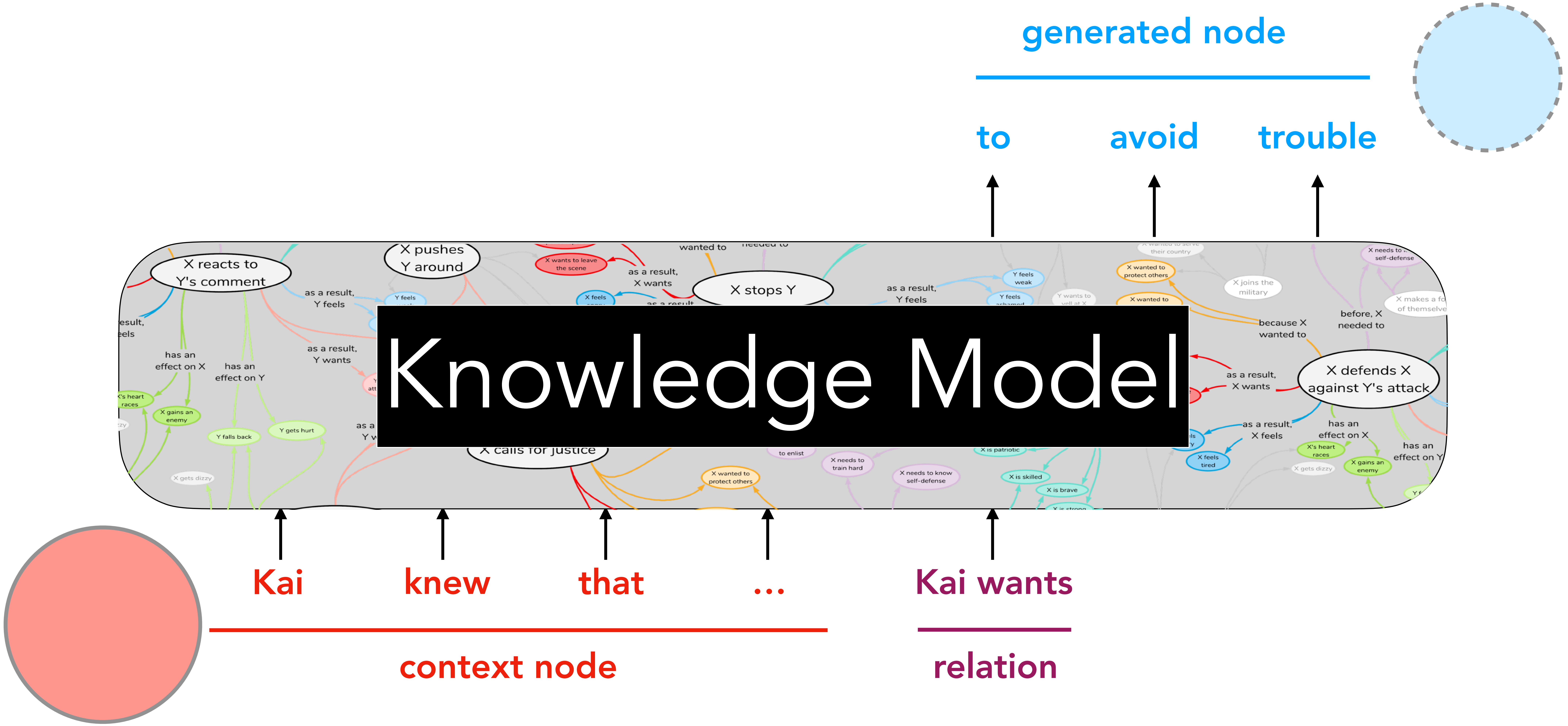


root node
generated node
answer node
a_1
a_2
a_3
g_1
c
a_1
g_2
a_2
a_3



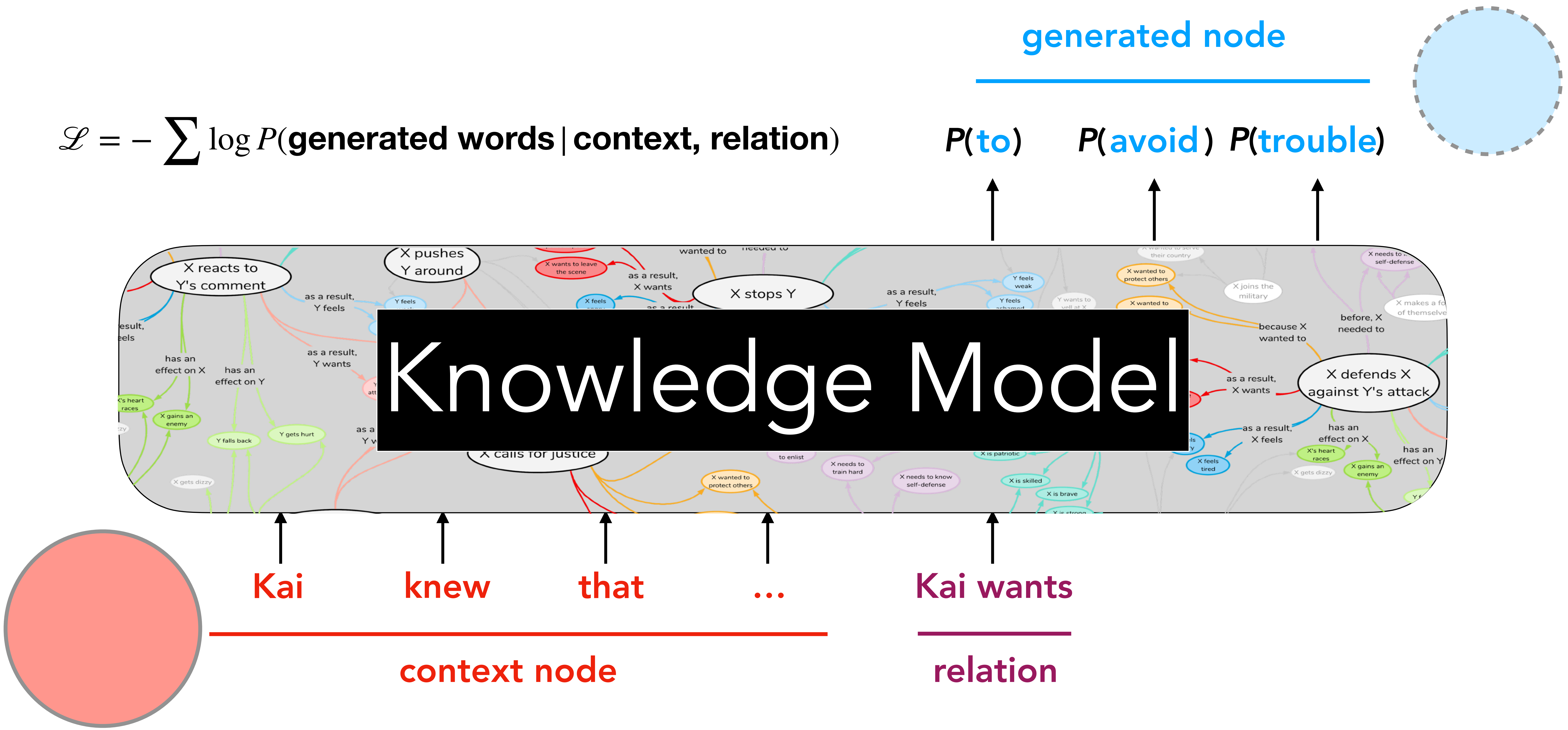
root node
generated node
answer node
c
ϕ_{g_1}
ϕ_{g_2}
g_1
g_2
a_1
a_2
a_3
a_1
a_2
a_3

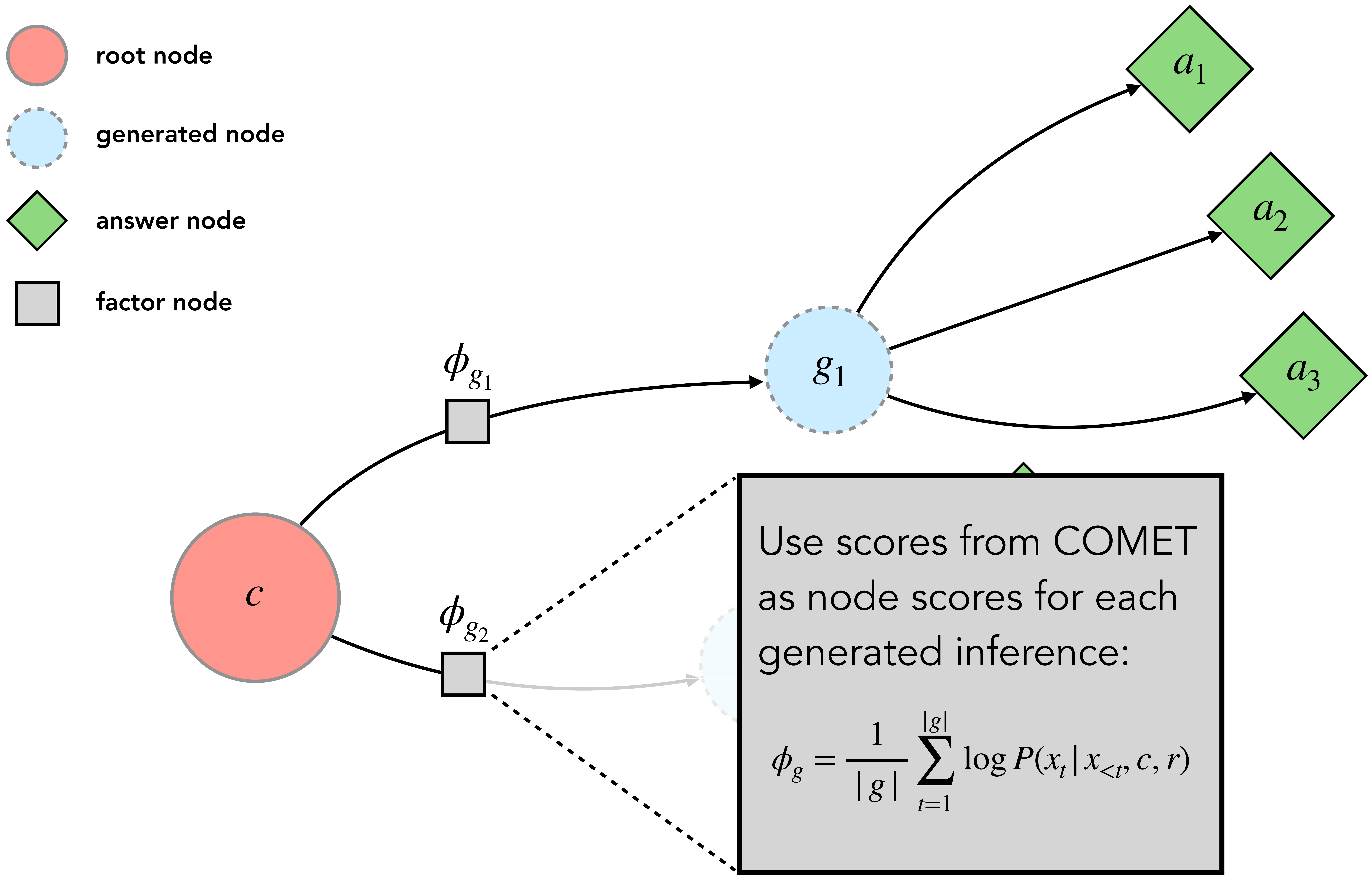
Language Model learns Knowledge Structure



Language Model learns Knowledge Structure

$$\mathscr{L} = -\sum \log P(\textbf{generated words} \,|\, \textbf{context, relation})$$

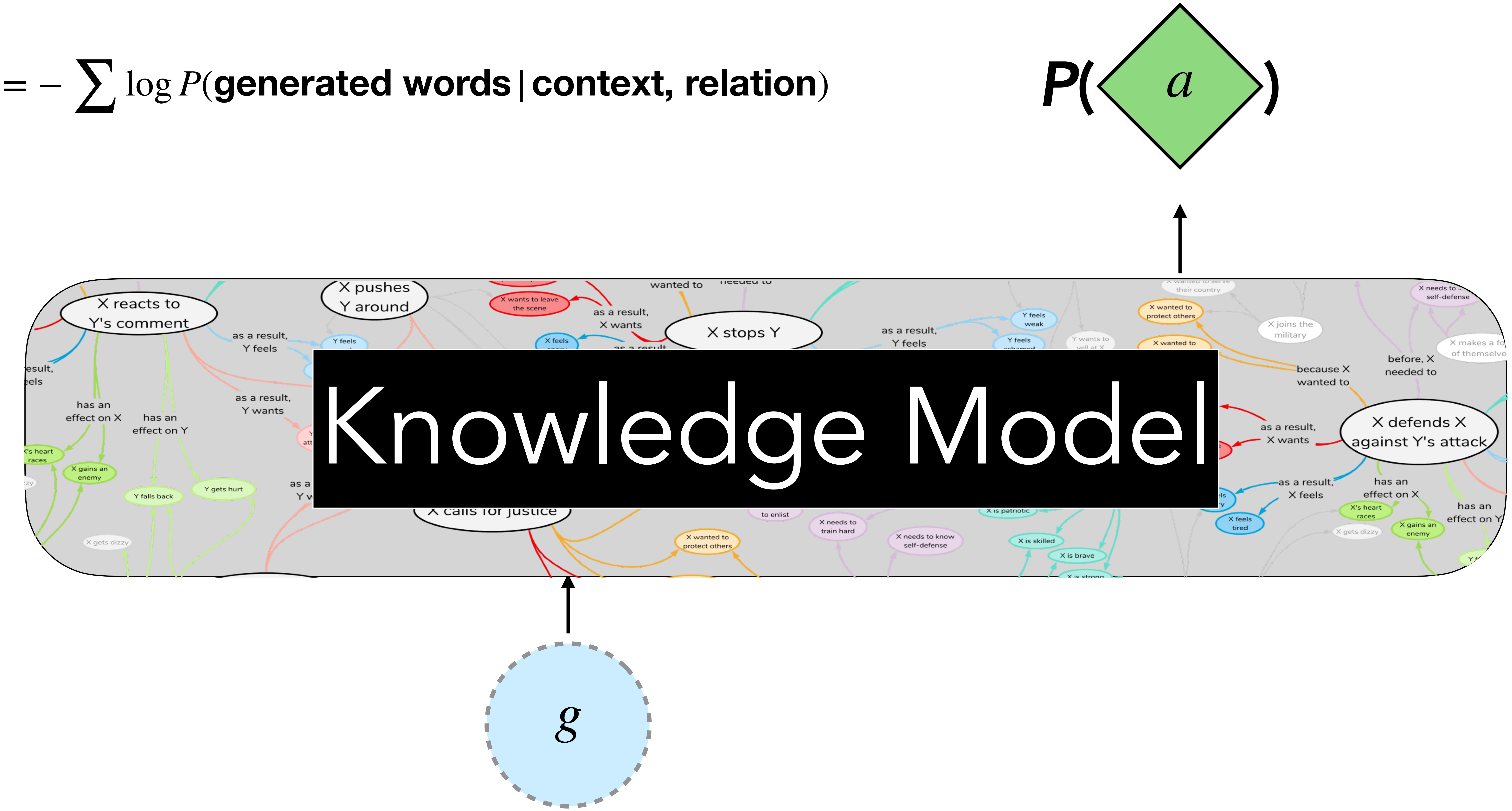


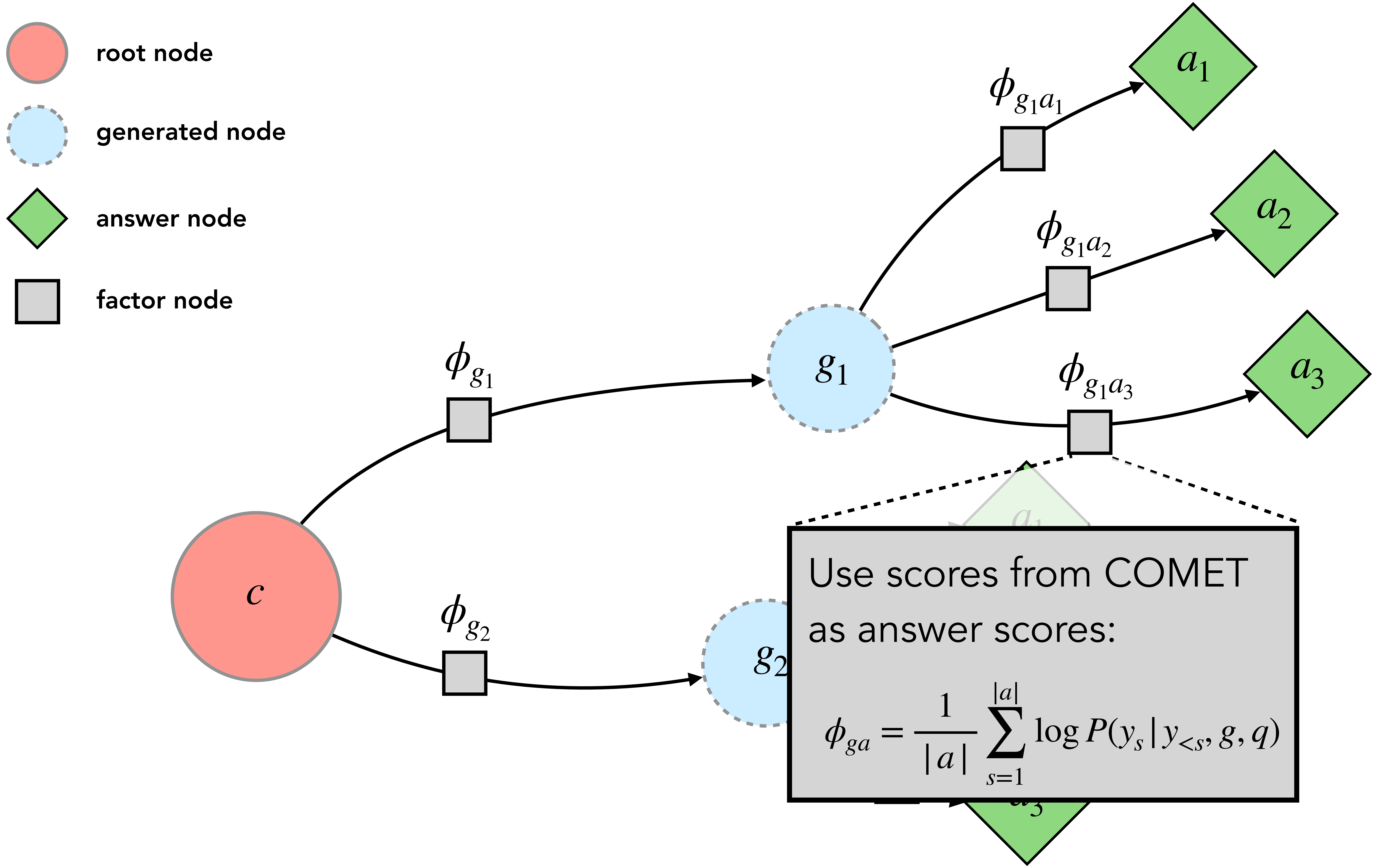


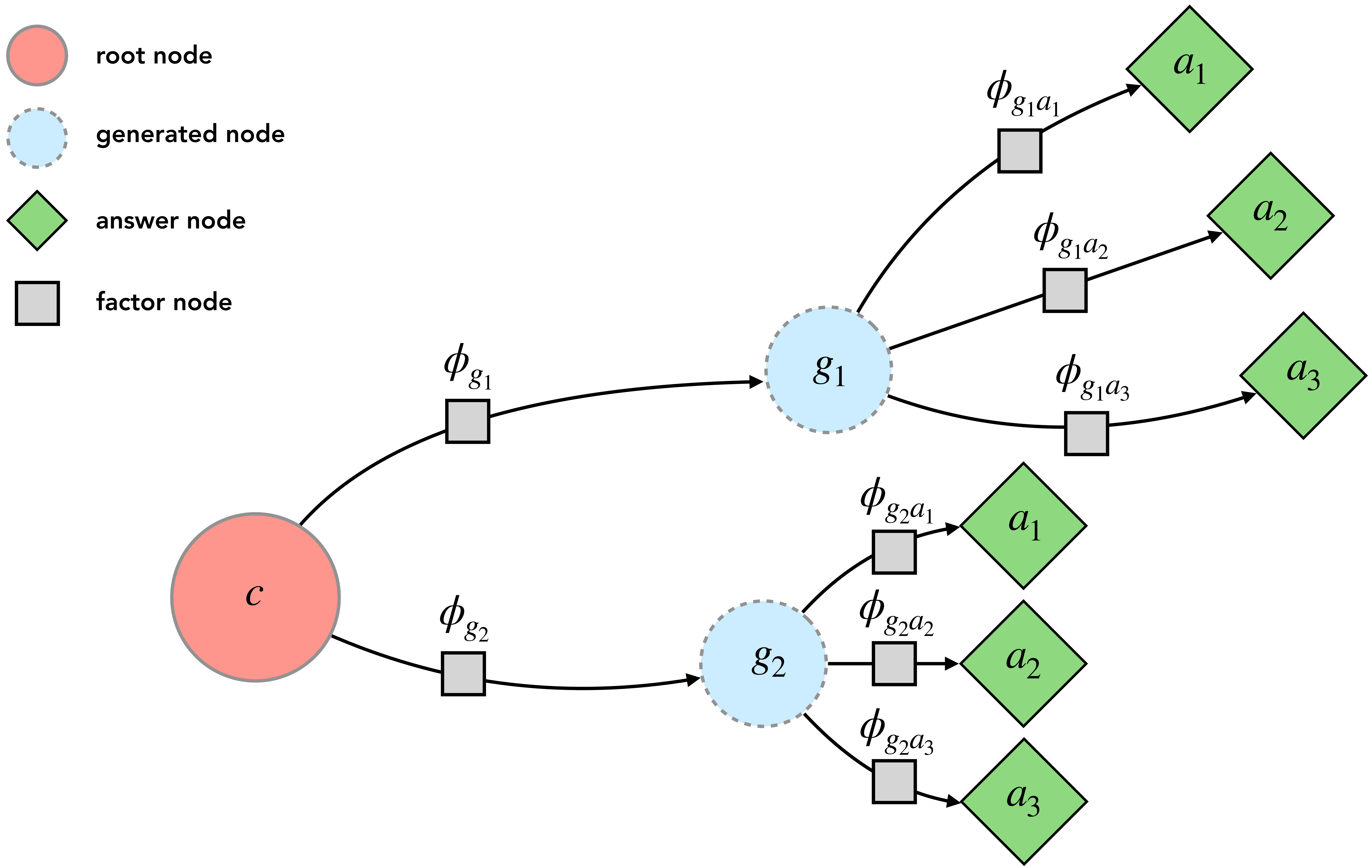
root node
generated node
answer node
factor node
a_1
a_2
a_3
g_1
ϕ_{g_1}
ϕ_{g_2}
c
Use scores from COMET as node scores for each generated inference:
$\phi_g = \frac{1}{|g|} \sum_{t=1}^{|g|} \log P(x_t | x_{<t}, c, r)$

Language Model learns Knowledge Structure

$$\mathscr{L} = -\sum \log P(\textbf{generated words} \,|\, \textbf{context, relation})$$

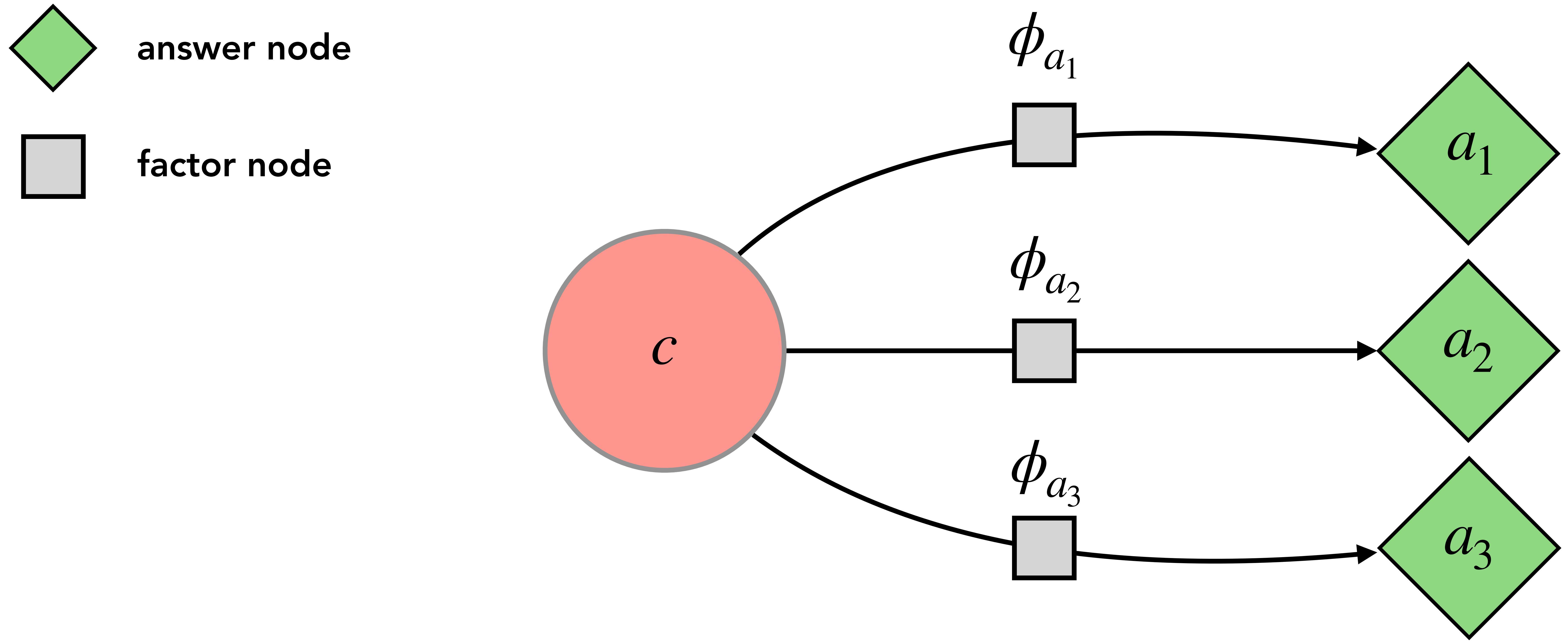



root node
generated node
answer node
factor node
c
ϕ_{g_1}
ϕ_{g_2}
g_1
g_2
$\phi_{g_1 a_1}$
$\phi_{g_1 a_2}$
$\phi_{g_1 a_3}$
a_1
a_2
a_3
Use scores from COMET
as answer scores:
$$\phi_{ga} = \frac{1}{|a|} \sum_{s=1}^{|a|} \log P(y_s \mid y_{<s}, g, q)$$

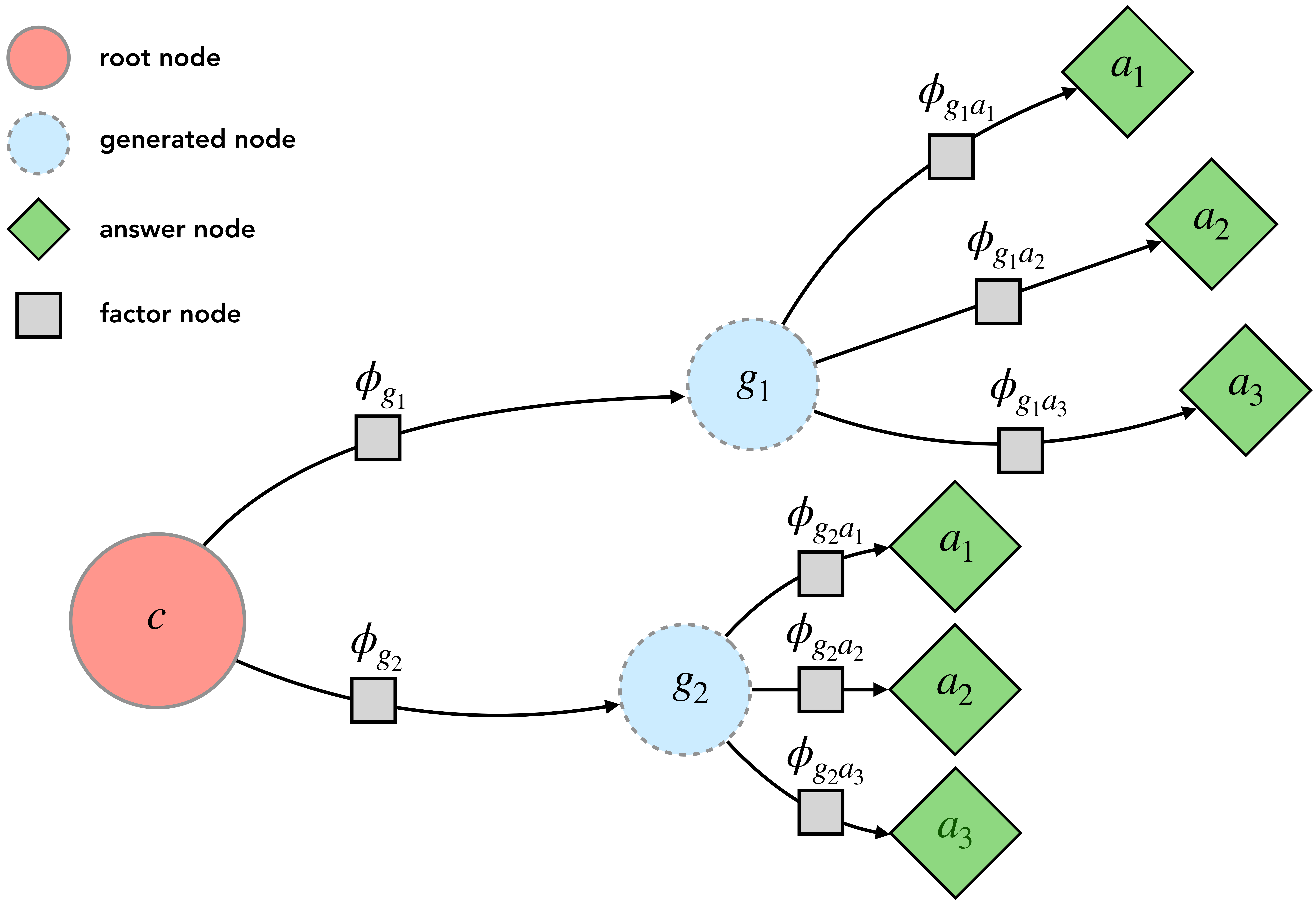


root node
generated node
answer node
factor node
c
ϕ_{g_1}
ϕ_{g_2}
g_1
g_2
$\phi_{g_1 a_1}$
$\phi_{g_1 a_2}$
$\phi_{g_1 a_3}$
$\phi_{g_2 a_1}$
$\phi_{g_2 a_2}$
$\phi_{g_2 a_3}$
a_1
a_2
a_3
a_1
a_2
a_3

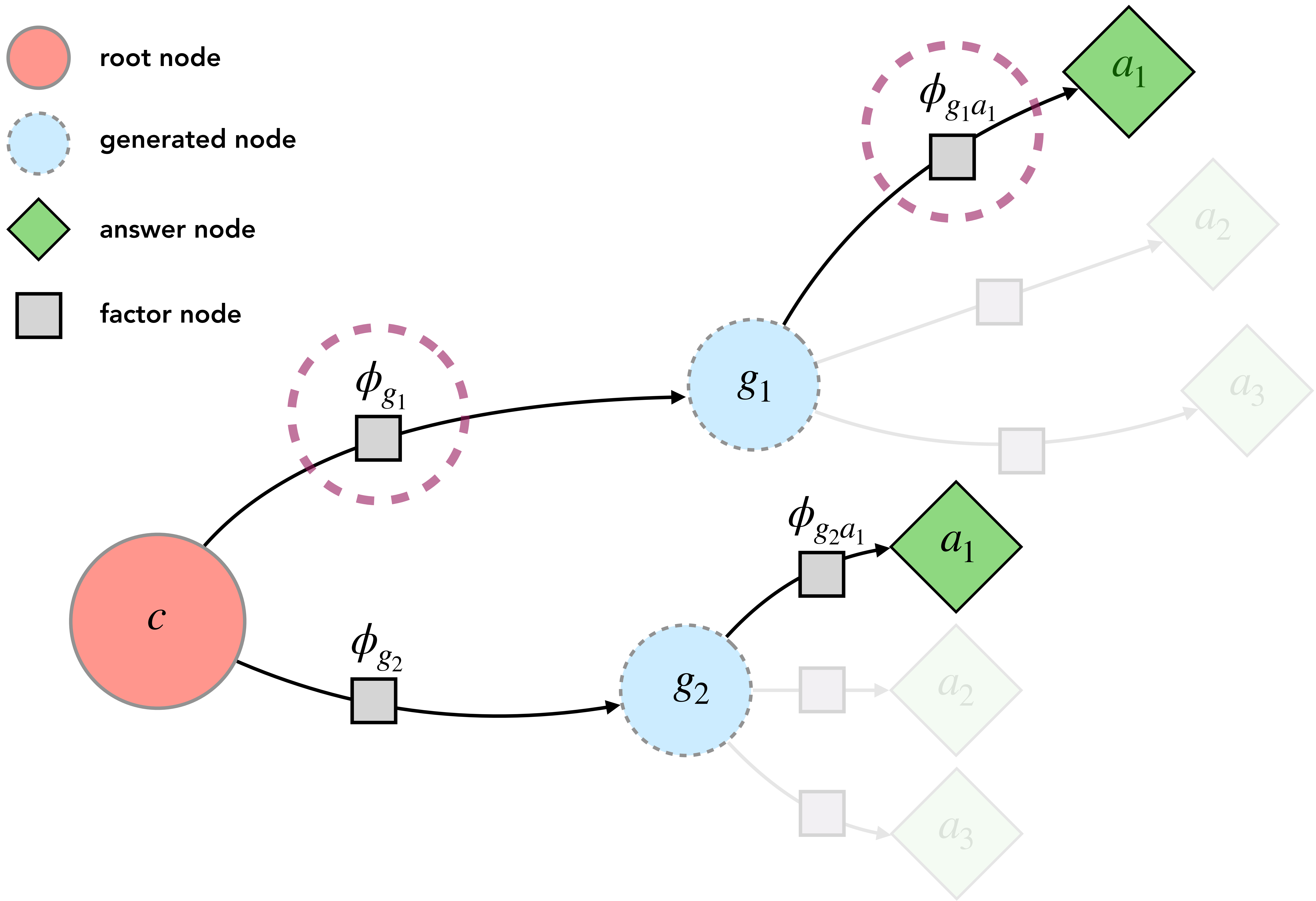
Desired Final Structure



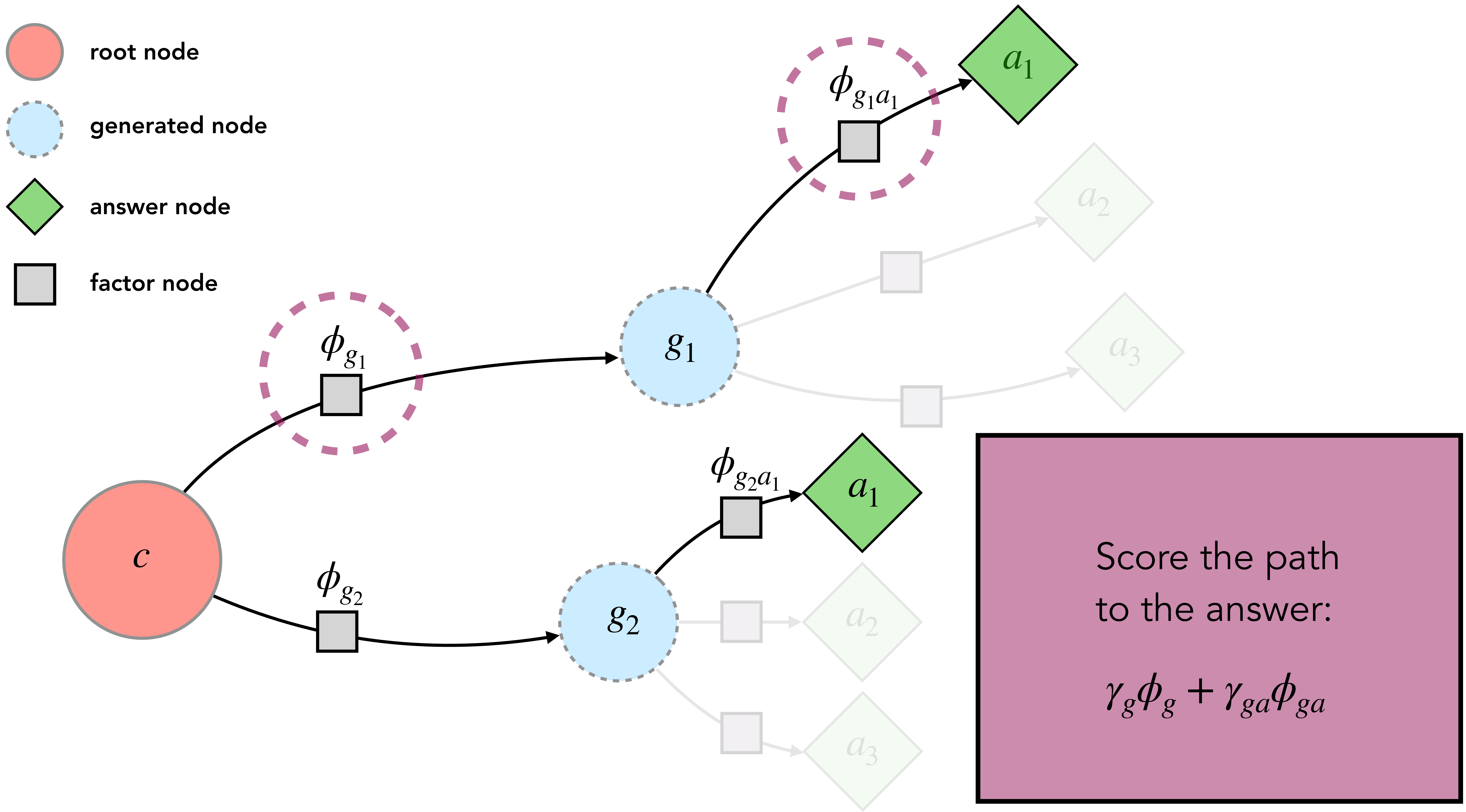
$$\hat{a} = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, \phi_a$$

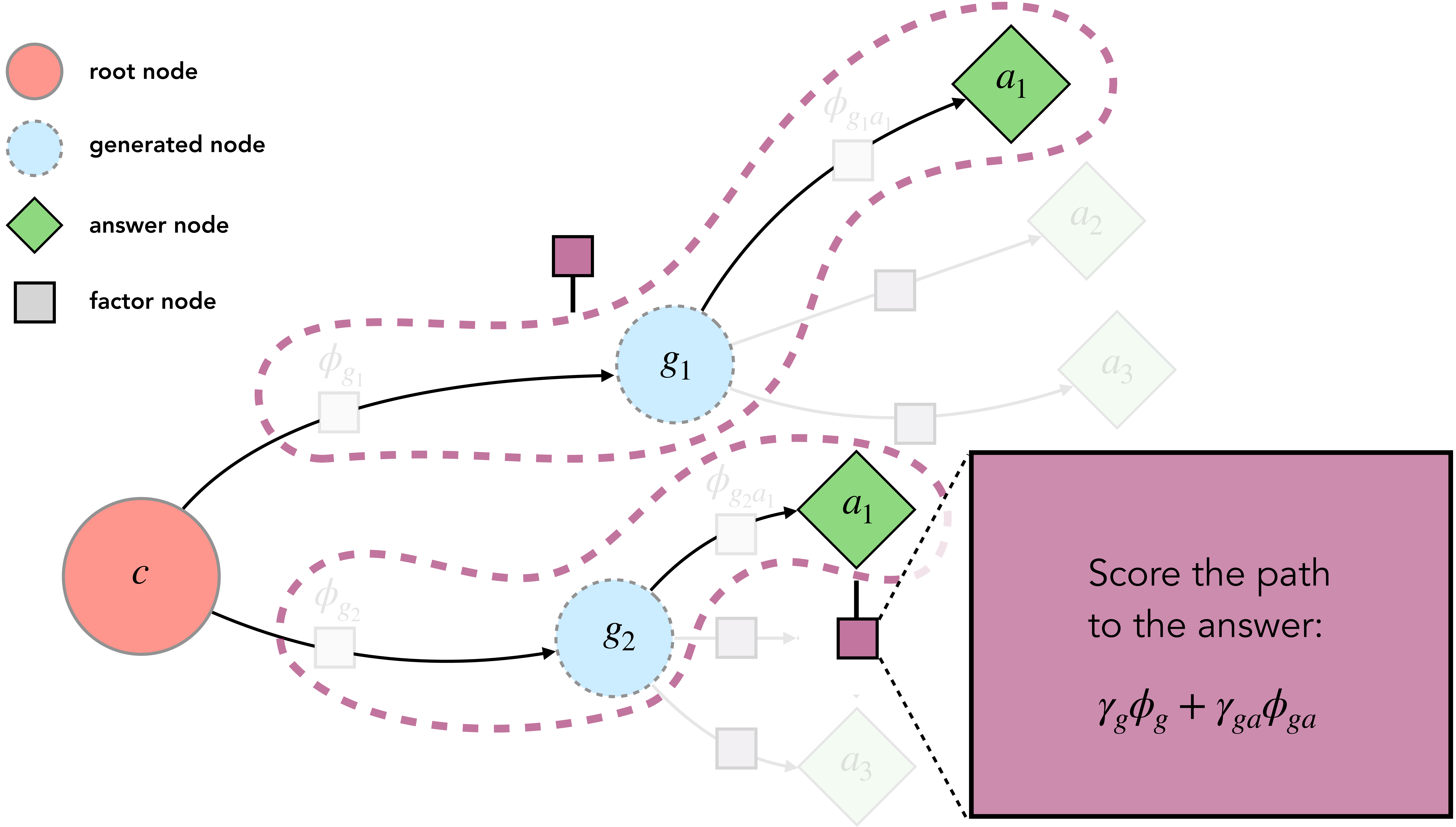


root node
generated node
answer node
factor node
c
ϕ_{g_1}
ϕ_{g_2}
g_1
g_2
$\phi_{g_1 a_1}$
$\phi_{g_1 a_2}$
$\phi_{g_1 a_3}$
$\phi_{g_2 a_1}$
$\phi_{g_2 a_2}$
$\phi_{g_2 a_3}$
a_1
a_2
a_3
a_1
a_2
a_3

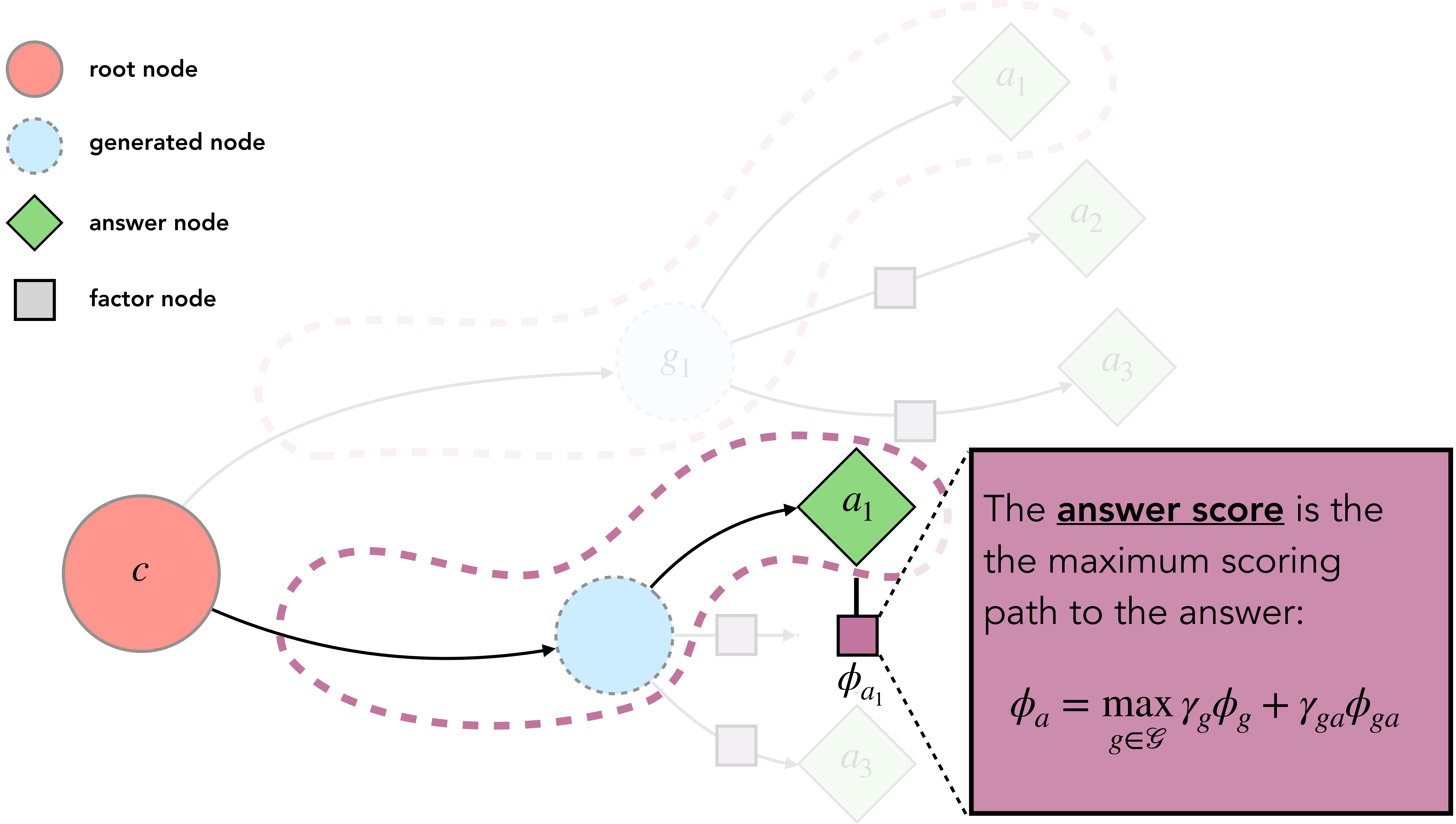


root node
generated node
answer node
factor node
c
ϕ_{g_1}
ϕ_{g_2}
g_1
g_2
$\phi_{g_1 a_1}$
$\phi_{g_2 a_1}$
a_1
a_2
a_3
a_1
a_2
a_3

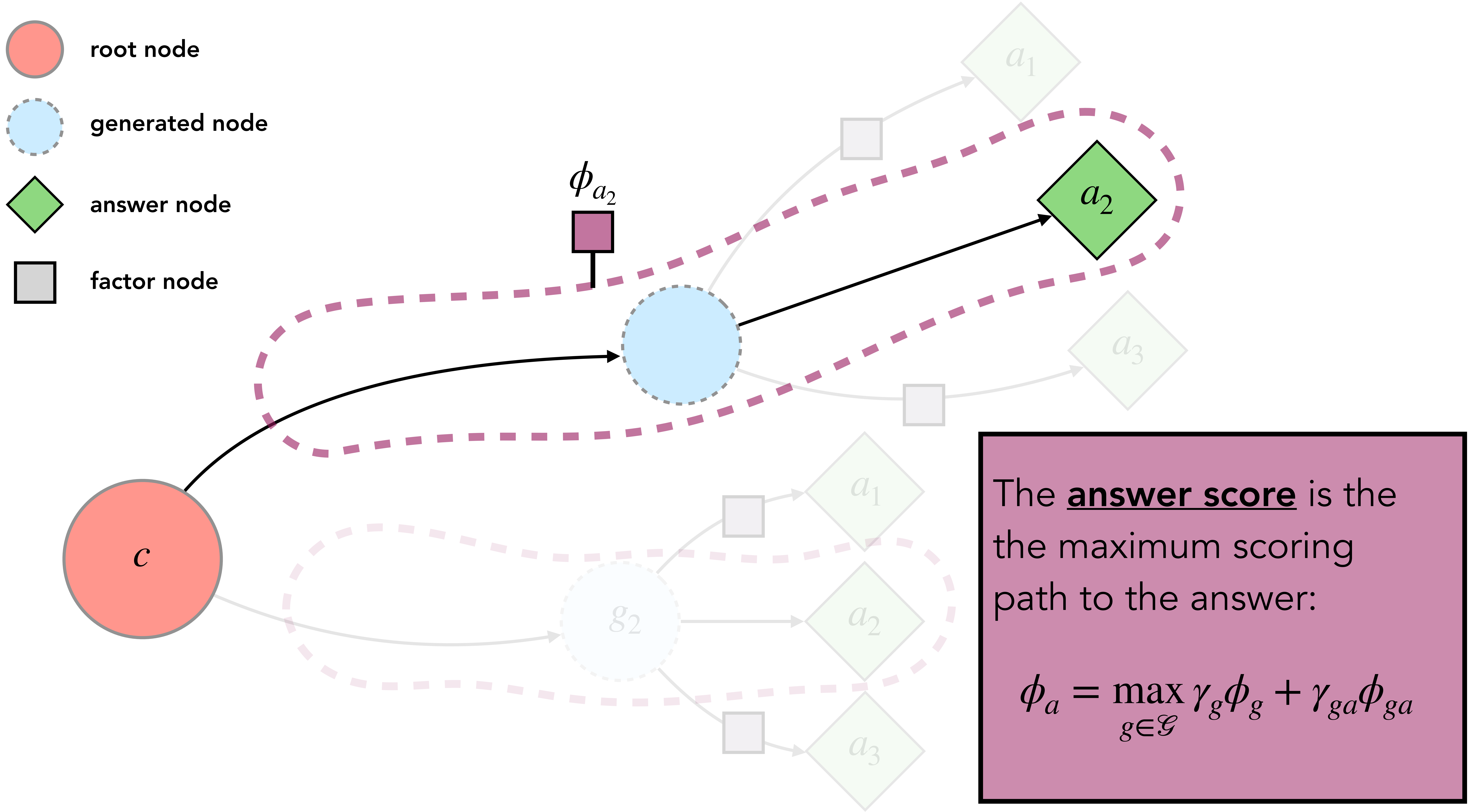

root node
generated node
answer node
factor node
c
ϕ_{g_1}
g_1
$\phi_{g_1 a_1}$
a_1
a_2
a_3
ϕ_{g_2}
g_2
$\phi_{g_2 a_1}$
a_1
a_2
a_3
Score the path
to the answer:
$\gamma_g \phi_g + \gamma_{ga} \phi_{ga}$

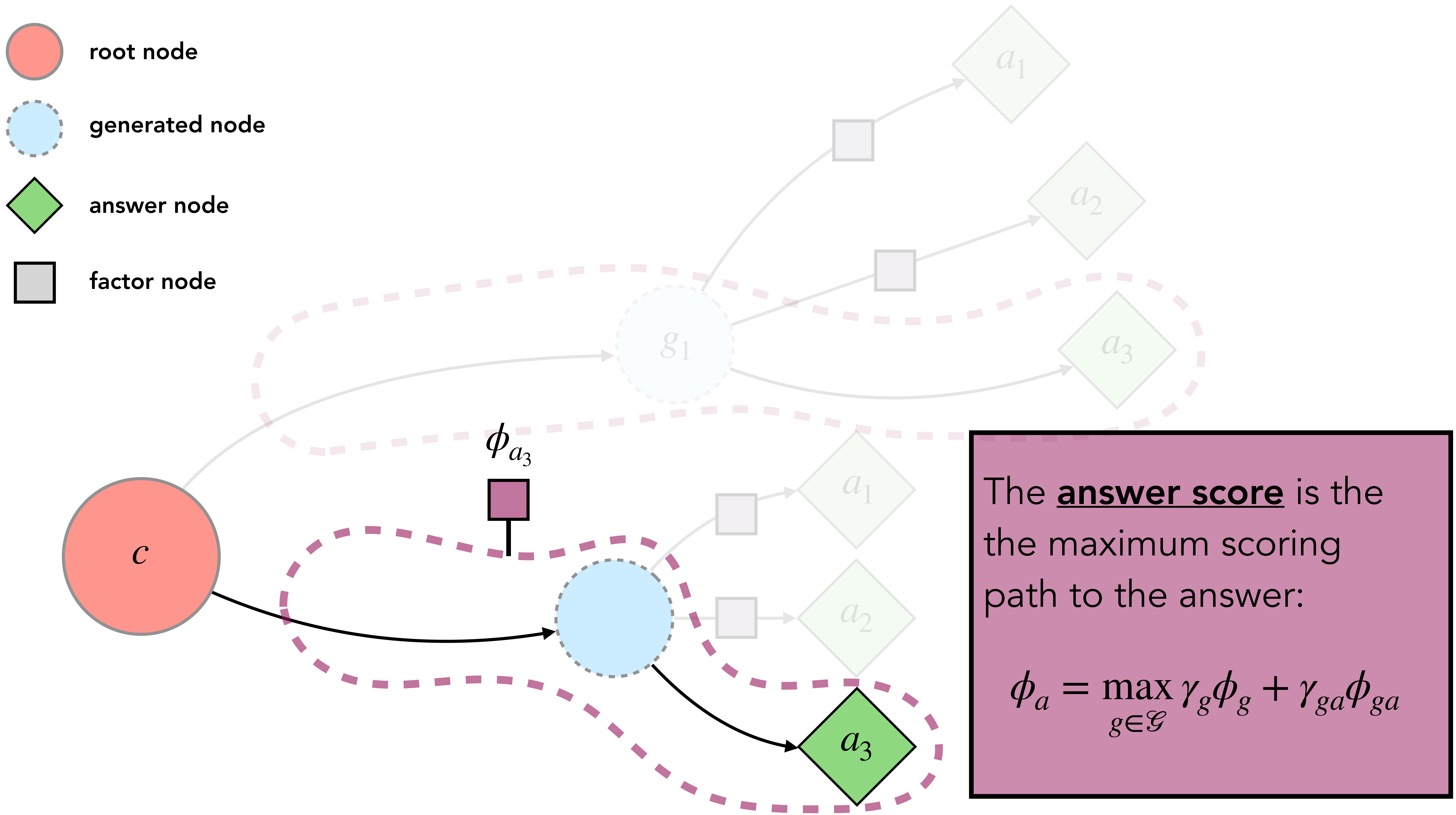


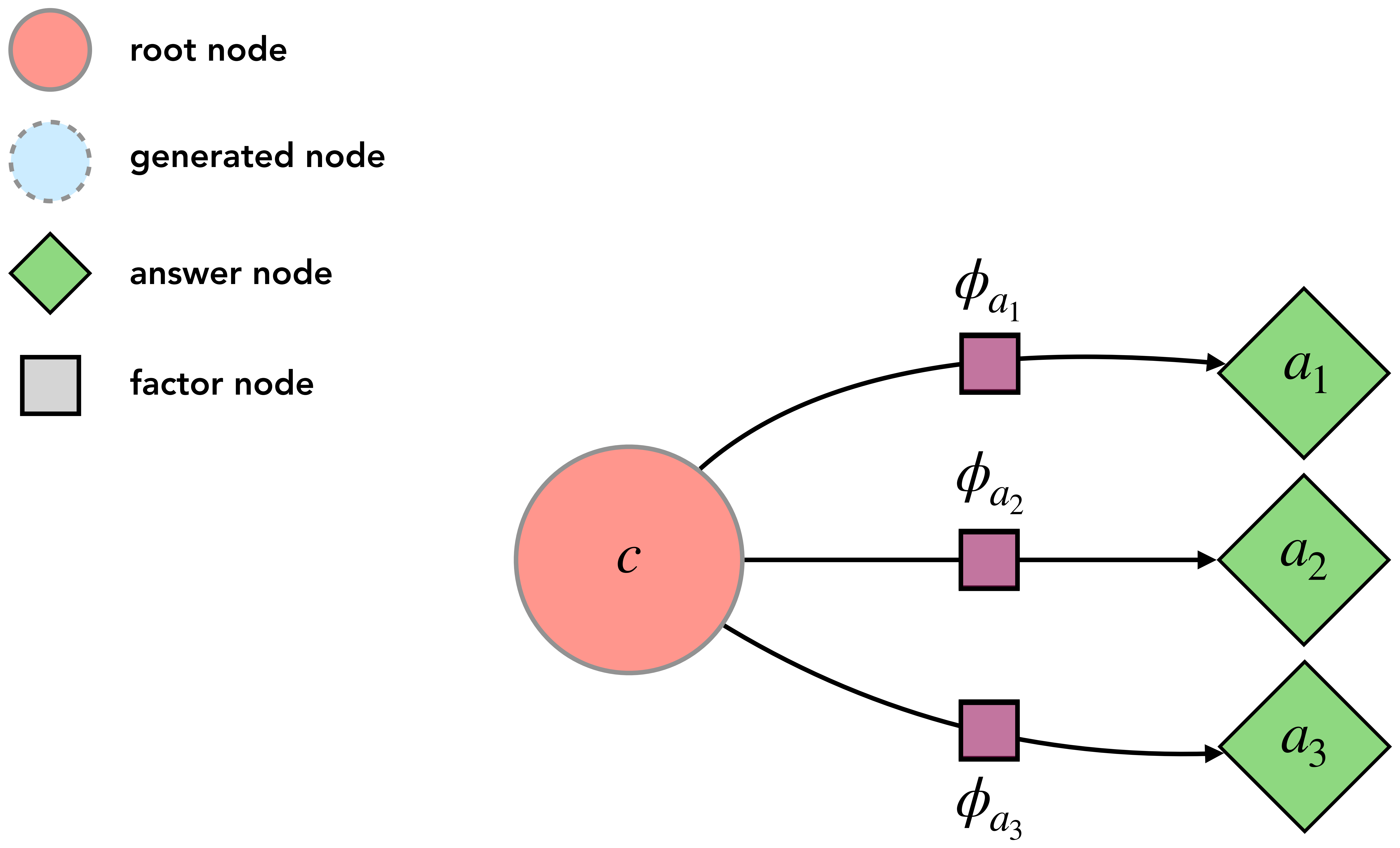
root node
generated node
answer node
factor node
c
g_1
g_2
a_1
a_2
a_3
ϕ_{g_1}
ϕ_{g_2}
$\phi_{g_1 a_1}$
$\phi_{g_2 a_1}$
Score the path
to the answer:
$\gamma_g \phi_g + \gamma_{ga} \phi_{ga}$



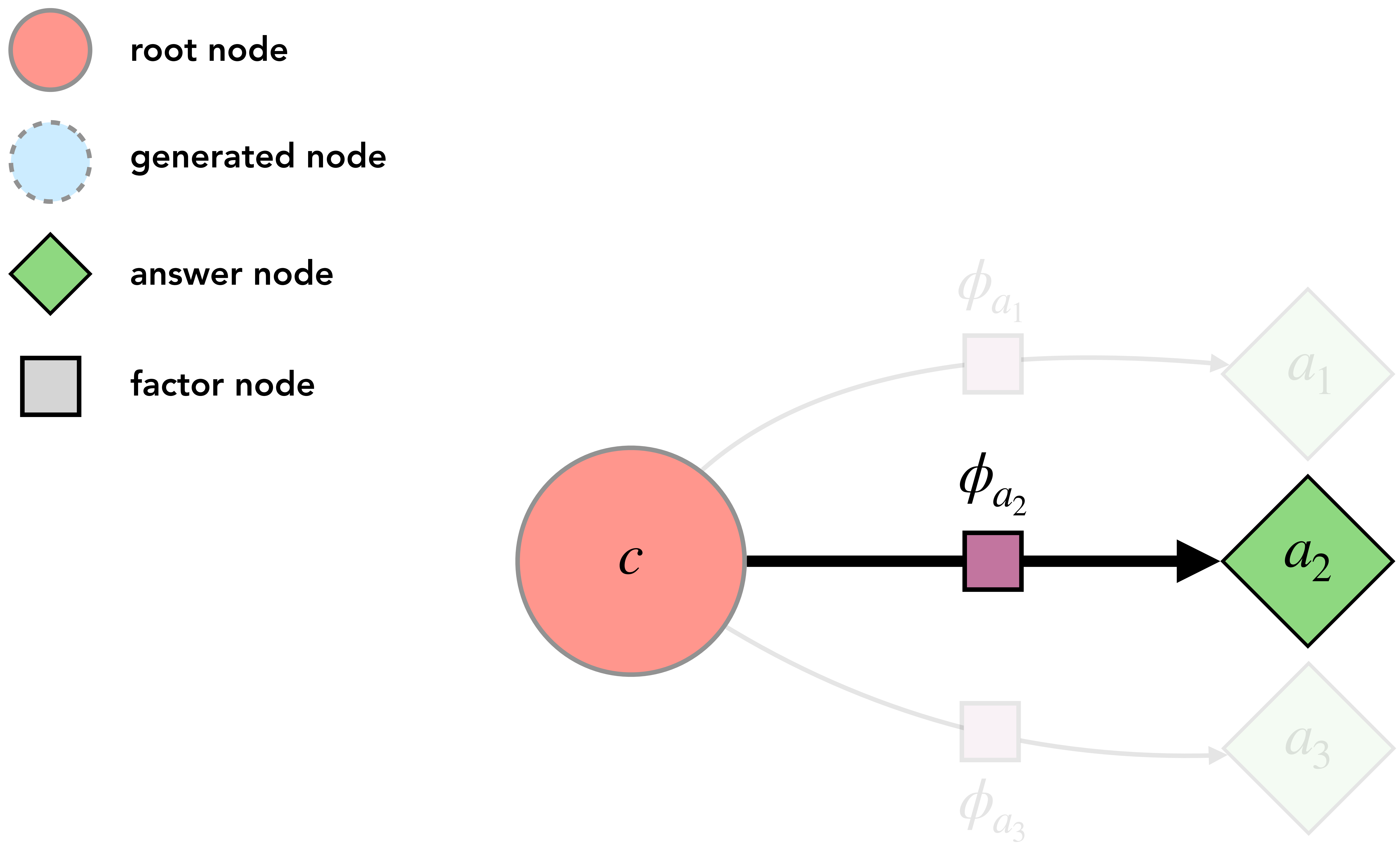
root node
generated node
answer node
factor node
c
g_1
a_1
a_2
a_3
a_1
a_3
ϕ_{a_1}
The **answer score** is the the maximum scoring path to the answer:
$\phi_a = \max_{g \in \mathcal{G}} \gamma_g \phi_g + \gamma_{ga} \phi_{ga}$


root node
generated node
answer node
factor node
ϕ_{a_2}
a_1
a_2
a_3
c
g_2
a_1
a_2
a_3
The **answer score** is the
the maximum scoring
path to the answer:
$\phi_a = \max_{g \in \mathcal{G}} \gamma_g \phi_g + \gamma_{ga} \phi_{ga}$


root node
generated node
answer node
factor node
a_1
a_2
g_1
a_3
ϕ_{a_3}
c
a_1
a_2
a_3
The **answer score** is the the maximum scoring path to the answer:
$$\phi_a = \max_{g \in \mathcal{G}} \gamma_g \phi_g + \gamma_{ga} \phi_{ga}$$



root node
generated node
answer node
factor node
ϕ_{a_1}
ϕ_{a_2}
ϕ_{a_3}
c
a_1
a_2
a_3



$$\hat{a} = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, \phi_a$$

Understanding Social Situations