Globally Normalized Reader

Jonathan Raiman and John Miller

Task: Extractive QA (SQuAD)

Who was the first to recognize that the Analytical Engine had applications beyond pure calculation?

bage KH FRS (/ˈbæbɪdʒ/; 26 December 1791 – 18 October
n English polymath.[1] A mathematician, philosopher,
mechanical engineer, Babbage originated the concept
rogrammable computer.[2]
by some to be a "father of the computer",[2][3][4][5]
credited with inventing the first mechanical computer
lly led to more complex electronic designs, though all
l ideas of modern computers are to be found in
nalytical engine.[2][6] His varied work in other fields has
be described as "pre-eminent" among the many
f his century.[1]
bage's incomplete mechanisms are on display in the
eum in London. In 1991, a functioning difference engine
cted from Babbage's original plans. Built to tolerances
n the 19th century, the success of the finished engine
at Babbage's machine would have worked.
t]
birthplace is disputed, but according to the Oxford
f National Biography he was most likely born at 44
Walworth Road, London, England.[7] A blue plaque on
of Larcom Street and Walworth Road commemorates

birth was given in his obituary in The Times as 26
792; but then a nephew wrote to say that Babbage was
ear earlier, in 1791. The parish register of St. Mary's
London, shows that Babbage was baptised on 6 January
ting a birth year of 1791.[9][10][11]

Ada Lovelace (née Byron; 10 December 1815 – 27 November 1852) was an English mathematician and writer, chiefly known for her work on Charles Babbage's proposed mechanical general-purpose computer, the Analytical Engine. She was the first to recognise that the machine had applications beyond pure calculation, and created the first algorithm intended to be carried out by such a machine. As a result, she is often regarded as the first to recognise the full potential of a "computing machine" and the first computer programmer.
Ada Lovelace was the only legitimate child of the poet Lord Byron, and his wife Anne Isabella Milbanke ("Annabella"), Lady Wentworth. All of Byron's other children were born out of wedlock to other women. Byron separated from his wife a month after Ada was born and left England forever four months later, eventually dying of disease in the Greek War of Independence when Ada was eight years old. Her mother remained bitter towards Lord Byron and promoted Ada's interest in mathematics and logic in an effort to prevent her from developing what she saw as the insanity seen in her father, but Ada remained interested in him despite this (and was, upon her eventual death, buried next to him at her request). Often ill, she spent most of her childhood sick. Ada married William King in 1835. King was made Earl of Lovelace in 1838, and she became Countess of Lovelace.
Her educational and social exploits brought her into contact with scientists such as Andrew Crosse, Sir David Brewster, Charles Wheatstone, Michael Faraday and the author Charles Dickens, which she used to further her education. Ada described her approach as...

A computer is a device that can be instructed to carry
sequences of arithmetic or logical operations autom
ability of computers to follow generalized sets of opera
programs, enables them to perform an extremely wi
tasks.
Such computers are used as control systems for a very
of industrial and consumer devices. This includes si
purpose devices like microwave ovens and remote con
devices such as industrial robots and computer assiste
also in general purpose devices like personal computer
devices such as smartphones. The Internet is run on co
it connects millions of other computers.
Since ancient times, simple manual devices like the a
people in doing calculations. Early in the Industrial Revo
mechanical devices were built to automate long tediou
as guiding patterns for looms. More sophisticate
machines did specialized analog calculations in the
century. The first digital electronic calculating ma
developed during World War II. The speed, power, and
computers has increased continuously and dramatically
Conventionally, a modern computer consists of a
processing element, typically a central processing un
some form of memory. The processing element
arithmetic and logical operations, and a sequencing and
can change the order of operations in respons
information. Peripheral devices include input devices
mice, joystick, etc.), output devices (monitor screens, pr

~300 words

Goal

Who was the first to recognize that the Analytical Engine had applications beyond pure calculation?

millions of documents

bage KH FRS (/ˈbæbɪdʒ/; 26 December 1791 – 18 October
n English polymath.[1] A mathematician, philosopher,
mechanical engineer, Babbage originated the concept
rogrammable computer.[2]
by some to be a "father of the computer",[2][3][4][5]
credited with inventing the first mechanical computer
lly led to more complex electronic designs, though all
al ideas of modern computers are to be found in
nalytical engine.[2][6] His varied work in other fields has
be described as "pre-eminent" among the many
f his century.[1]
bage's incomplete mechanisms are on display in the
eum in London. In 1991, a functioning difference engine
cted from Babbage's original plans. Built to tolerances
n the 19th century, the success of the finished engine
at Babbage's machine would have worked.

Ada Lovelace (née Byron; 10 December 1815 – 27 November 1852) was an English mathematician and writer, chiefly known for her work on Charles Babbage's proposed mechanical general-purpose computer, the Analytical Engine. She was the first to recognise that the machine had applications beyond pure calculation, and created the first algorithm intended to be carried out by such a machine. As a result, she is often regarded as the first to recognise the full potential of a "computing machine" and the first computer programmer.
Ada Lovelace was the only legitimate child of the poet Lord Byron, and his wife Anne Isabella Milbanke ("Annabella"), Lady Wentworth. All of Byron's other children were born out of wedlock to other women. Byron separated from his wife a month after Ada was born and left England forever four months later, eventually dying of disease in the Greek War of Independence when Ada was eight years old. Her mother remained bitter towards Lord Byron and promoted

A computer is a device that can be instructed to carry
sequences of arithmetic or logical operations autom
ability of computers to follow generalized sets of opera
programs, enables them to perform an extremely wi
tasks.
Such computers are used as control systems for a very
of industrial and consumer devices. This includes si
purpose devices like microwave ovens and remote con
devices such as industrial robots and computer assiste
also in general purpose devices like personal computer
devices such as smartphones. The Internet is run on co
it connects millions of other computers.
Since ancient times, simple manual devices like the a
people in doing calculations. Early in the Industrial Revo
mechanical devices were built to automate long tediou
as guiding patterns for looms. More sophisticate

How do we get there?

Related Work:

- Bi-Attention Flow (Seo et al., 2016)
- Dynamic Co-Attention Network (Xiong et al., 2016)
- R-Net (Wang et al., 2017)
- Rasor (Lee et al., 2016)
- Hybrid AoA Reader (2018)

Challenges:

- Bi-directional attention
- Rank all possible spans
- No available data augmentation

Globally Normalized Reader

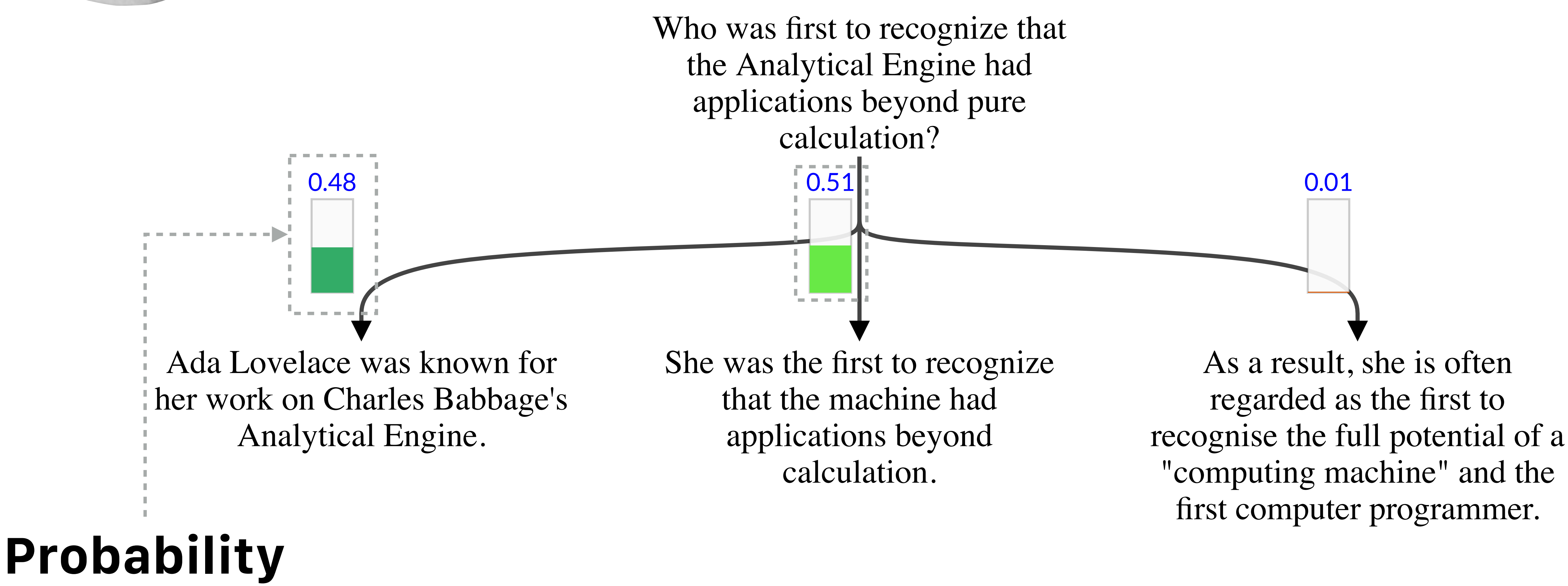
- Factorize search into sentences, span start & end
- Globally Normalize search (Andor et al. @ ACL 2016)
- Beam Search during training w/. Early Updates

Contributions:

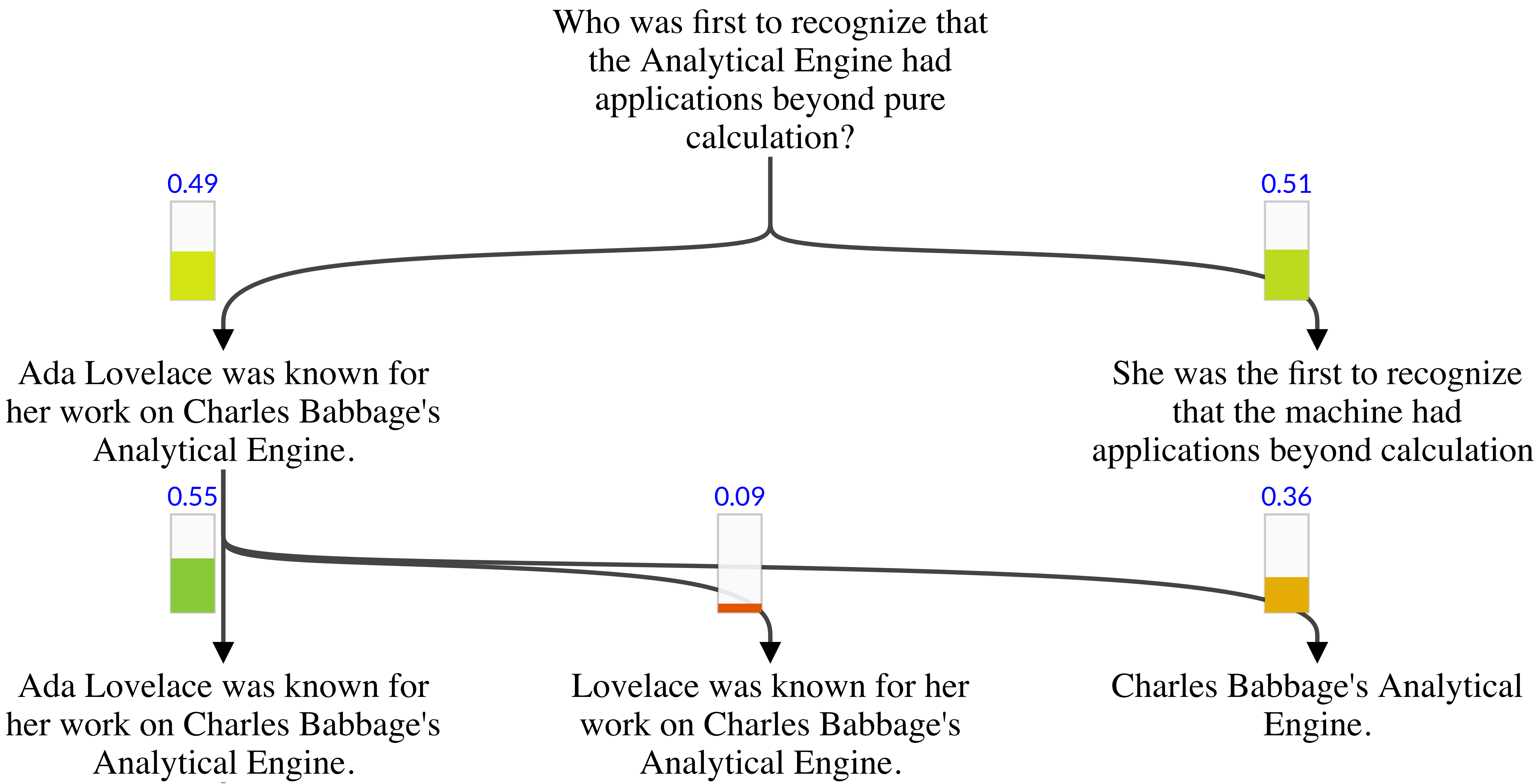
- **Conditional computation** (allocate computation to promising search beams)
- **Quasi-infinite training data** w/. Type Swaps
- **24.7x** speedup over bi-attention-flow
- Dev **68.4 EM, 76.21 F1** w/o bidirectional attention

Example

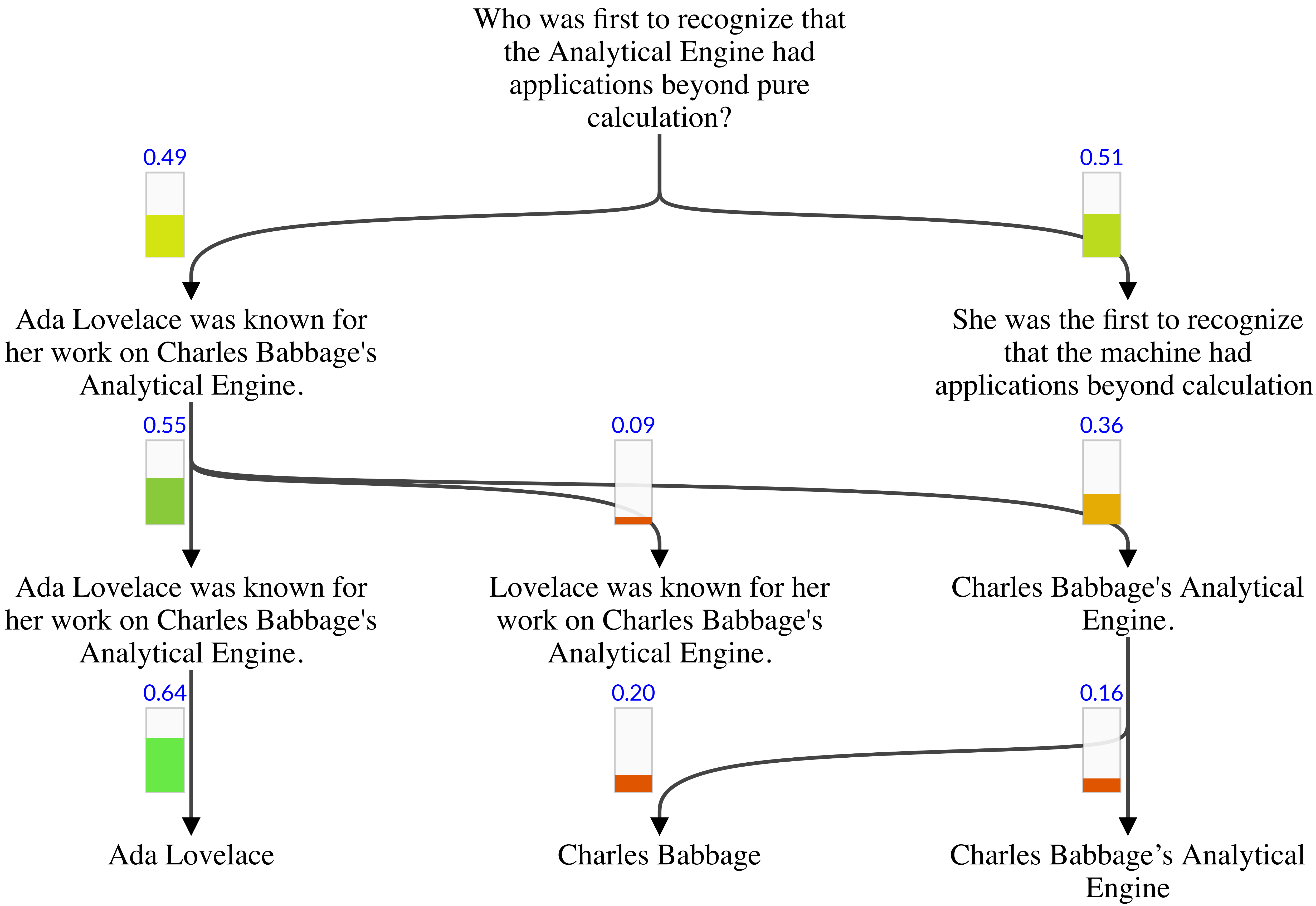
Ada Lovelace was known for her work on Charles Babbage's Analytical Engine. She was the first to recognize that the machine had applications beyond calculation. As a result, she is often regarded as the first to recognise the full potential of a "computing machine" and the first computer programmer.



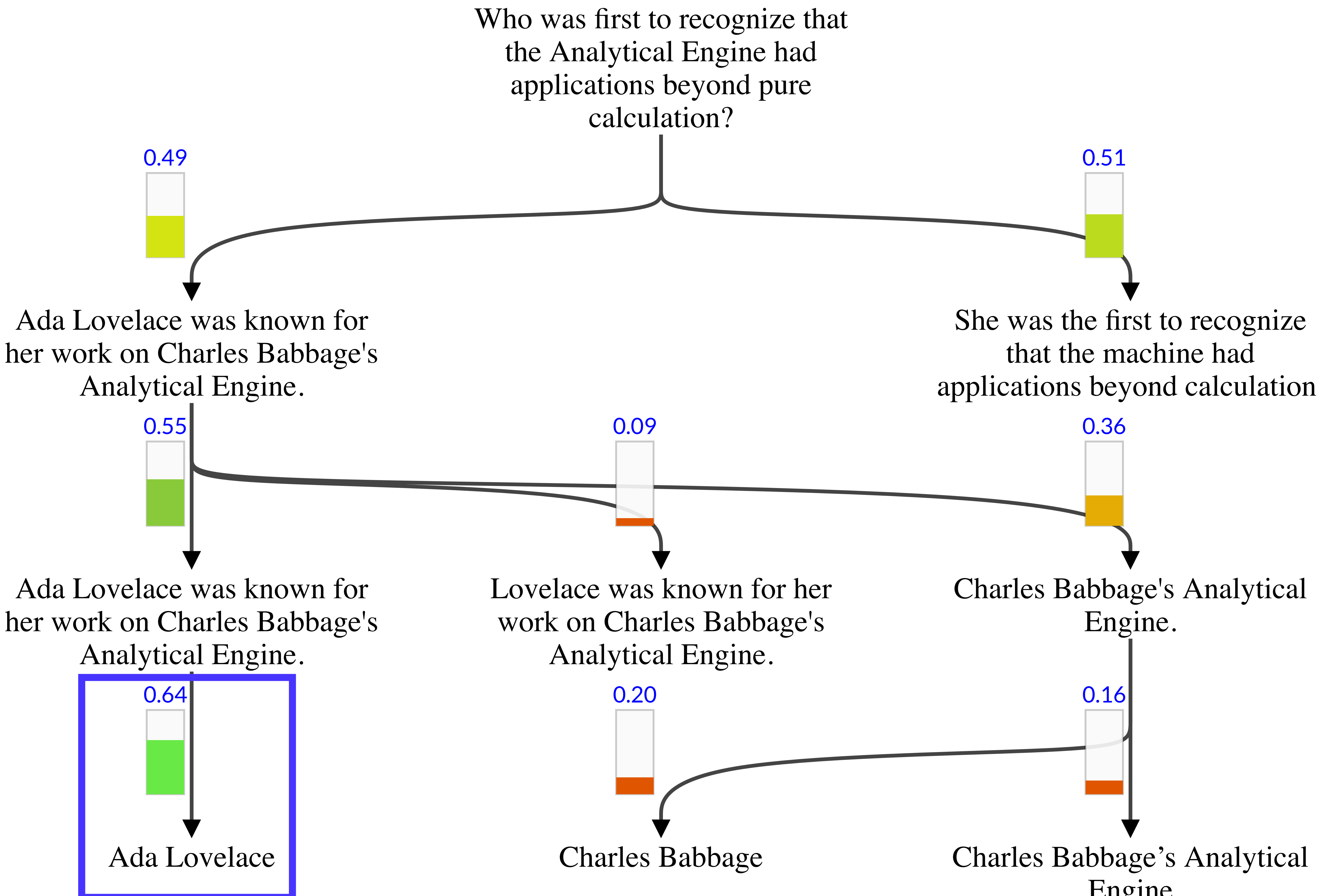
Pick a Sentence



Start word chosen for each sentence



Select end word among remainder

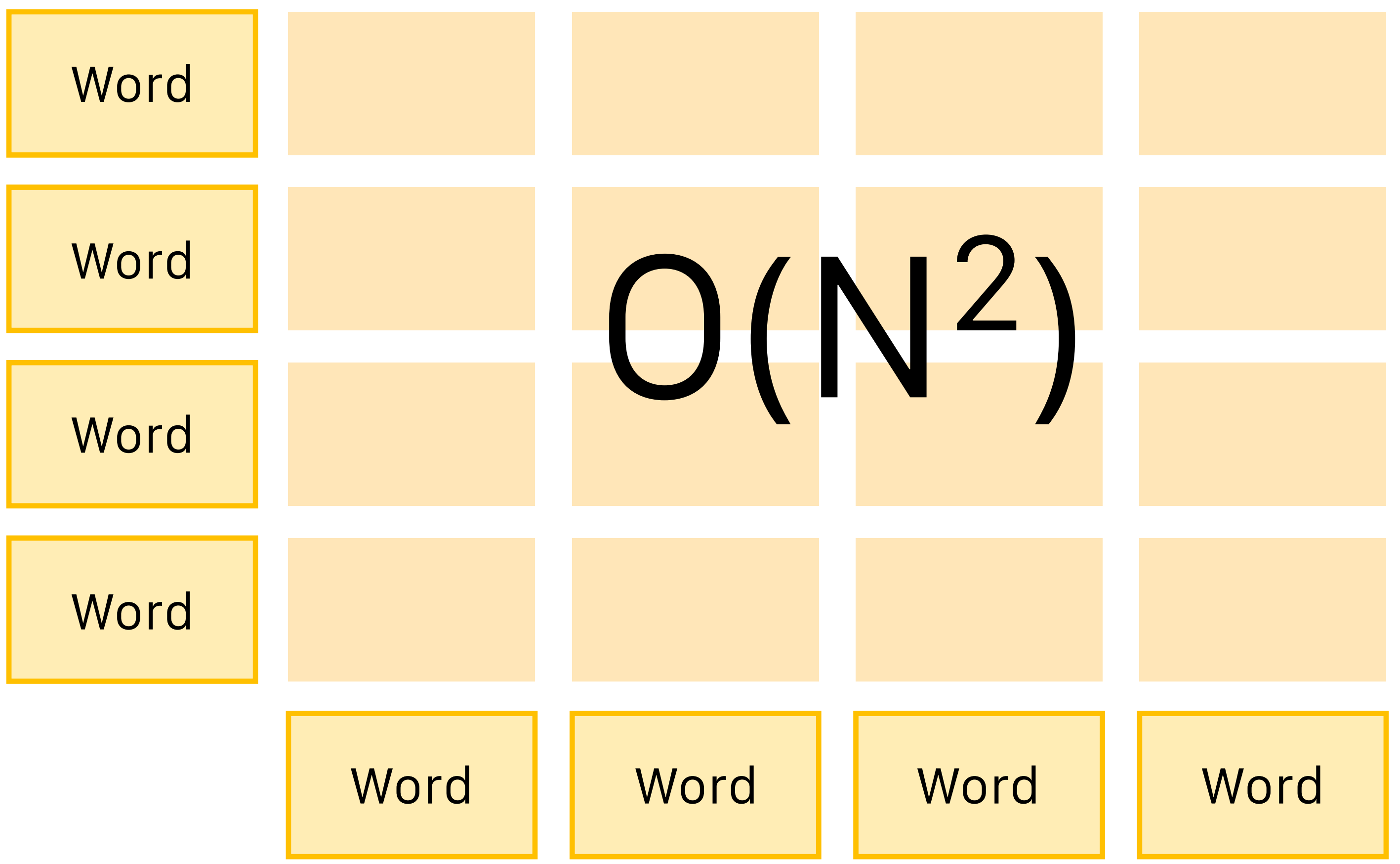


GNR’s Answer: Ada Lovelace

Outline

1. Approach
   1) Challenges
   2) Architecture
   3) Early updates
   4) Conditional Computation
   5) Global Normalization
   6) Type Swaps
2. Results
   1) Comparison
   2) Data Augmentation
   3) Speedup

Span Selection/Attention



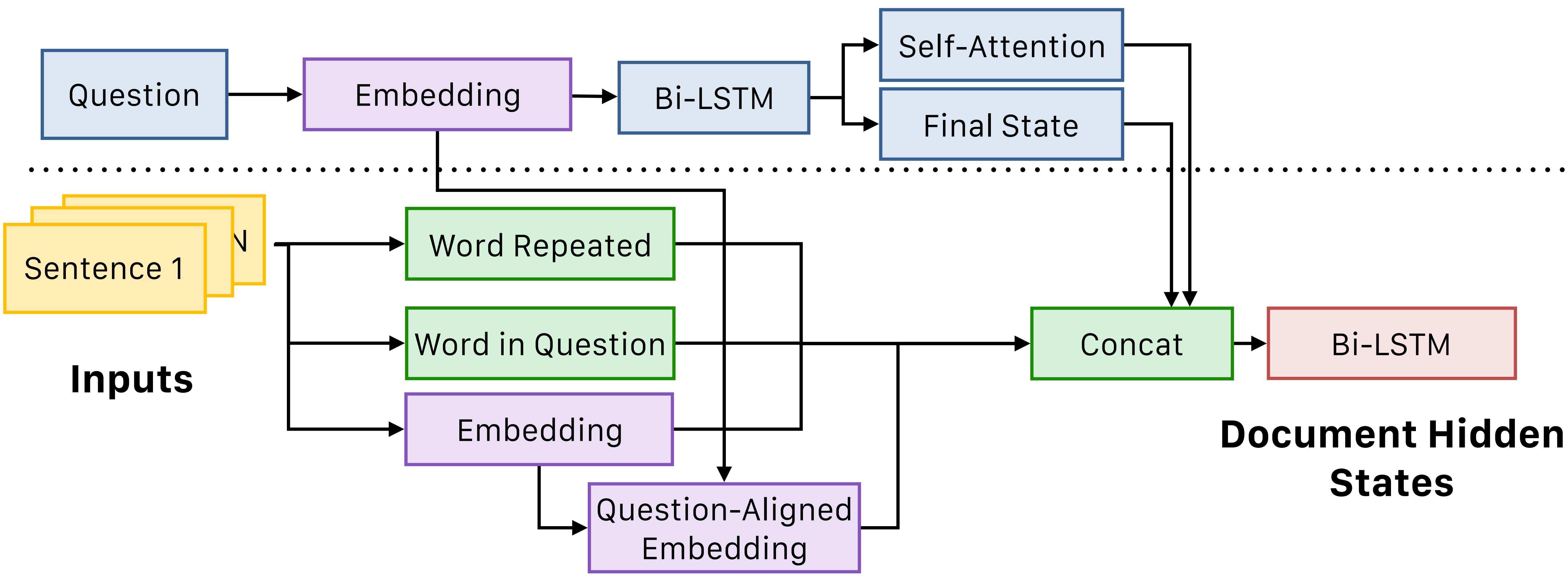
Overfitting

- 100k QA Pairs
- Dropout
- Weight Decay
- Tuning
- Pretrained word vectors
- Ensembles
- Label bias

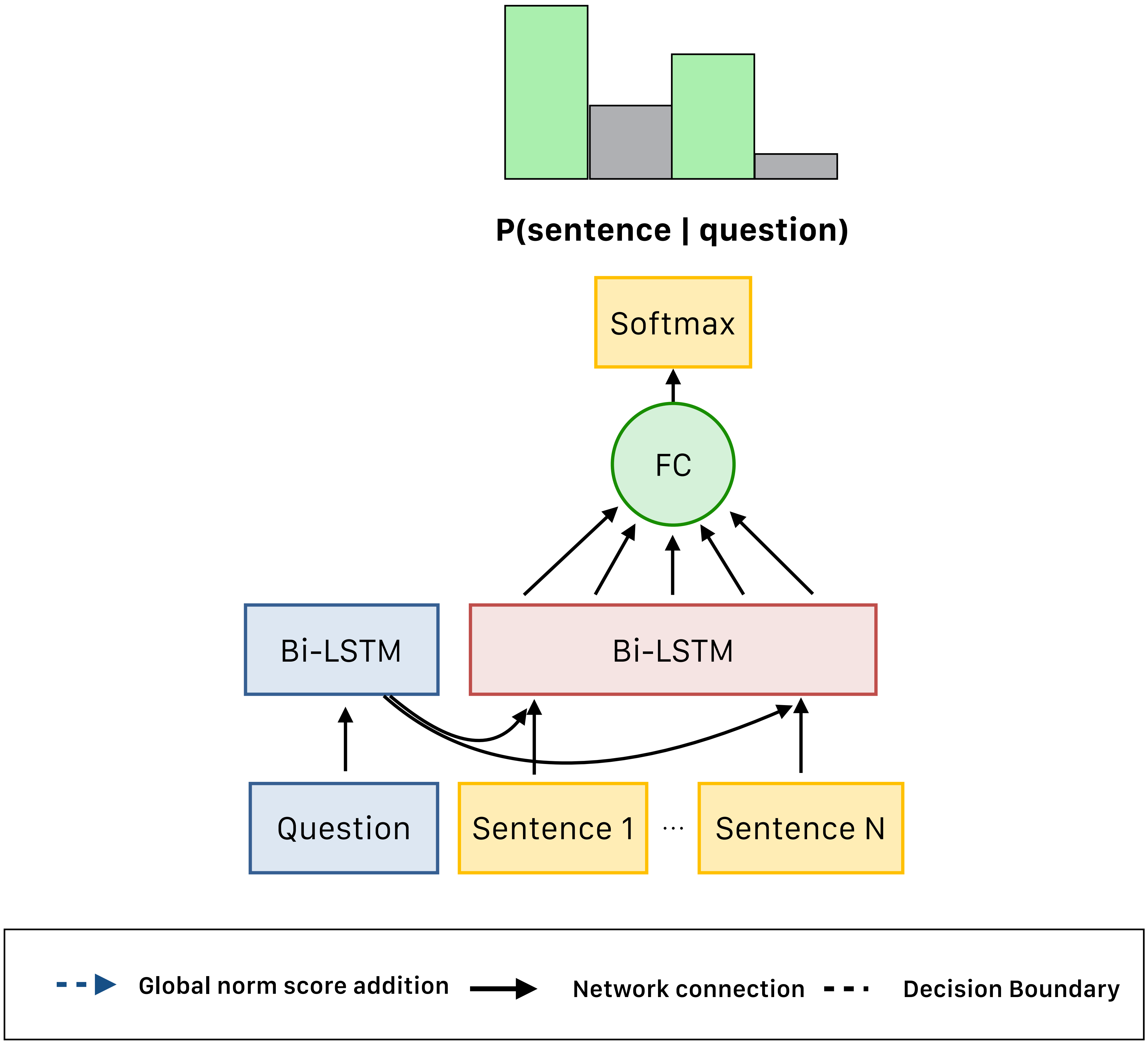
Approach

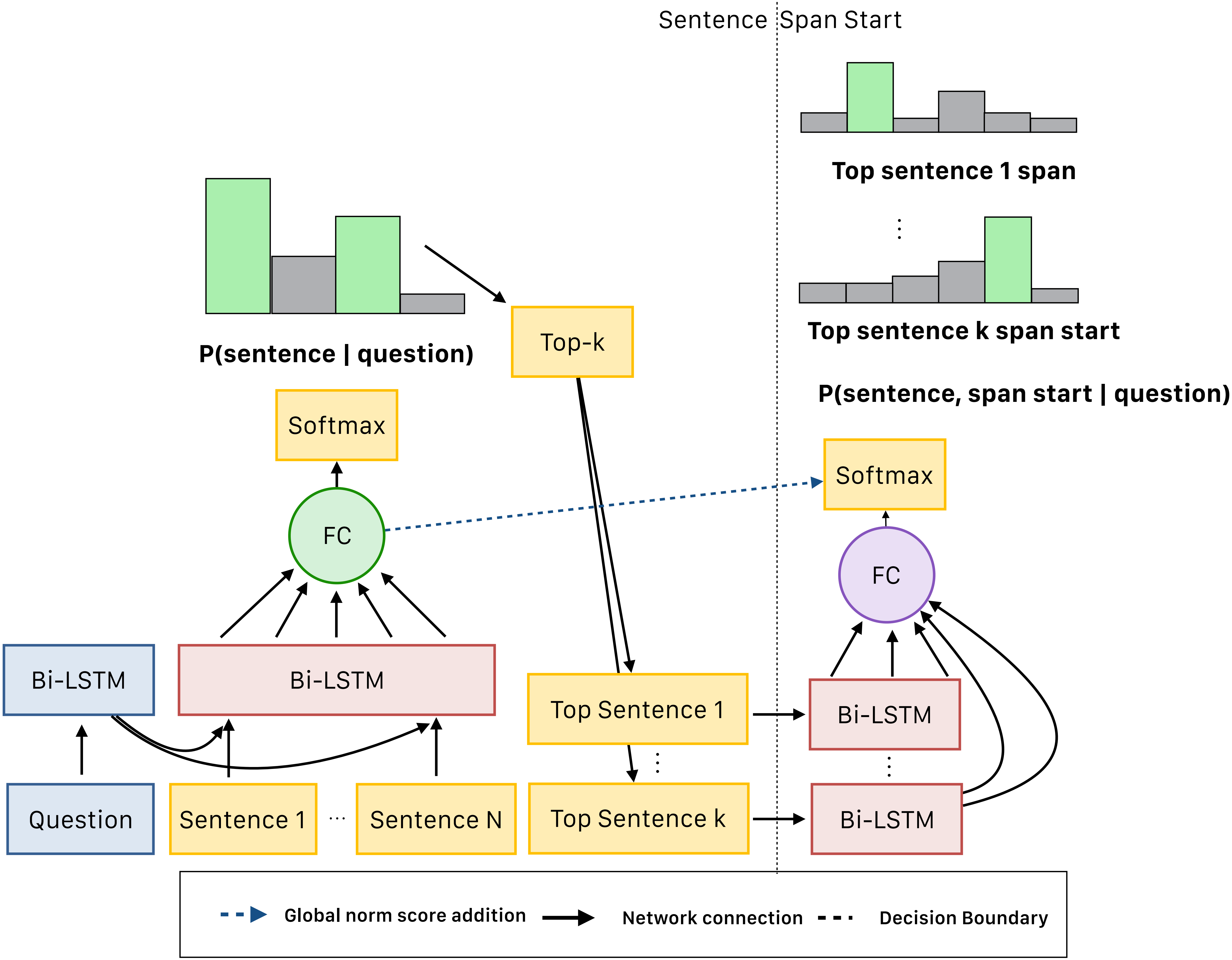
- Search to shrink candidate space & scope of attention
- Data augmentation

Architecture:
Question-Aware Document Encoding

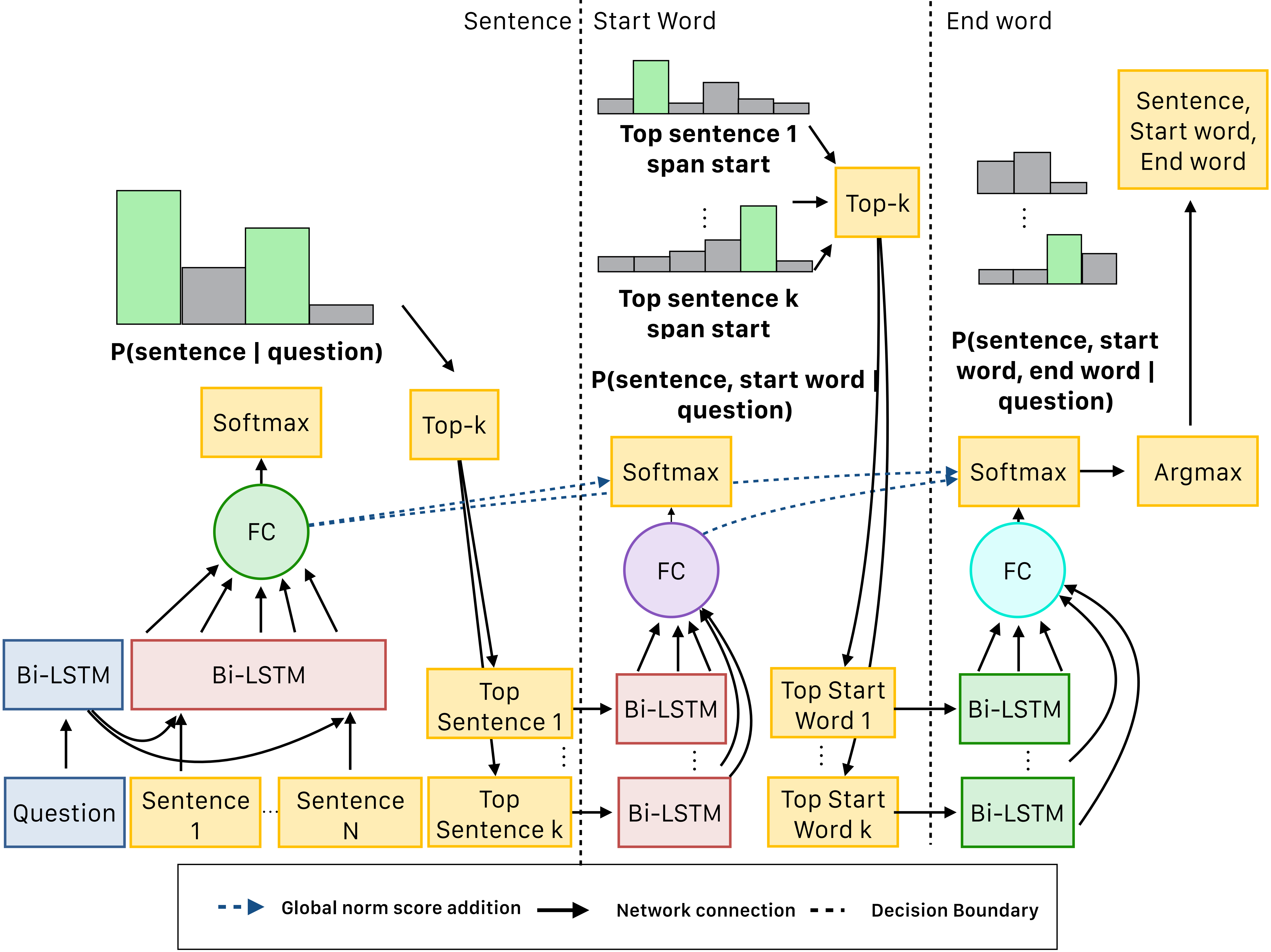


Architecture



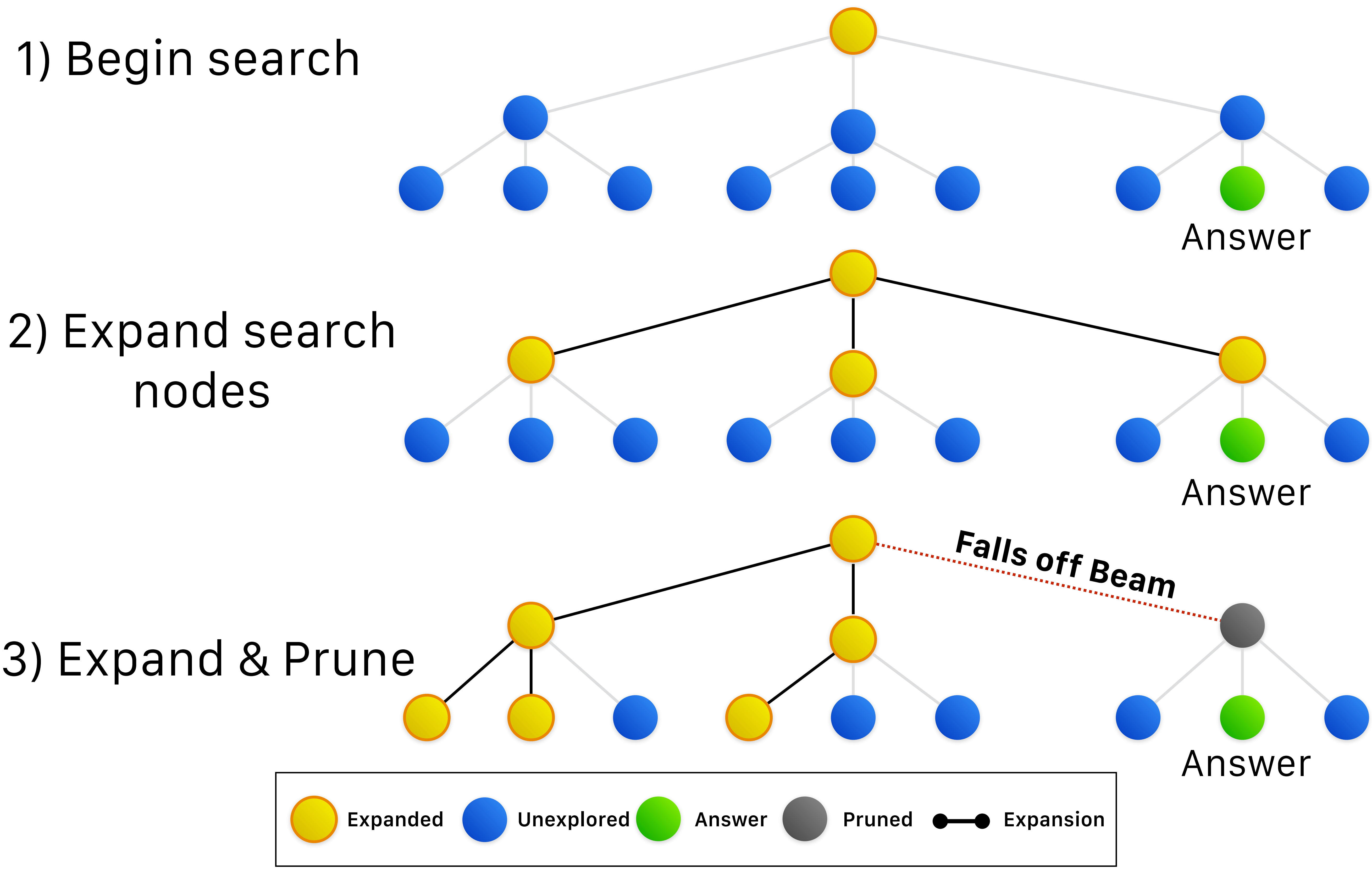


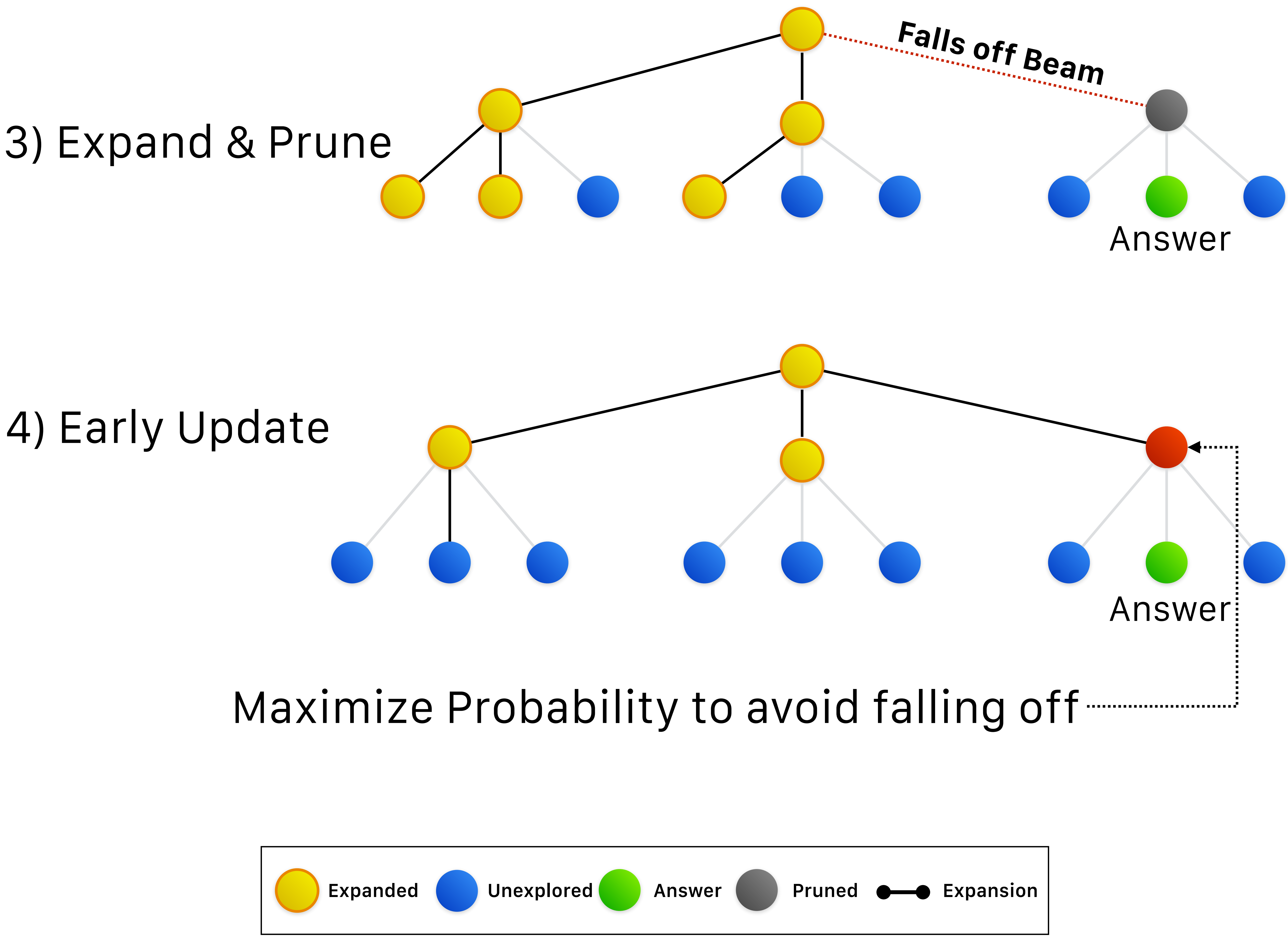
Sentence
Span Start
Top sentence 1 span
Top sentence k span start
P(sentence | question)
Top-k
P(sentence, span start | question)
Softmax
Softmax
FC
FC
Bi-LSTM
Bi-LSTM
Top Sentence 1
Bi-LSTM
Question
Sentence 1
···
Sentence N
Top Sentence k
Bi-LSTM
Global norm score addition
Network connection
Decision Boundary



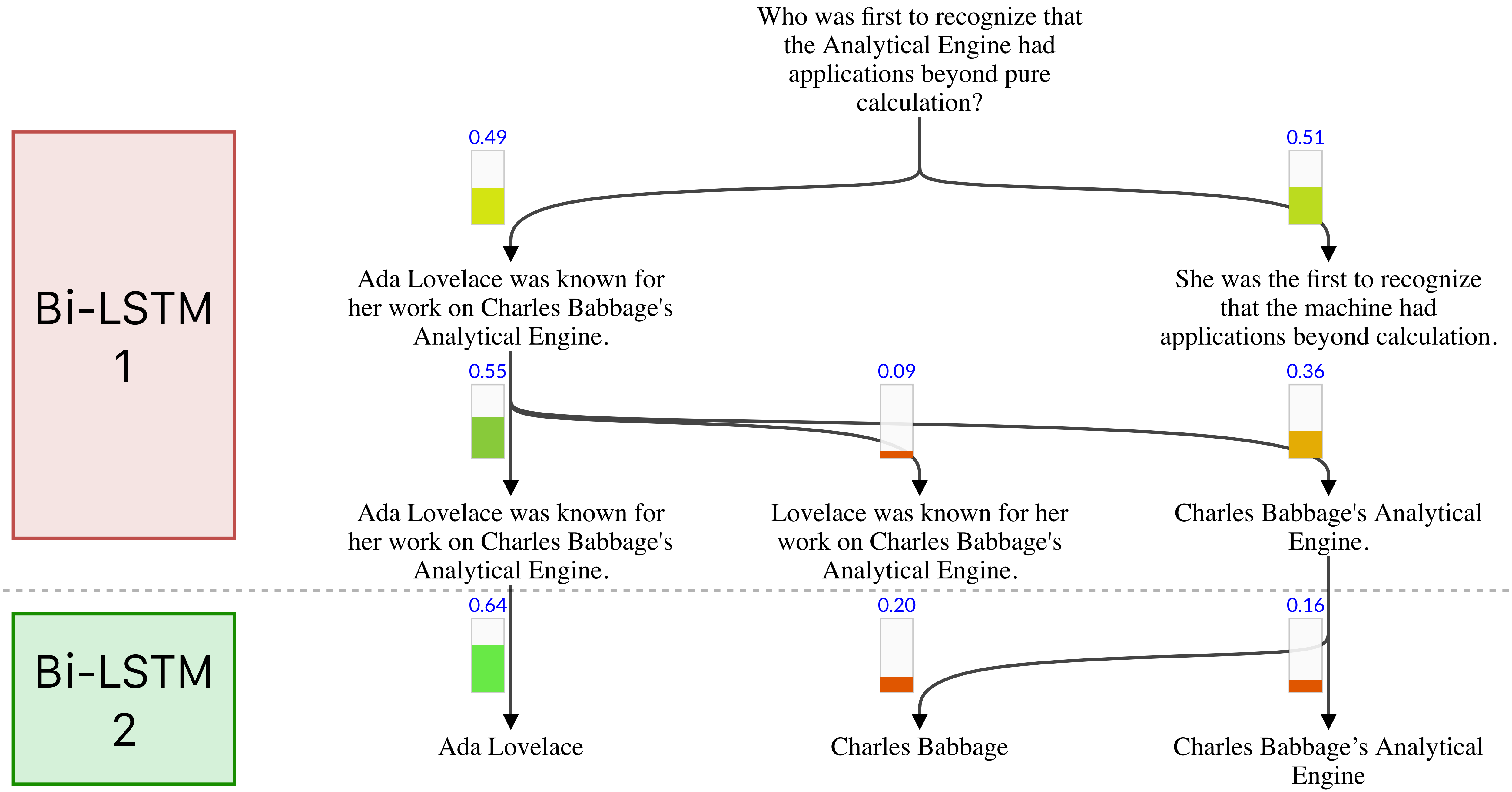
Sentence
Start Word
End word
Top sentence 1 span start
Top sentence k span start
Top-k
Sentence, Start word, End word
P(sentence | question)
P(sentence, start word | question)
P(sentence, start word, end word | question)
Softmax
Top-k
Softmax
Softmax
Argmax
FC
FC
FC
Bi-LSTM
Bi-LSTM
Top Sentence 1
Bi-LSTM
Top Start Word 1
Bi-LSTM
Question
Sentence 1
Sentence N
Top Sentence k
Bi-LSTM
Top Start Word k
Bi-LSTM
Global norm score addition
Network connection
Decision Boundary

Early Updates




3) Expand & Prune
Falls off Beam
Answer
4) Early Update
Answer
Maximize Probability to avoid falling off
Expanded
Unexplored
Answer
Pruned
Expansion

Conditional Computation



Sentence prediction accuracy 88-89%: can focus computation on subset!

Local Normalization

$$\begin{aligned}\mathbb{P}(a|d,q) &= \mathbb{P}_{\text{sent}}(i|d,q)\cdot\mathbb{P}_{\text{sw}}(j|i,d,q)\cdot\mathbb{P}_{\text{ew}}(k|j,i,d,q)\\ &= \frac{\exp(\phi_{\text{sent}}(i,d,q))}{Z_{\text{sent}(d,q)}}\cdot\frac{\exp(\phi_{\text{sw}}(j,i,d,q))}{Z_{\text{sw}(i,d,q)}}\cdot\frac{\exp(\phi_{\text{ew}}(k,j,i,d,q))}{Z_{\text{ew}(j,i,d,q)}}\end{aligned}$$

$$\begin{aligned} a &= \text{answer}\\ d &= \text{document}\\ q &= \text{question}\\ i &= \text{sentence}\\ j &= \text{start word}\\ k &= \text{end word}\\ \phi(\cdot) &= \text{Score function}\\ Z &= \text{Partition function}\end{aligned}$$

Global Normalization

$$\text{score}(a, d, q) = \phi_{\text{sent}}(d_i) + \phi_{\text{sw}}(d_{i,j}) + \phi_{\text{ew}}(d_{i,j:k})$$

$$\mathbb{P}(a \mid d, q) = \frac{\exp(\text{score}(a, d, q))}{Z}$$

$$Z = \sum_{a' \in \mathcal{A}(d)} \exp(\text{score}(a', d, q))$$

$a =$ answer

$d =$ document

$q =$ question

$i =$ sentence

$j =$ start word

$k =$ end word

$\phi(\cdot) =$ Score function

$Z =$ Partition function

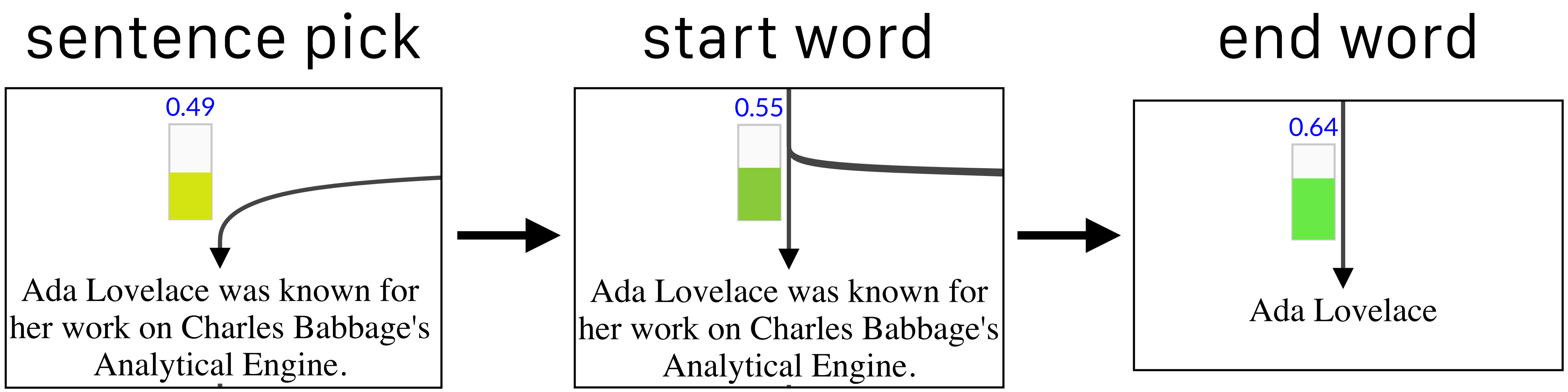
Set grows exponentially.
Approximate using beam search

$\mathcal{A}(d) =$ Set of all possible answer spans

$d_{i,j:k} =$ span from word j to k, in sentence i

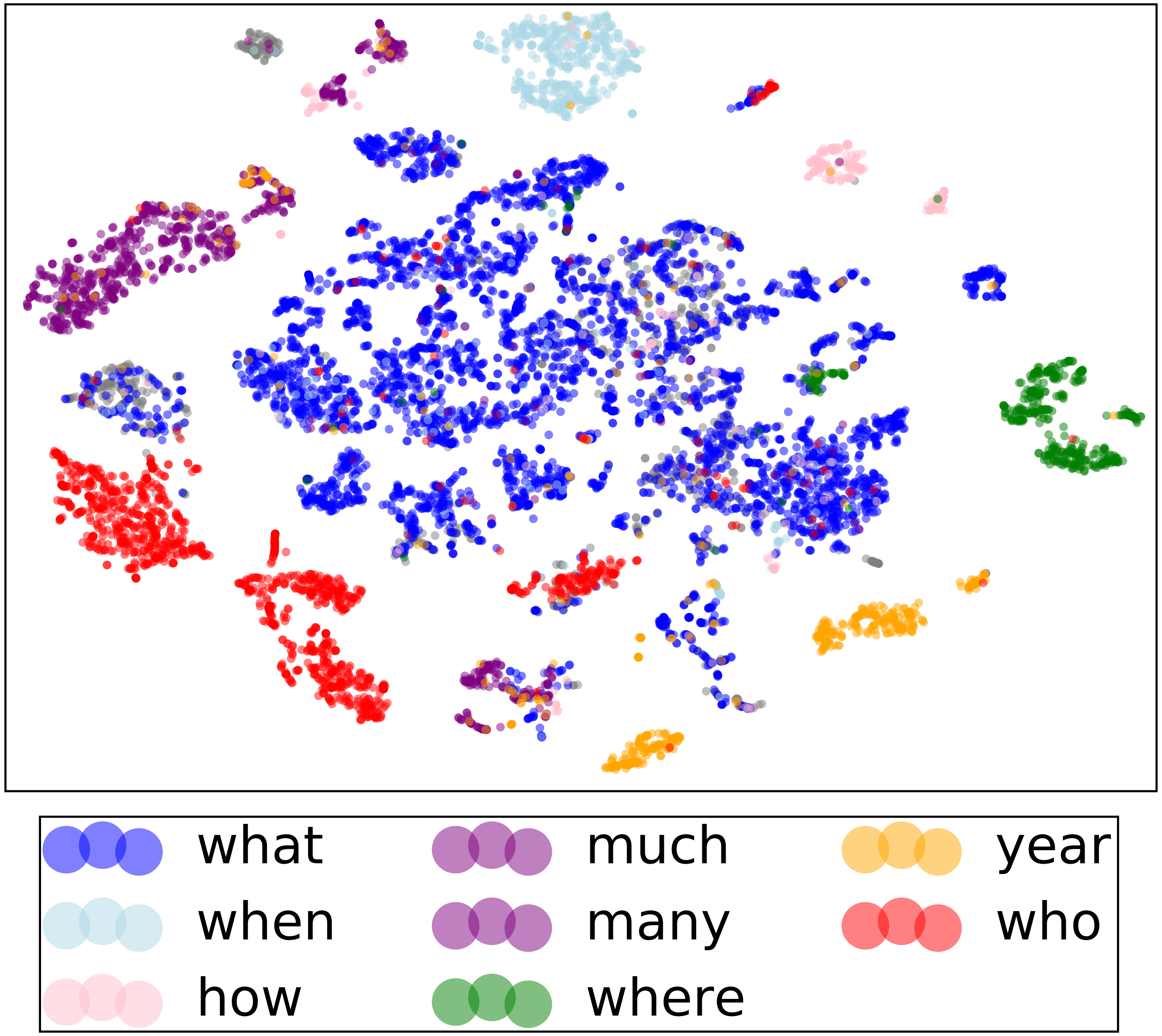
Global Normalization

Answer probability grows as search advances (we are not multiplying probabilities!)



Note: globally normalized models remain undecided until the end word, while local models usually have spiked distributions

Type Swaps



T-SNE Question hidden state naturally clusters according to question type.

How can we exploit this?

Type Swaps

- Common SQuAD pitfall: pick wrong answer with right type (human, organization, etc.)
- Solution: increase typed-based QA pairs

Who said in December 2012 that the fight would change from military to law enforcement?

. . . Basic objectives of the Bush Administration "war on terror", such as targeting al Qaeda and building international counterterrorism alliances, remain in place. In December 2012, Jeh Johnson, the General Counsel of the Department of Defense stated that the military fight will be replaced by a law enforcement operation when speaking at Oxford University . . .

Answer: Jeh Johnson

Type Swaps

- Common SQuAD pitfall: pick wrong answer with right type (human, organization, etc.)
- Solution: increase typed-based QA pairs

Who said in April 25, 2011 that the fight would change from military to law enforcement?

. . . Basic objectives of the Cabinet of Japan "war on terror", such as targeting al Qaeda and building international counterterrorism alliances, remain in place. In April 25, 2011, Sheryl Sandberg, the General Counsel of the ministry of education stated that the military fight will be replaced by a law enforcement operation when speaking at Ain Shams University. . .

Answer: Sheryl Sandberg

Type Swaps

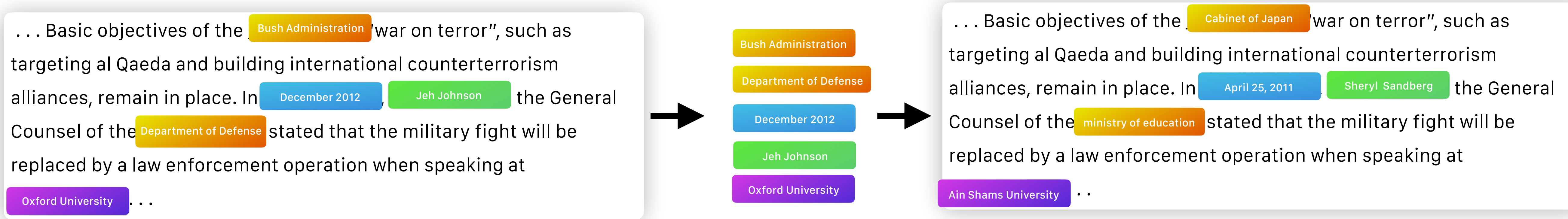
- Common SQuAD pitfall: pick wrong answer with right type (human, organization, etc.)
- Solution: increase typed-based QA pairs

Who said in 2012 that the fight would change from military to law enforcement?

. . . Basic objectives of the British Empire "war on terror", such as targeting al Qaeda and building international counterterrorism alliances, remain in place. In 2012, Genghis Khan, the General Counsel of the EMNLP stated that the military fight will be replaced by a law enforcement operation when speaking at George Washington University . . .

Answer: Genghis Khan

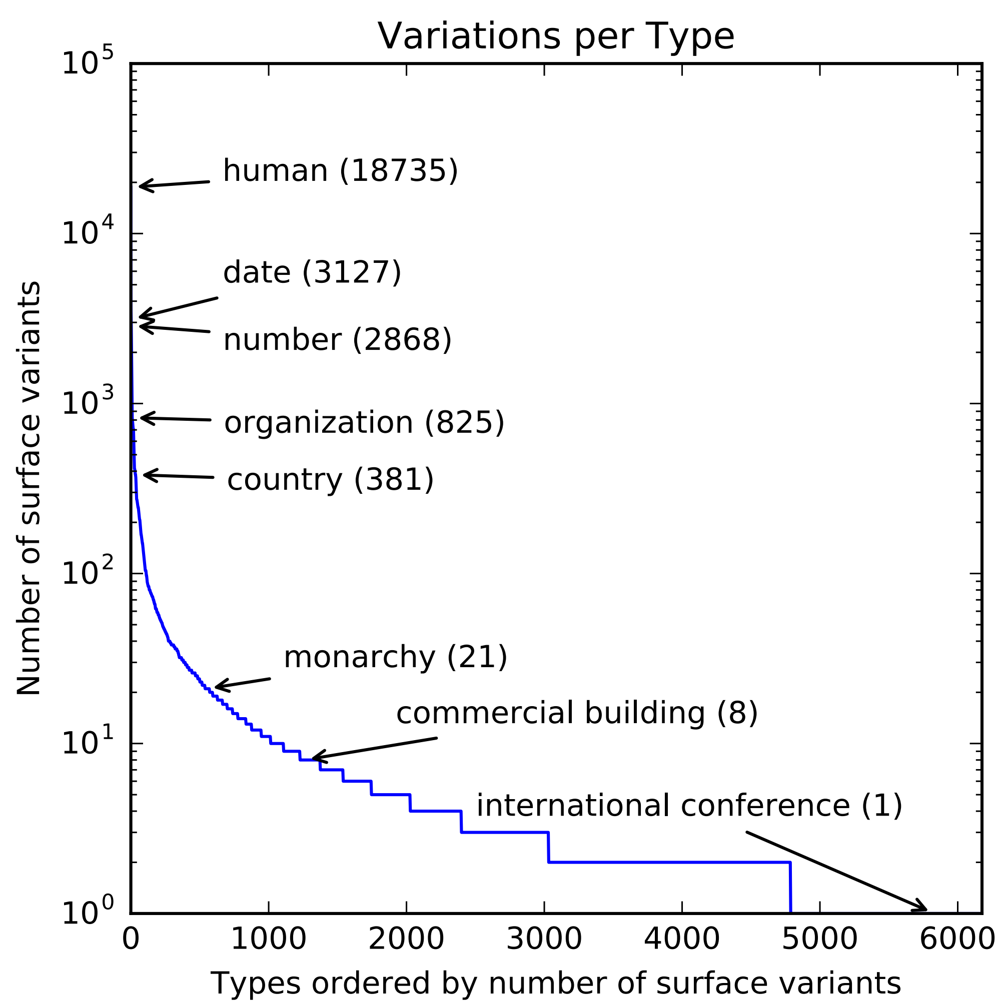
Type Swaps



1) Extract Entities

2) Assign Types

3) Swap with same type variations



$$2.92 \cdot 10^{369}$$

unique documents

Experiments

- Evaluate GNR against baselines on SQuAD dev set (100,000 QA pairs)[1]
- GNR Ablations:
    - Data Augmentation
    - Global Normalization
- Measure Speedup

[1] https://rajpurkar.github.io/SQuAD-explorer/

Comparison

Model	EM	F1
GNR	68.4	76.2
Bi-Attention-Flow (Seo et al., 2016)	67.7	77.3
Rasor (Lee et al., 2016)	66.4	74.9
DCN (Xiong et al., 2016)	65.4	75.6
FastQA (Weissenborn et al., 2017)	67.8	76.3
R-Net (Wang et al., 2017)	**72.3**	**80.6**

Model	EM	F1
GNR	68.4	76.2
GNR w/o Global Norm	67.21	76.0
GNR w/o Type Swaps	66.6	75.0

Data Augmentation

Impact of number of augmented examples:

Number of Swaps	EM	F1
0	66.6	75.0
1000	66.9	75,0
10 000	**68.4**	**76.21**
50 000	66.8	75.3
100 000	66.1	74.3

Impact of Type Swaps on the DCN+[1]:

Number of Swaps	Train F1	Dev F1
0	81.3	78.1
50 000	72.5	**78.2**

[1] **Updated DCN model, see https://rajpurkar.github.io/SQuAD-explorer/**

Speedup

- Full dev set, batch size 32, average 5 runs, on Titan X:
  - Bi-Attention-Flow[1]: 1260.23 ±17.26 seconds
  - GNR: **51.58 ±0.266 seconds**
- Key reasons:
  - Efficient batching of Beam Search
  - Only rank subset of spans
  - Factorize search with document structure

[1] github.com/allenai/bi-att-flow

Conclusion

Key contributions:

- Learning-to-Search w/. early updates & global norm enables conditional computation.
- Data augmentation that improves performance
- 24.7x speedup over bi-attention-flow
- Achieve >= results than bi-directional attention

Future Work

- Conditional computation for generative models & large search spaces
- Program induction/search with perfect simulator
- Model amplification (AlphaZero-style)
- Type Swaps on other NLP tasks? Grammar-aware type-swaps? Adversarial Type Swaps?

Code & Dataset:
github.com/baidu-research/GloballyNormalizedReader

Thank You!

DeepType

Multilingual Entity Linking by Neural Type System Evolution

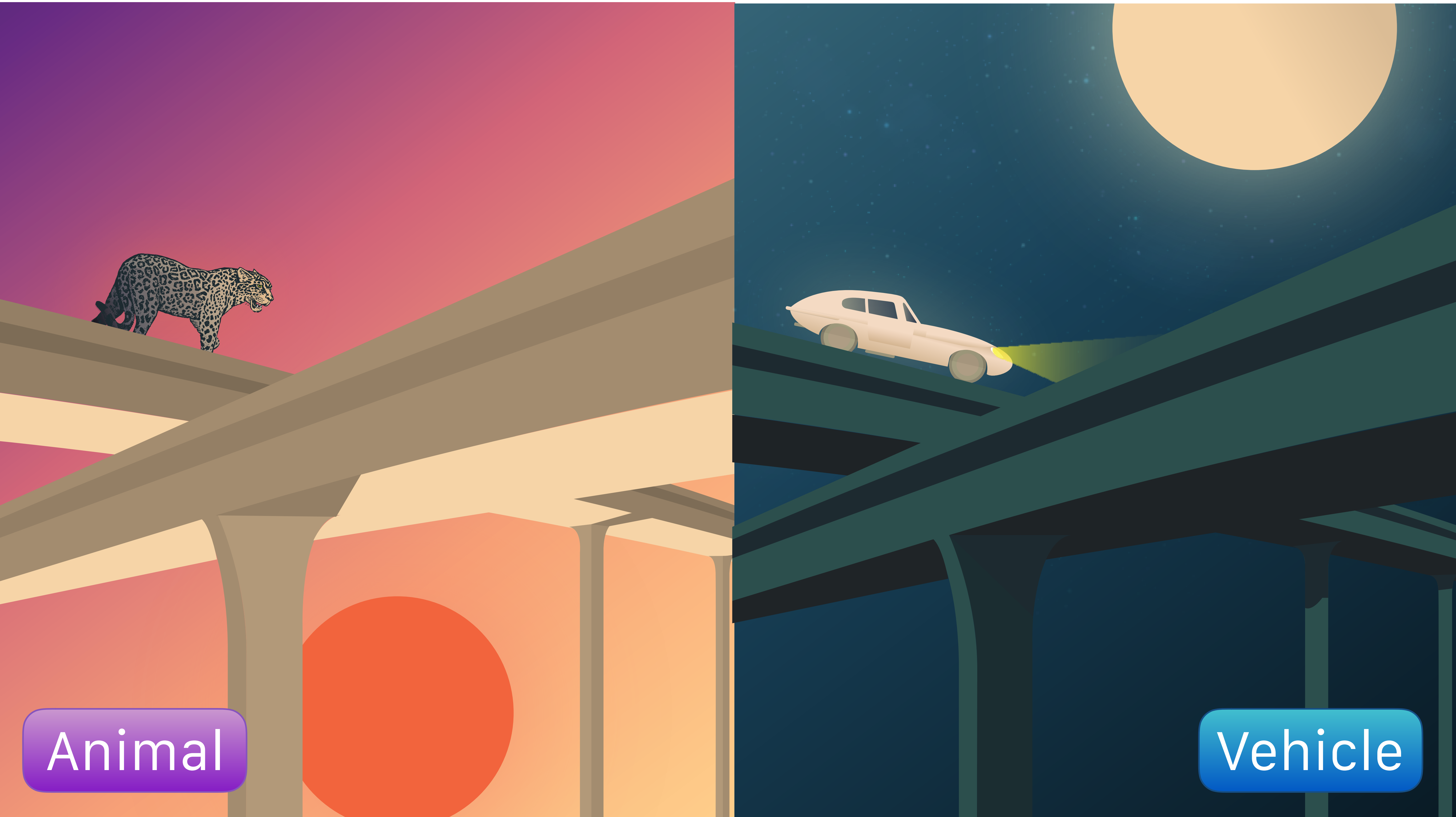
Jonathan Raiman & **Olivier Raiman**

OpenAI

Agilience

Entity Linking

The man saw a Jaguar speed on the highway.



Entity Linking

The prey saw a Jaguar cross the jungle.

The man saw a Jaguar speed on the highway.

vehicle

With types accuracy reaches **98.6-99%**

(type **oracle** on TAC KBP 2010/CoNLL YAGO)

The prey saw a Jaguar cross the jungle.

animal

Summary

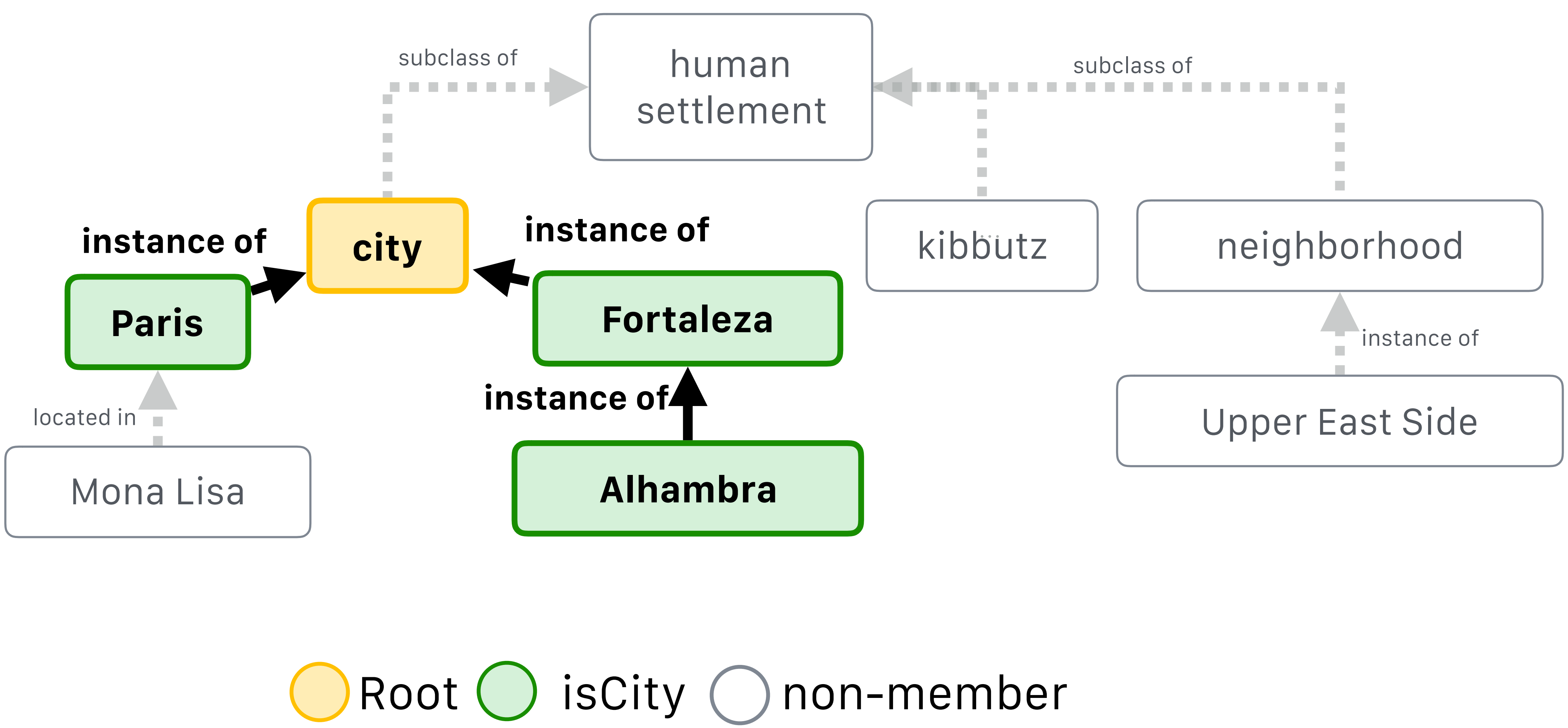
- Design a neural type system
- Results
- Contributions

Type generation

Wikidata is a graph with 40M+ entities

isCity = child(city, instance of)

root relation



Design a Type System

$$\boxed{\max_{\mathcal{A}}} \max_{\theta} S_{\text{model}}(\mathcal{A}, \theta) = \boxed{\frac{\sum_{(m, e_{\text{GT}}, \mathcal{E}_m) \in M} \mathbb{1}_{e_{\text{GT}}}(e^*)}{|M|}}$$

Type System

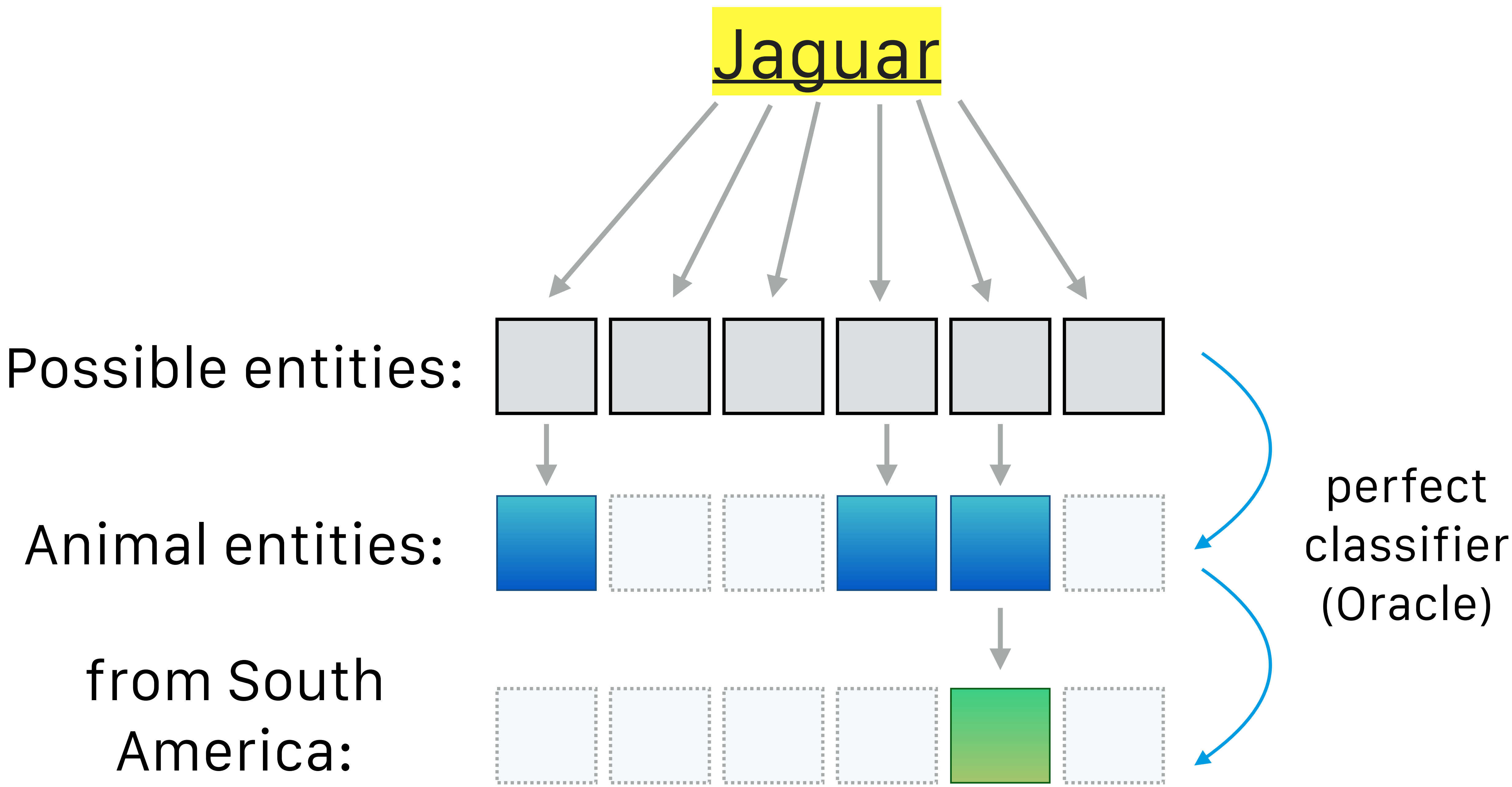
Disambiguation accuracy

Intractable mixed integer problem

Subproblems

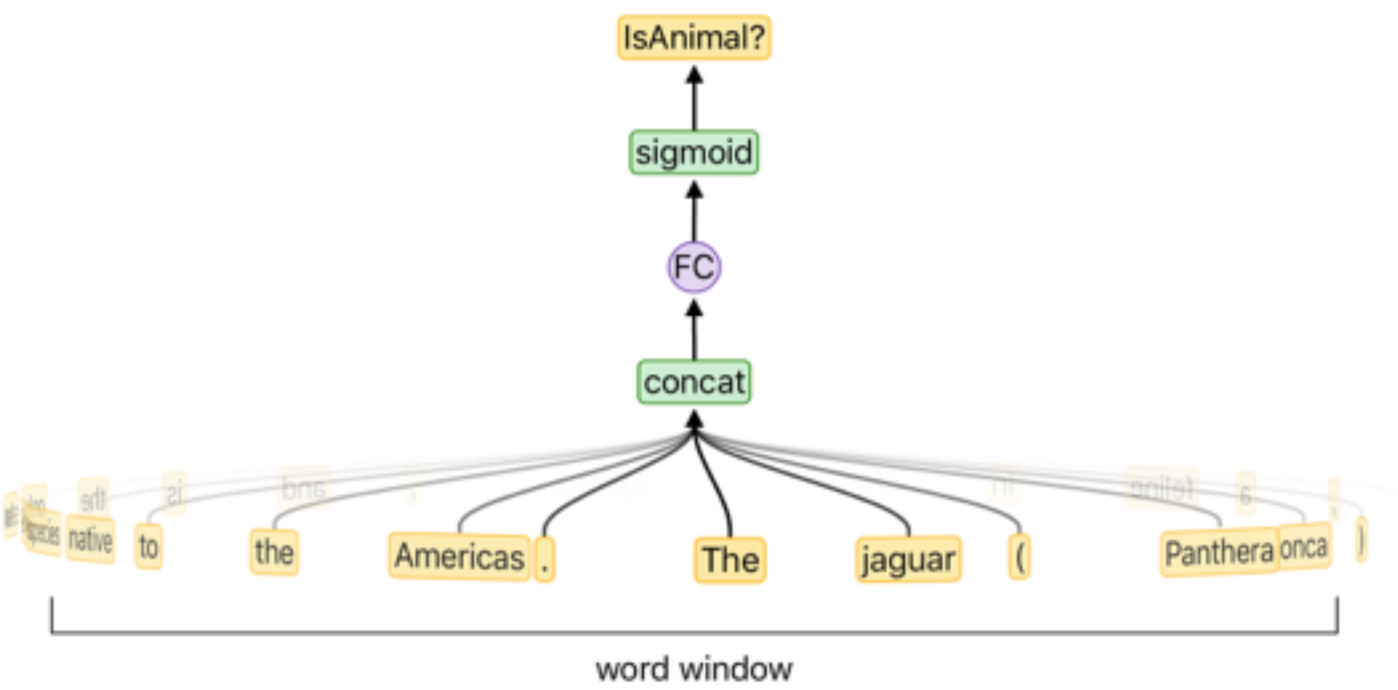
1. Stochastic optimization/heuristic search to design type system
2. Gradient descent to train a type classifier

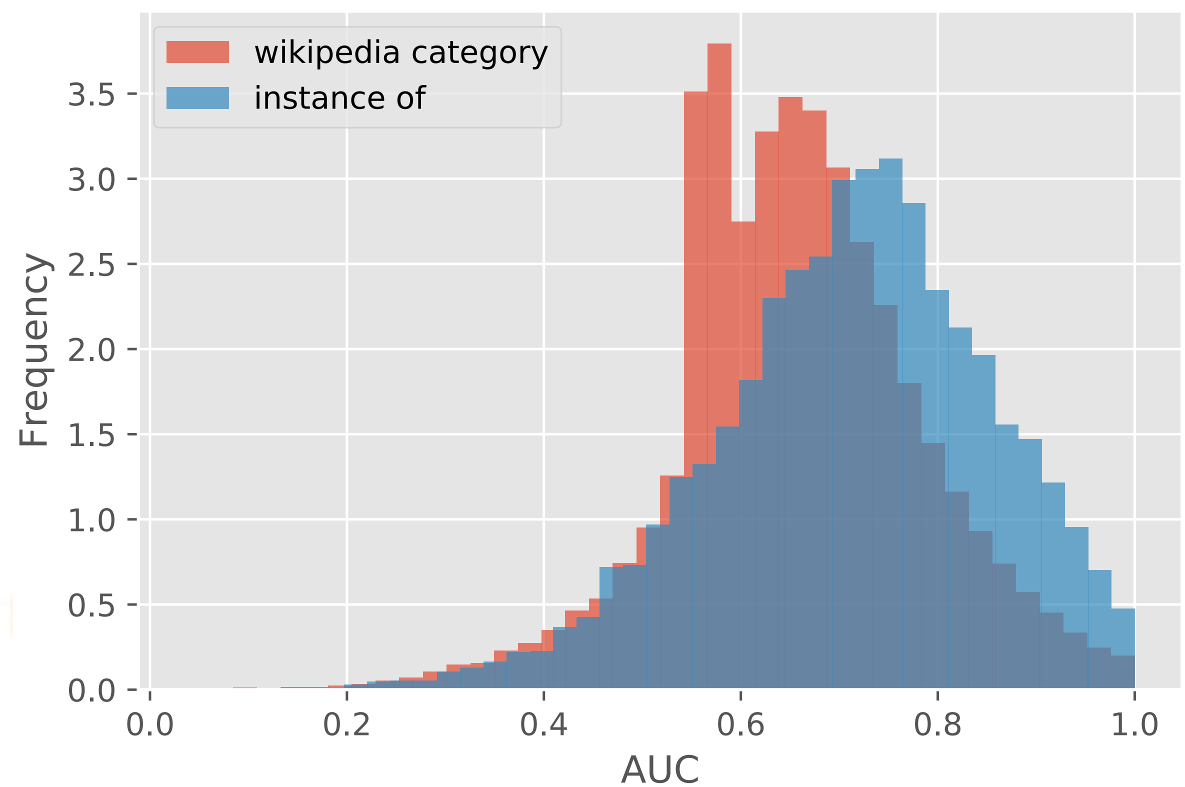
Oracle Accuracy



Type Learnability

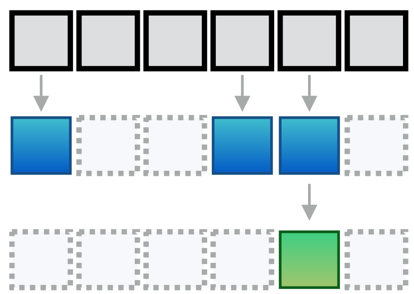
- Which types are predictable from context?
- Train a proxy binary classifier for each type
- AUC* of classifier is an estimate of Learnability

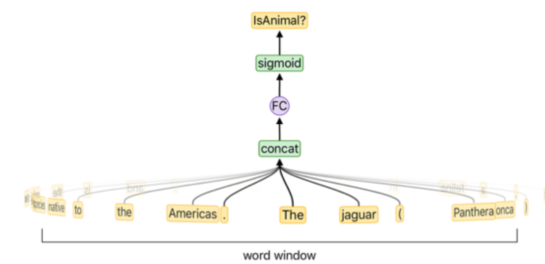




* average AUC over 4 training runs

Type System Evolution

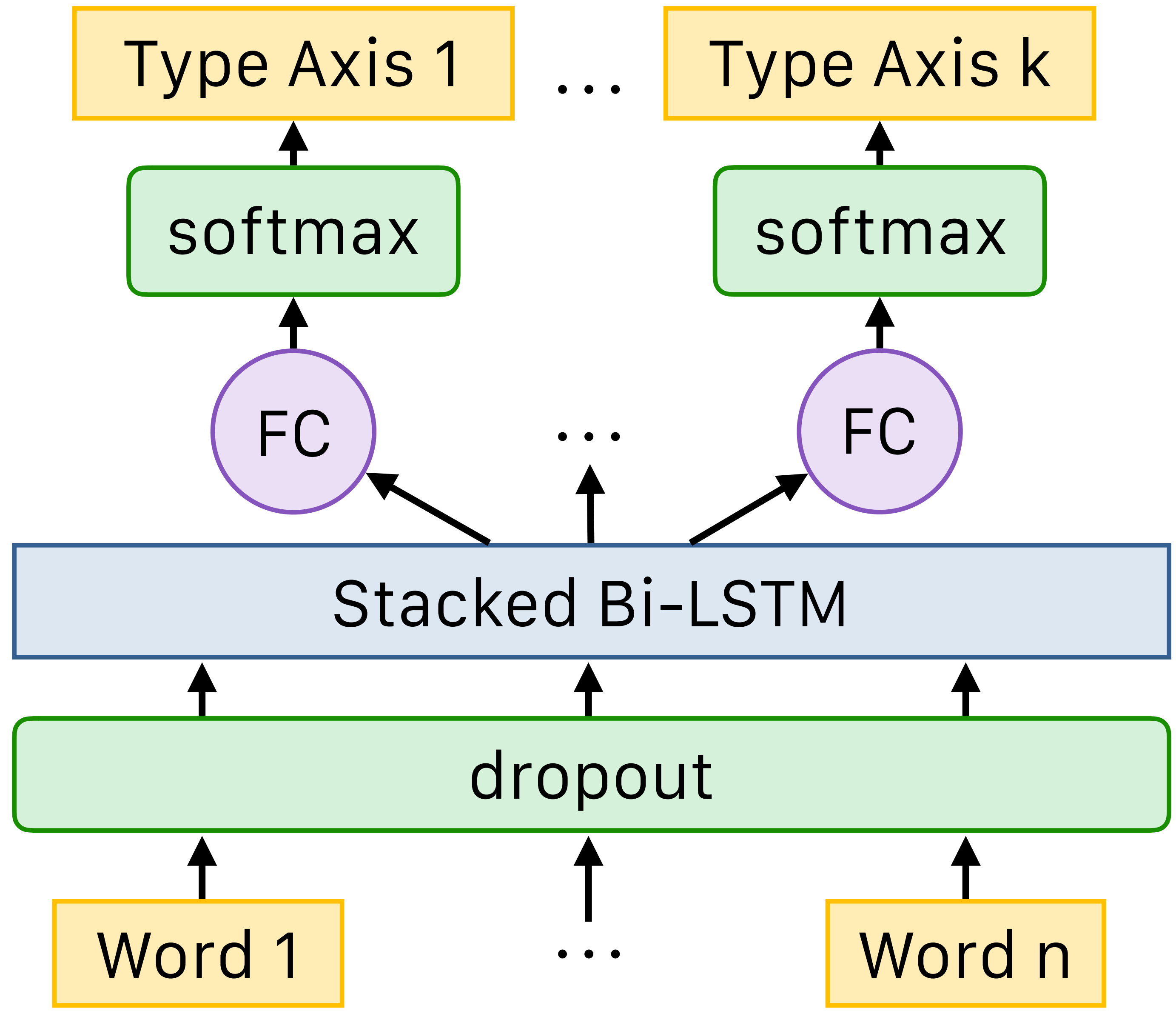




$$J(\mathcal{A}) = (\mathrm{Accuracy}_{\mathrm{oracle}}(\mathcal{A}) - \mathrm{Accuracy}_{\mathrm{greedy}}) \cdot \mathrm{Learnability}(\mathcal{A}) - \lambda \cdot |\mathcal{A}|$$

Diminishing Returns

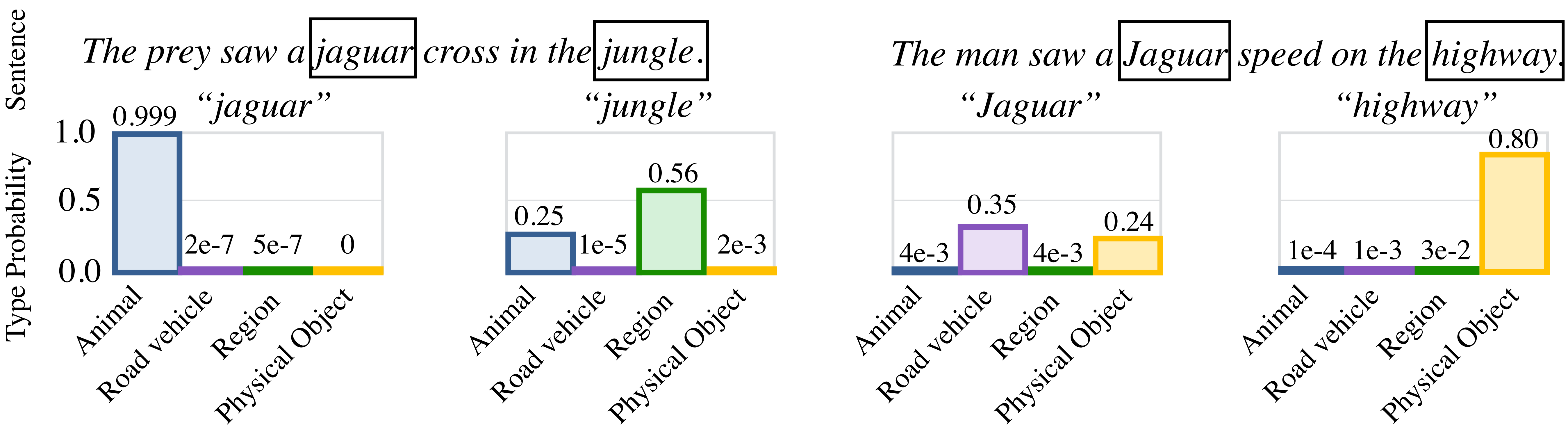
Train a type classifier



Intra-wiki links are type labels!

(in any Wikipedia language)

Inference



Entity	**jaguar**	**Jaguar**	**jungle**	**jungle**	**jaguar**	**Jaguar**	**highway**	**Highway**
Type	Animal	Road vehicle	Region	Music	Animal	Road vehicle	Physical Object	Film
only link Prob.	0.29	**0.60**	**0.35**	0.17	0.29	**0.60**	**0.85**	0.04
Prob. w/. types	**1.0**	0.0	**1.0**	0.0	0.0	**1.0**	**1.0**	0.0

Results

	Model	CoNLL (YAGO)	TAC 2010
no types	Globerson et al. 2016	91,70 %	87,20 %
	Yamada et al. 2016	91,50 %	85,20 %
	NTEE (Yamada et al. 2017)	-	87,70 %
types	DeepType (human types)	93,11 %	90,74 %
	DeepType (greedy)	94,15 %	**90,85 %**
	DeepType (GA)	**94,88 %**	90,31 %
	DeepType (CEM)	93,96 %	90,30 %

Contributions

- Outperform state of the art on several entity linking benchmark datasets
- Add entities without retraining by specifying their types
- Design & integrate symbolic structure to constrain neural network outputs

Code:
github.com/openai/deeptype

Objective

- Given:
  - ambiguous mentions $m \in M$
  - The ground truth entity e_{GT} for each m
  - Model prediction e*
  - Model accuracy S_{model}

$$\max_{\mathcal{A}} \max_{\theta} S_{\text{model}}(\mathcal{A}, \theta) = \frac{\sum_{(m, e_{\text{GT}}, \mathcal{E}_m) \in M} \mathbb{1}_{e_{\text{GT}}}(e^*)}{|M|}$$

- Find type system $\boldsymbol{\mathcal{A}}$, and parameters θ to maximise disambiguation accuracy
- $\boldsymbol{\mathcal{A}}$ are discrete variables selecting the types to use

Inference

- $\mathbb{P}(\text{types}(e)|c)$ = compute type probabilities per token
- $\mathbb{P}_{\text{link}}(e|c)$ = # intra-wiki links from anchor→articles
- Baye's rule, entity e, context c:
  - $\mathbb{P}(\text{e}|\text{c}) = \mathbb{P}_{\text{link}}(e|c) * \mathbb{P}(\text{types}(e)|c)$

Type system objective

$$J(\mathcal{A}) = (\mathrm{Acc}_{oracle}(\mathcal{A}) - \mathrm{Acc}_{greedy}) \bullet \mathrm{Learnability}(\mathcal{A}) - \boldsymbol{\lambda} \bullet |\mathcal{A}|$$