Overflow + Case Study

LING 571 — Deep Processing Methods in NLP

Shane Steinert-Threlkeld

Announcements

- HW9:
  - Coarse-grained sense labels (e.g. “Comparison”, not “Comparison.Contrast”)
  - Pre-processing: not required, but lower-casing and removing punctuation are good ones
  - Unknown words (even after processing); ~two options when lookup fails
    - All zeros vector
    - Average of all GloVe vectors
  - For these last two points (and any other choices): no “wrong” answers, but tell us what you did in your readme!
- AMA for Wednesday discussion: https://forms.gle/qgM8V5Pp1mQL6gCg8

Coreference Resolution Humor

Coreference Resolution Humor

Coreference Resolution Humor pt. 2

A young artist exhibits his work for the first time and a well known art critic is in attendance.

The critic says to the young artist, "would you like my opinion on your work?"

"Yes, " says the artist.

"It's worthless," says the critic

The artist replies, "I know, but tell me anyway."

Coreference Resolution Humor pt. 2

A young artist exhibits his work for the first time and a well known art critic is in attendance.

The critic says to the young artist, "would you like **my opinion** on **your work**?"

"Yes, " says the artist.

"**It's** worthless," says the critic

The artist replies, "I know, but tell me anyway."

Roadmap

- Overflow: coherence and cohesion
- Case study
  - deep vs. shallow processing in question answering
  - A bit on then vs now

Question-Answering:

A Case Study in Shallow vs. Deep Methods

Question Answering: The Problem

- Grew out of information retrieval community

Question Answering: The Problem

- Grew out of information retrieval community
- Document retrieval is great, but…
  - Sometimes you don't just want a ranked list of documents.
  - Sometimes you want an answer to a question
    - Short answer, possibly with supporting context

Question Answering: The Problem

- Grew out of information retrieval community
- Document retrieval is great, but…
  - Sometimes you don't just want a ranked list of documents.
  - Sometimes you want an answer to a question
    - Short answer, possibly with supporting context
- People ask questions on the web
  - *Which English translation of the Bible is used in official Catholic liturgies?*
  - *Who invented surf music?*
  - *What are the seven wonders of the world?*
  - These account for 12–15% of web log queries

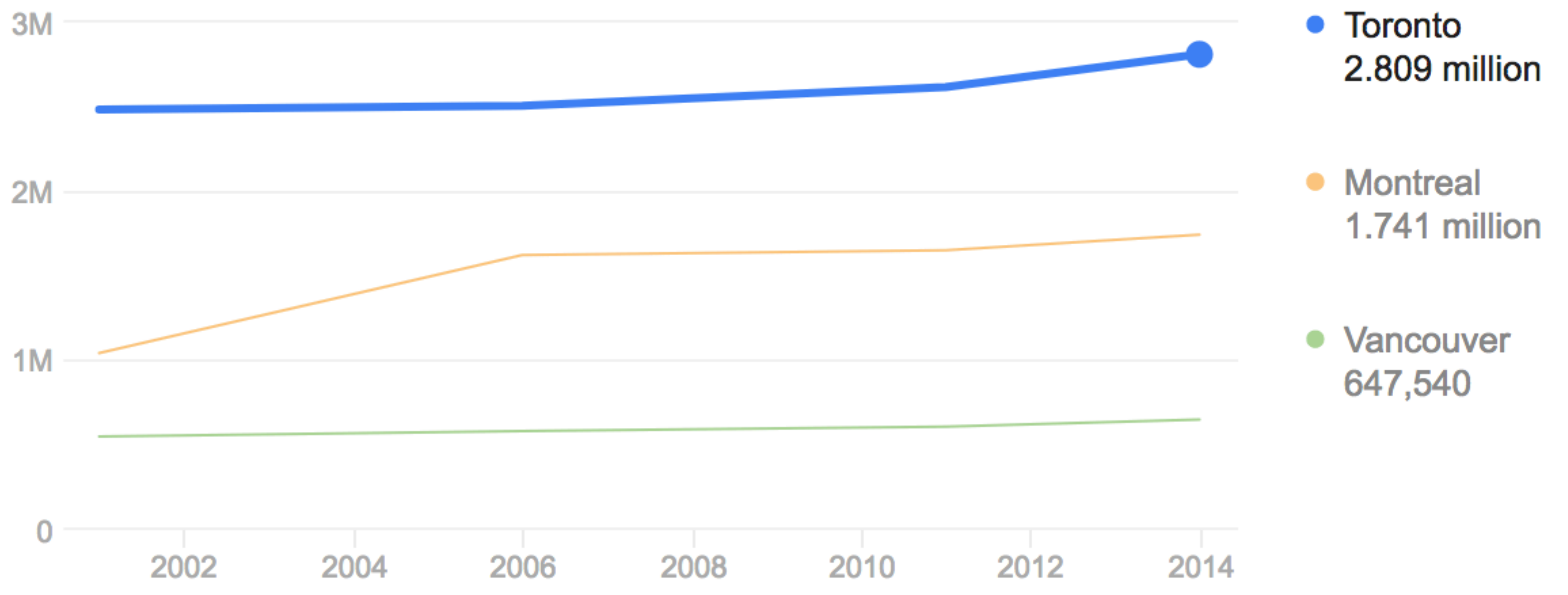
Search Engines and Questions

- What do search engines do with questions?
  - Increasingly, try to answer questions
  - Especially for Wikipedia infobox types of info
  - Backoff to keyword search
- How well does this work?

What Canadian city has the largest population?

Toronto / **Population**

2.809 million (2014)



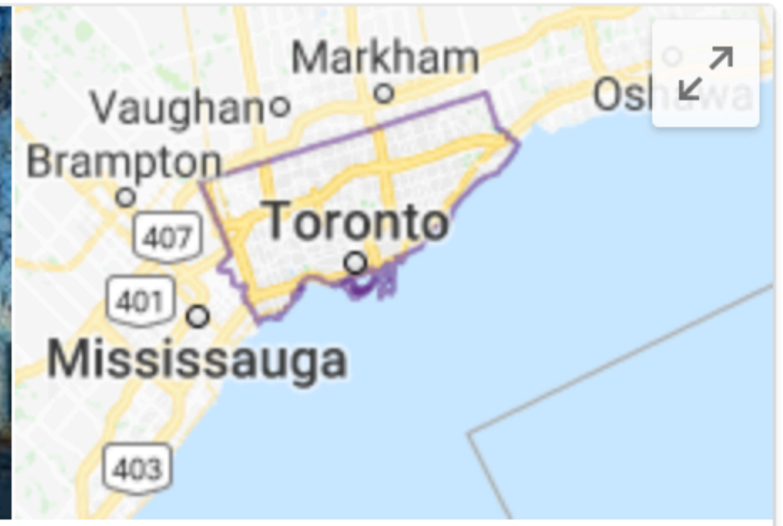
Explore more

Sources include: UNdata *Feedback*

People also ask

What are the 3 largest cities in Canada by population?

What are the 5 major cities in Canada?



Toronto

City in Ontario, Canada

Toronto, the capital of the province of Ontario, is a major Canadian city along Lake Ontario's northwestern shore. It's a dynamic metropolis with a core of soaring skyscrapers, all dwarfed by the iconic, free-standing CN Tower. Toronto also has many green spaces, from the orderly oval of Queen's Park to 400-acre High Park and its trails, sports facilities and zoo.

Population elsewhere

Canada	35.54 million (2014)
New York City	8.472 million (2014)
Chicago	2.719 million (2014)

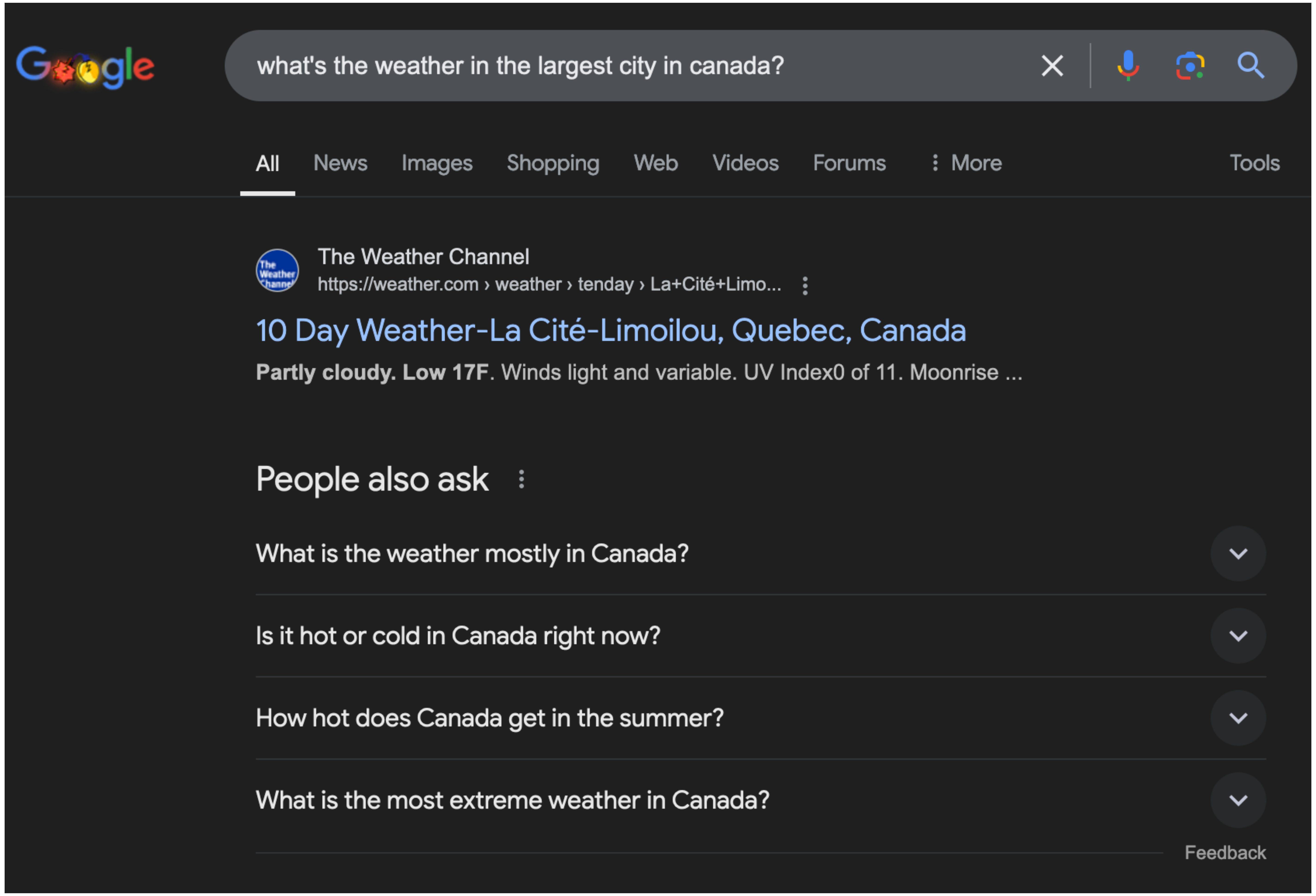
Sources include: World Bank, United States Census Bureau *Feedback*

* Answer from late 2022; more on this later

What's the weather in Toronto?

12/02/24

What's the weather in the largest city in Canada?



12/02/24

What is the total population of the ten largest capitals in the US?

- Rank 1 snippet:
  - As of 2013, 61,669,629 citizens lived in ***America's*** 100 ***largest cities***, which was 19.48 percent of the nation's ***total population***.
  - See the top 50 ***U.S. cities by population*** and rank. ... The table below lists the *largest* 50 *cities in the*
  - The table below lists the *largest* 10 *cities in the United States*…

Breaking QA Systems

https://twitter.com/xkcd/status/1333529967079120896

Breaking QA Systems

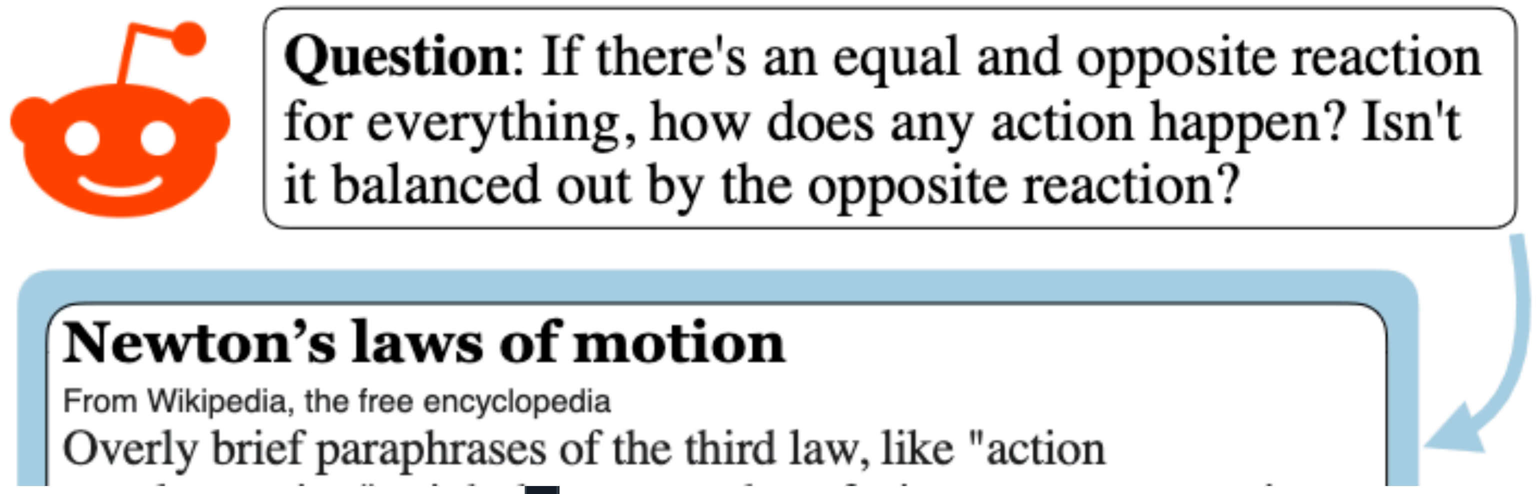
CREPE: Open-Domain Question Answering with False Presuppositions

Xinyan Velocity Yu[†] **Sewon Min**[†] **Luke Zettlemoyer**[†] **Hannaneh Hajishirzi**[†,‡]
[†]University of Washington [‡]Allen Institute for Artificial Intelligence
{xyu530,sewon,lsz,hannaneh}@cs.washington.edu

3335299670791 20896

Abstract

Information seeking users often pose questions with false presuppositions, especially when asking about unfamiliar topics. Most existing question answering (QA) datasets, in



https://doi.org/10.18653/v1/2023.acl-long.583

Search Engines and QA

- Search for exact question string
  - "Do I need a visa to go to Japan?"
    - Result: Exact match on Yahoo! Answers
    - Find "Best Answer" and return following chunk

Search Engines and QA

- Search for exact question string
  - “Do I need a visa to go to Japan?”
    - Result: Exact match on Yahoo! Answers
    - Find “Best Answer” and return following chunk
- Works great… if the question matches exactly
  - Many websites are building archives
  - What happens if it doesn’t match?
    - “Question mining” tries to learn paraphrases of questions to get answers.

Perspectives on QA

- TREC QA track (~2000—)
  - Initially pure factoid questions, with fixed length answers
    - Based on large collection of fixed documents (news)
    - Increasing complexity: definitions, biographical info, etc
      - Single response

Perspectives on QA

- TREC QA track (~2000—)
  - Initially pure factoid questions, with fixed length answers
    - Based on large collection of fixed documents (news)
    - Increasing complexity: definitions, biographical info, etc
      - Single response
- Reading comprehension (Hirschman et al, 1999—)
  - Think SAT/GRE
    - Short text or article (usually middle school level)
    - Answer questions based on text
  - Also, “Machine Reading”
  - SQuAD

Perspectives on QA

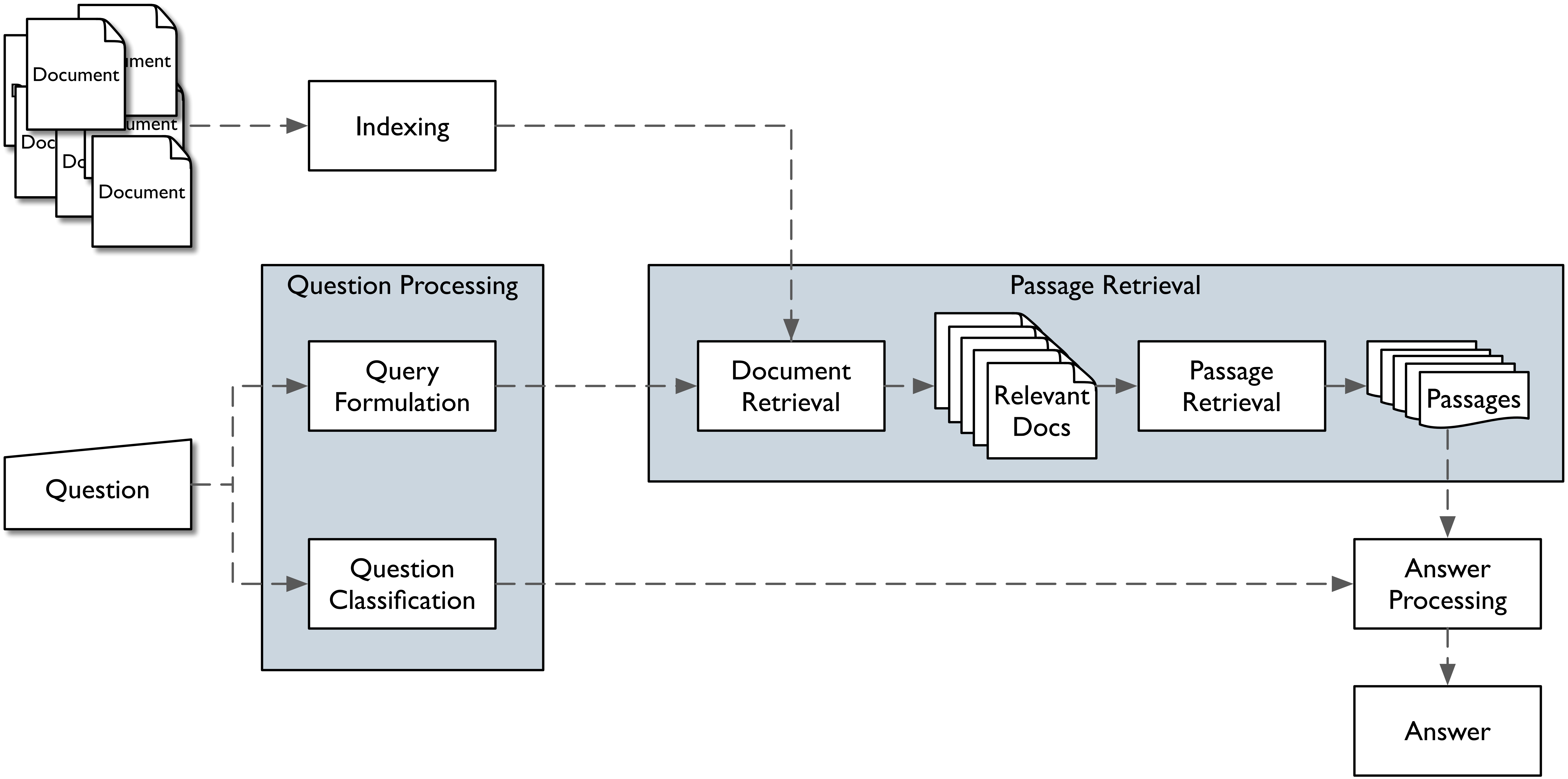
- TREC QA track (~2000—)
  - Initially pure factoid questions, with fixed length answers
    - Based on large collection of fixed documents (news)
    - Increasing complexity: definitions, biographical info, etc
      - Single response
- Reading comprehension (Hirschman et al, 1999—)
  - Think SAT/GRE
    - Short text or article (usually middle school level)
    - Answer questions based on text
  - Also, "Machine Reading"
  - SQuAD
- And, of course, Jeopardy! and Watson

Question Answering (*a la* TREC)

Question	Answer
Where is the Louvre Museum located?	in Paris, France
What's the abbreviation for limited partnership?	L.P.
What are the names of Odin's ravens?	Huginn and Muninn
What currency is used in China?	the yuan
What kind of nuts are used in marzipan?	almonds
What instrument does Max Roach play?	drums
What's the official language of Algeria?	Arabic
What is the telephone number for the University of Colorado, Boulder?	(303) 492-1411
How many pounds are there in a stone?	14

Basic Strategy

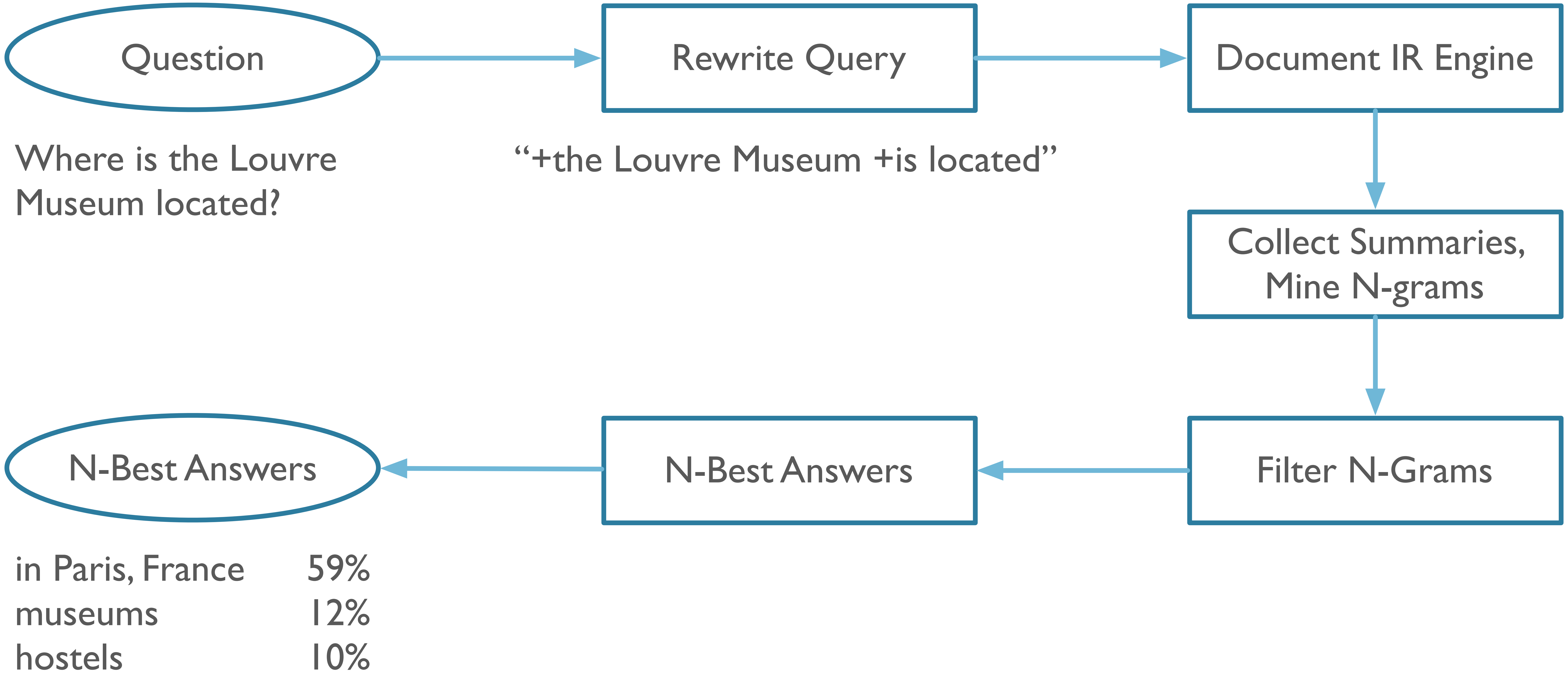
- Given an indexed document collection…
- …and a question…
- …execute the following steps:
  - Query Formulation
  - Question Classification
  - Passage Retrieval
  - Answer Processing
  - Evaluation



UNIVERSITY of WASHINGTON

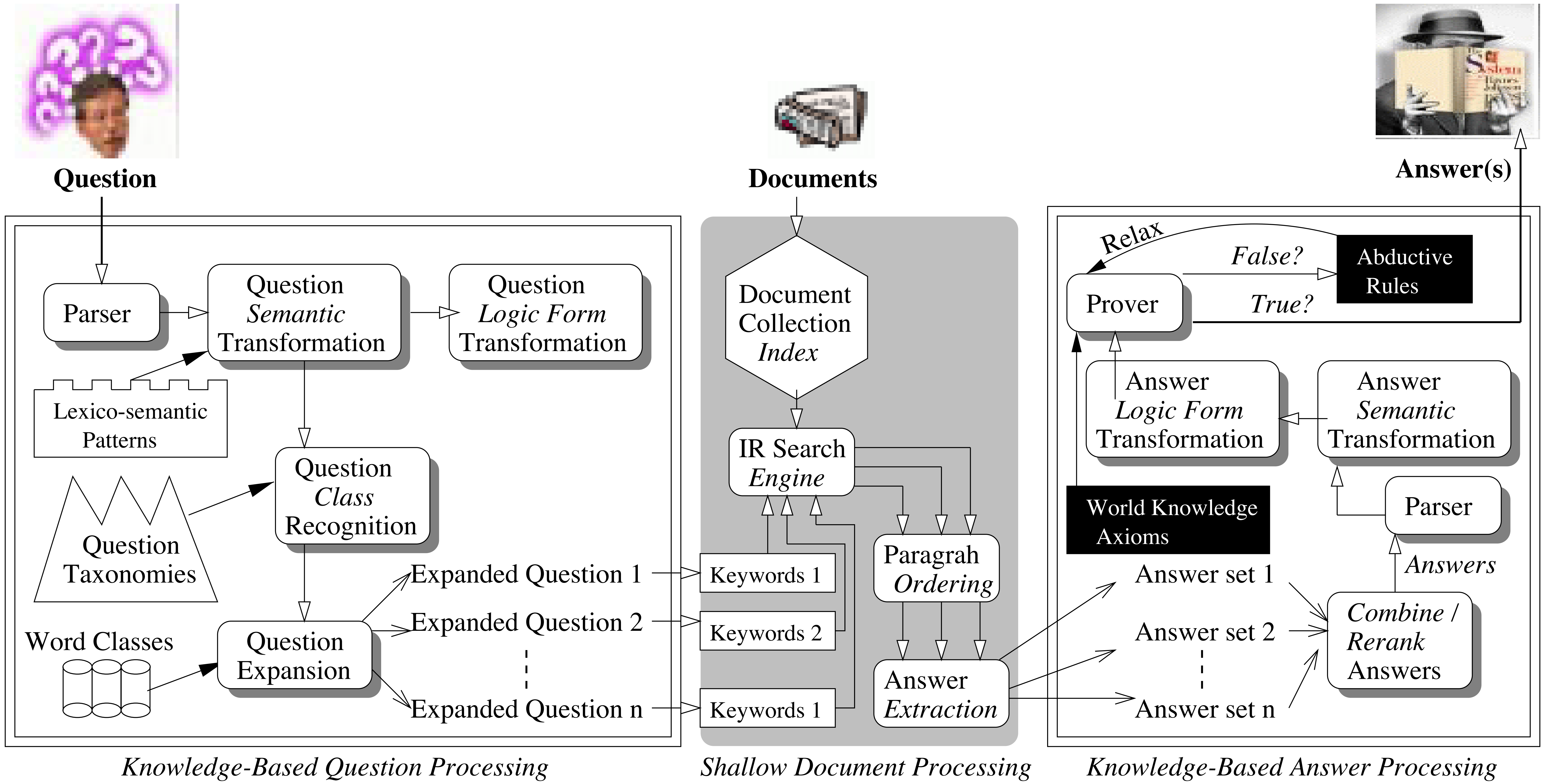
AskMSR/Aranea (Lin, Brill)

- Shallow Processing for QA



Deep Processing Technique for QA:
LCC PowerAnswer

- Experiments with open-domain textual Question Answering, [Moldovan, Harabagiu, et al, 2000]



A Victory for Deep Processing:

TREC 2002 QA Track

Run Tag	Confidence weighted Score	Correct Answers #	Correct Answers %	Number Inexact	NIL Accuracy Prec	NIL Accuracy Recall
LCCmain2002	**0.856**	**415**	**83.0**	**8**	**0.578**	**0.804**
exactanswer	0.691	271	54.2	12	0.222	0.848
pris2002	0.610	290	58.0	17	0.241	0.891
IRST02D1	0.589	192	38.4	17	0.167	0.217
IBMPQSQACYC	0.588	179	35.8	9	0.196	0.630
uwmtB3	0.512	184	36.8	20	0.000	0.000
BBN2002C	0.499	142	28.4	18	0.182	0.087
isi02	0.498	149	29.8	15	0.385	0.109
limsiQalir2	0.497	133	26.6	11	0.188	0.196
ali2002b	0.496	181	36.2	15	0.156	0.848
ibmsqa02c	0.455	145	29.0	44	0.224	0.239
FDUT11QA1	0.434	124	24.8	6	0.139	0.957
aranea02a	0.433	152	30.4	36	0.235	0.174
nuslamp2002	0.396	105	21.0	17	0.000	0.000
pqas22	0.358	133	26.6	11	0.145	0.674

Example of Deep Processing in LLM era

LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers

Theo X. Olausson*[1] Alex Gu*[1] Benjamin Lipkin*[2] Cedegao E. Zhang*[2]
Armando Solar-Lezama[1] Joshua B. Tenenbaum[1,2] Roger Levy[2]
{theoxo, gua, lipkinb, cedzhang}@mit.edu
[1]MIT CSAIL [2]MIT BCS
*Equal contribution.

Abstract

Logical reasoning, i.e., deductively inferring the truth value of a conclusion from a set of premises, is an important task for artificial intelligence with wide potential impacts on science, mathematics, and society. While many prompting-based strategies have been proposed to enable Large Language Models (LLMs) to do such reasoning more effectively, they still appear unsatisfactory, often failing in subtle and unpredictable ways. In this work, we investigate the validity of instead reformulating such tasks as modular neurosymbolic programming, which we call LINC: Logical Inference via Neurosymbolic Computation. In LINC, the LLM acts as a semantic parser, translating premises and conclusions from natural language to expressions in first-order logic. These expressions are then offloaded to an external theorem prover, which symbolically performs deductive inference. Leveraging this approach, we observe significant performance gains on FOLIO and a balanced subset of ProofWriter for three different models in nearly all experimental conditions we evaluate. On

1 Introduction

Widespread adoption of large language models (LLMs) such as GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023), and PaLM (Chowdhery et al., 2022) have led to a series of remarkable successes in tasks ranging from text summarization to program synthesis. Some of these successes have encouraged the hypothesis that such models are able to flexibly and systematically reason (Huang and Chang, 2022), especially when using prompting strategies that explicitly encourage verbalizing intermediate reasoning steps before generating the final answer (Nye et al., 2021; Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023b). However, this reasoning ability appears to be unreliable for tasks that require reasoning out of domain (Liang et al., 2022; Saparov et al., 2023), understanding negation (Anil et al., 2022), and following long reasoning chains (Dziri et al., 2023). Furthermore, while the standard approach of "scaling up" seems to improve performance across some reasoning domains, other domains, e.g., reasoning involving use of Modus Tollens, show no such improvements

https://openreview.net/forum?id=h00GHjWDEp

Example of Deep Processing in LLM era

LINC: A Neurosymbolic Approach for Logical Reasoning by Language Models with First-Order Logic Provers

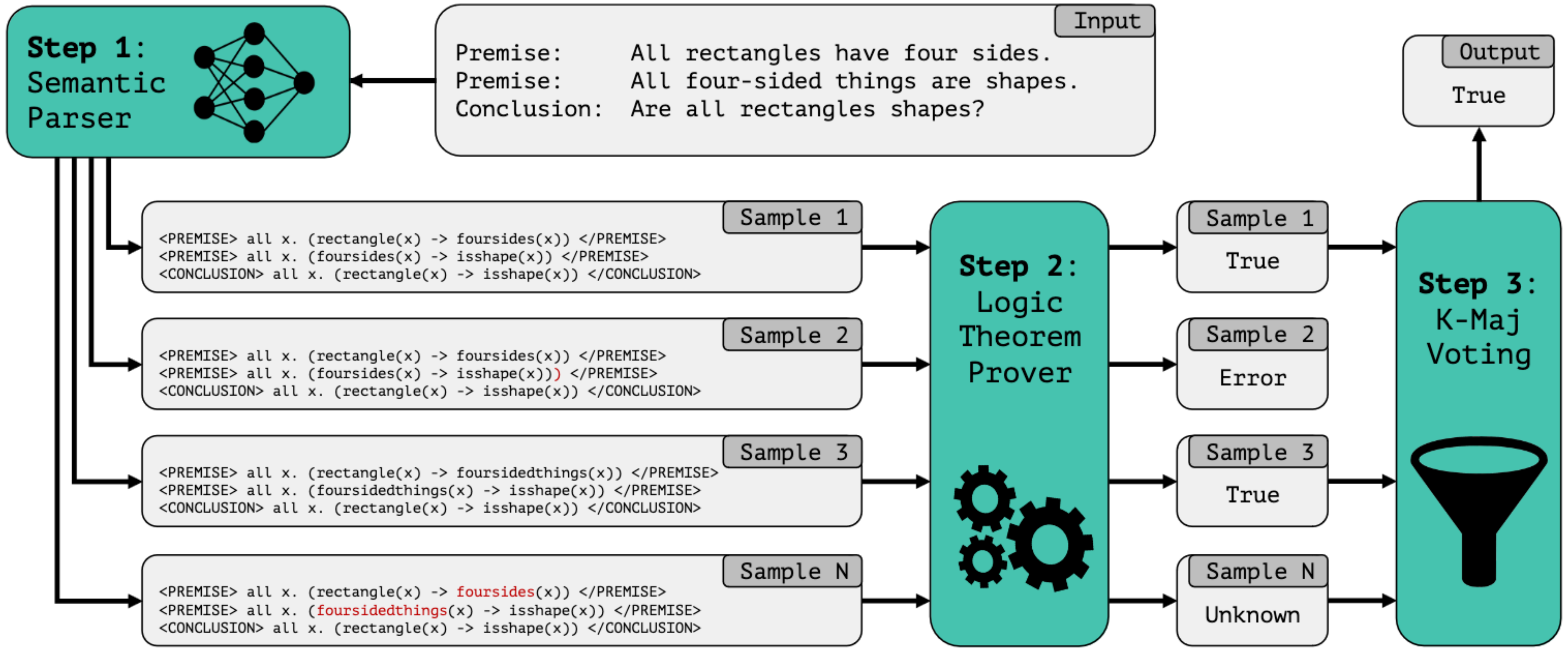
Theo X. Olausson*[1] Alex Gu*[1] Benjamin Lipkin*[2] Cedegao E.
Armando Solar-Lezama[1] Joshua B. Tenenbaum[1,2] Roger L
{theoxo, gua, lipkinb, cedzhang}@mit.edu
[1]MIT CSAIL [2]MIT BCS
Equal contribution.

Abstract

Logical reasoning, i.e., deductively inferring the truth value of a conclusion from a set of premises, is an important task for artificial intelligence with wide potential impacts on science, mathematics, and society. While many prompting-based strategies have been proposed to enable Large Language Models (LLMs) to do such reasoning more effectively, they still appear unsatisfactory, often failing in subtle and unpredictable ways. In this work, we investigate the validity of instead reformulating such tasks as modular neurosymbolic programming, which we call LINC: Logical Inference via Neurosymbolic Computation. In LINC, the LLM acts as a semantic parser, translating premises and conclusions from natural language to expressions in first-order logic. These expressions are then offloaded to an external theorem prover, which symbolically performs deductive inference. Leveraging this approach, we observe significant performance gains on FOLIO and a balanced subset of ProofWriter for three different models in nearly all experimental conditions we evaluate. On

1 Introduction

Widespread adoption of large l (LLMs) such as GPT-3 (Brown et 4 (OpenAI, 2023), and PaLM (C 2022) have led to a series of rema in tasks ranging from text summ gram synthesis. Some of these su couraged the hypothesis that such to flexibly and systematically rea Chang, 2022), especially when using prompting strategies that explicitly encourage verbalizing intermediate reasoning steps before generating the final answer (Nye et al., 2021; Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023b). However, this reasoning ability appears to be unreliable for tasks that require reasoning out of domain (Liang et al., 2022; Saparov et al., 2023), understanding negation (Anil et al., 2022), and following long reasoning chains (Dziri et al., 2023). Furthermore, while the standard approach of "scaling up" seems to improve performance across some reasoning domains, other domains, e.g., reasoning involving use of Modus Tollens, show no such improvements



https://openreview.net/forum?id=h00GHjWDEp

Example of Deep Processing in LLM era

LINC: A Neurosymbolic Approach for Logical Reasoning by Language Models with First-Order Logic Provers

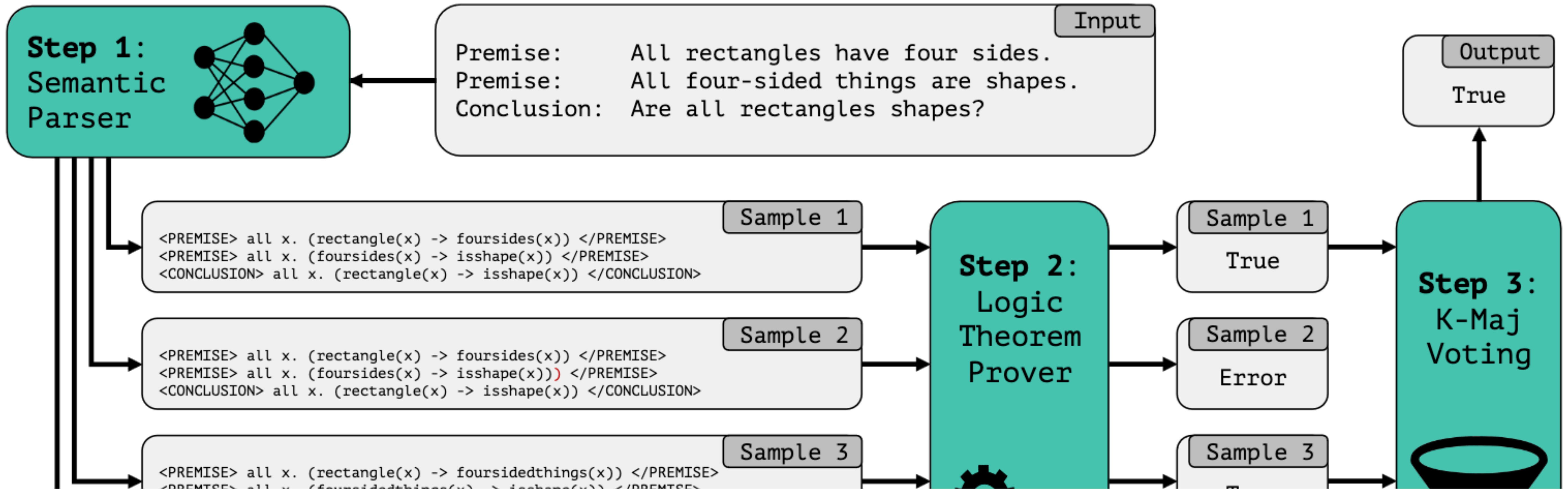
Theo X. Olausson*[1] Alex Gu*[1] Benjamin Lipkin*[2] Cedegao E.
Armando Solar-Lezama[1] Joshua B. Tenenbaum[1,2] Roger L
{theoxo, gua, lipkinb, cedzhang}@mit.edu
[1]MIT CSAIL [2]MIT BCS
Equal contribution.

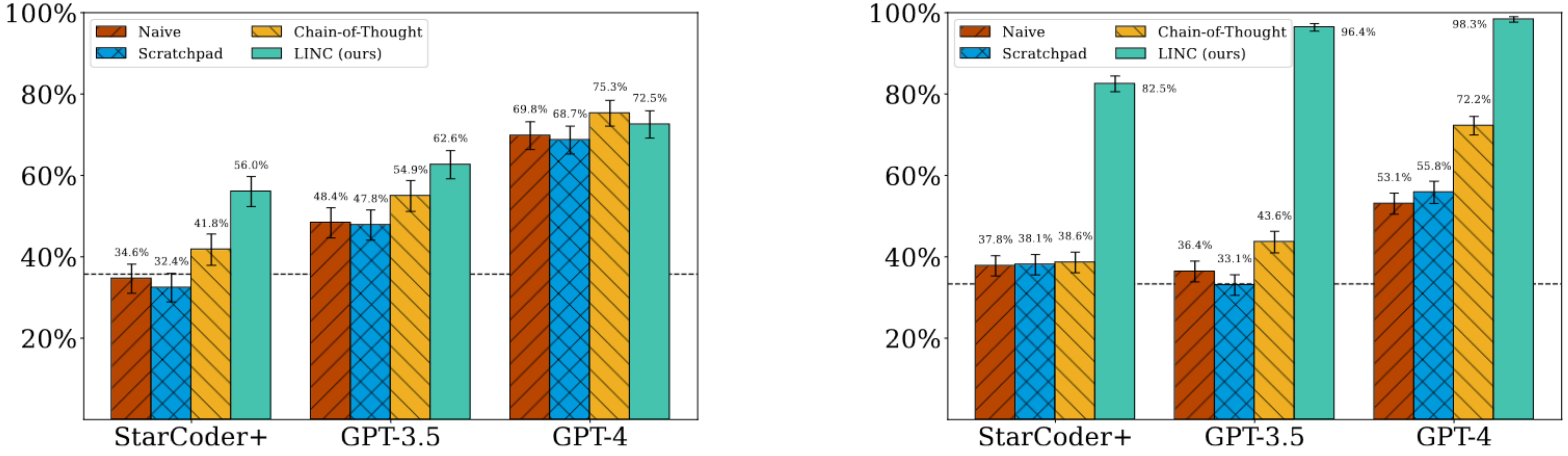
Abstract

Logical reasoning, i.e., deductively inferring the truth value of a conclusion from a set of premises, is an important task for artificial intelligence with wide potential impacts on science, mathematics, and society. While many prompting-based strategies have been proposed to enable Large Language Models (LLMs) to do such reasoning more effectively, they still appear unsatisfactory, often failing in subtle and unpredictable ways. In this work, we investigate the validity of instead reformulating such tasks as modular neurosymbolic programming, which we call LINC: Logical Inference via Neurosymbolic Computation. In LINC, the LLM acts as a semantic parser, translating premises and conclusions from natural language to expressions in first-order logic. These expressions are then offloaded to an external theorem prover, which symbolically performs deductive inference. Leveraging this approach, we observe significant performance gains on FOLIO and a balanced subset of ProofWriter for three different models in nearly all experimental conditions we evaluate. On

1 Introduction

Widespread adoption of large l
(LLMs) such as GPT-3 (Brown et
4 (OpenAI, 2023), and
2022) have led to a seri
in tasks ranging from
gram synthesis. Some
couraged the hypothesi
to flexibly and systema
Chang, 2022), especia
strategies that explicitl
termediate reasoning s
final answer (Nye et a
Kojima et al., 2022; Wa
this reasoning ability a
tasks that require reaso
et al., 2022; Saparov e
negation (Anil et al.,
reasoning chains (Dziri
while the standard appr
to improve performance
mains, other domains, e
of Modus Tollens, show no such improvements



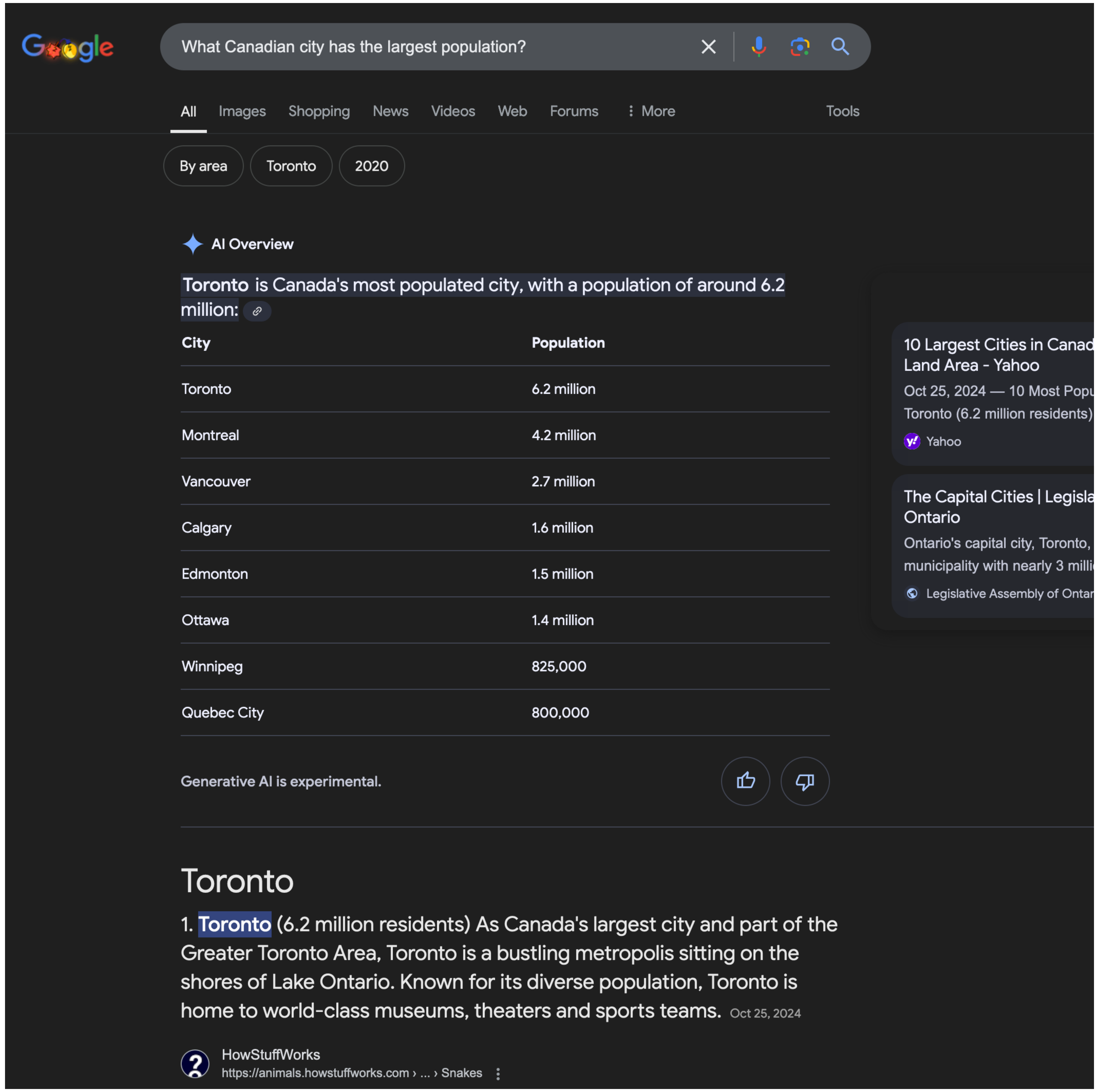


(a) FOLIO.

(b) ProofWriter.

https://openreview.net/forum?id=h00GHjWDEp

Re-visiting some questions


What Canadian city has the largest population?

All Images Shopping News Videos Web Forums More Tools

By area Toronto 2020

AI Overview

Toronto is Canada's most populated city, with a population of around 6.2 million:

City	Population
Toronto	6.2 million
Montreal	4.2 million
Vancouver	2.7 million
Calgary	1.6 million
Edmonton	1.5 million
Ottawa	1.4 million
Winnipeg	825,000
Quebec City	800,000

Generative AI is experimental.

10 Largest Cities in Canad
Land Area - Yahoo
Oct 25, 2024 — 10 Most Popu
Toronto (6.2 million residents)
Yahoo

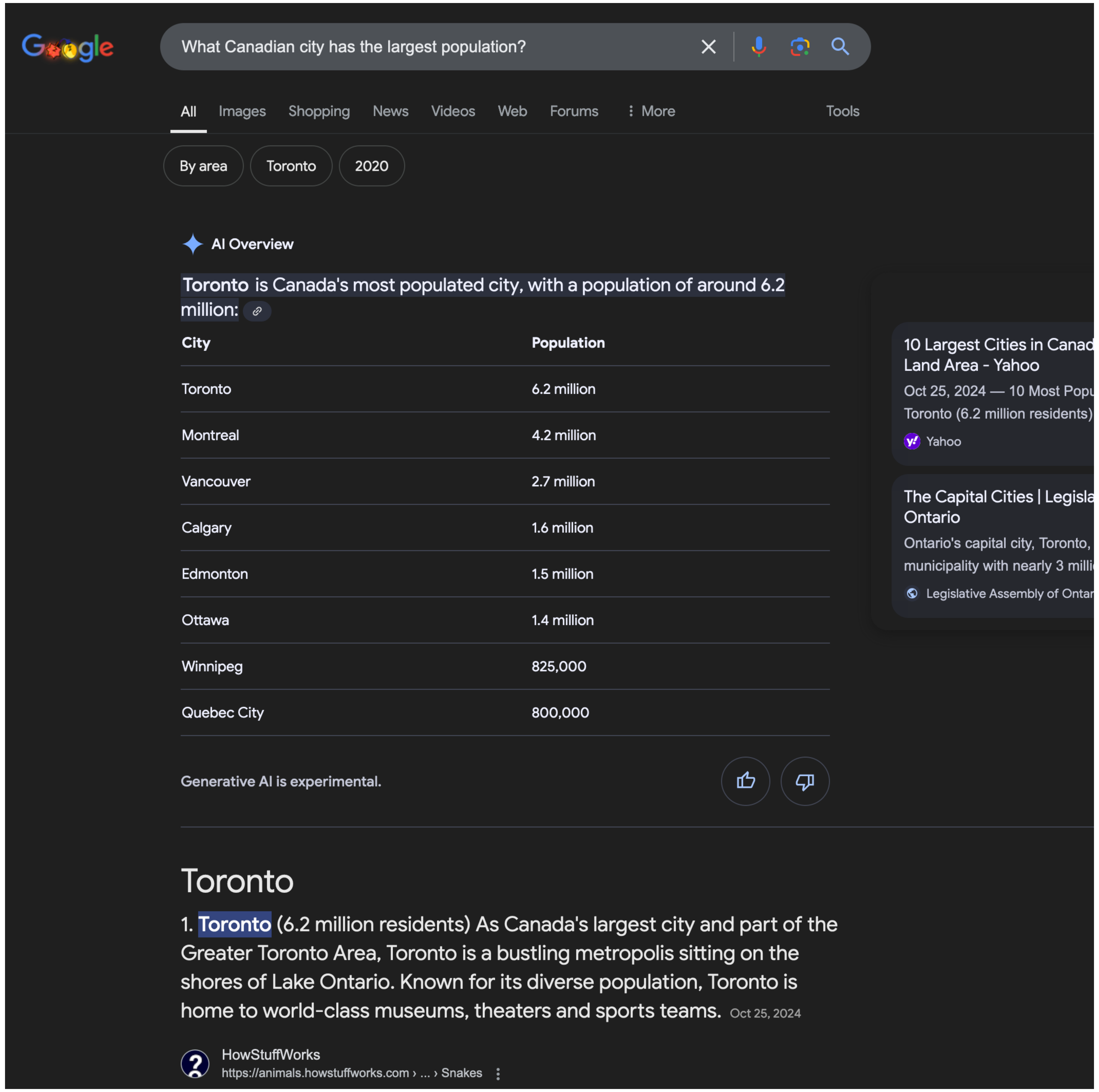
The Capital Cities | Legisla
Ontario
Ontario's capital city, Toronto,
municipality with nearly 3 milli
Legislative Assembly of Ontar

Toronto

1. Toronto (6.2 million residents) As Canada's largest city and part of the Greater Toronto Area, Toronto is a bustling metropolis sitting on the shores of Lake Ontario. Known for its diverse population, Toronto is home to world-class museums, theaters and sports teams. Oct 25, 2024

HowStuffWorks
https://animals.howstuffworks.com › ... › Snakes

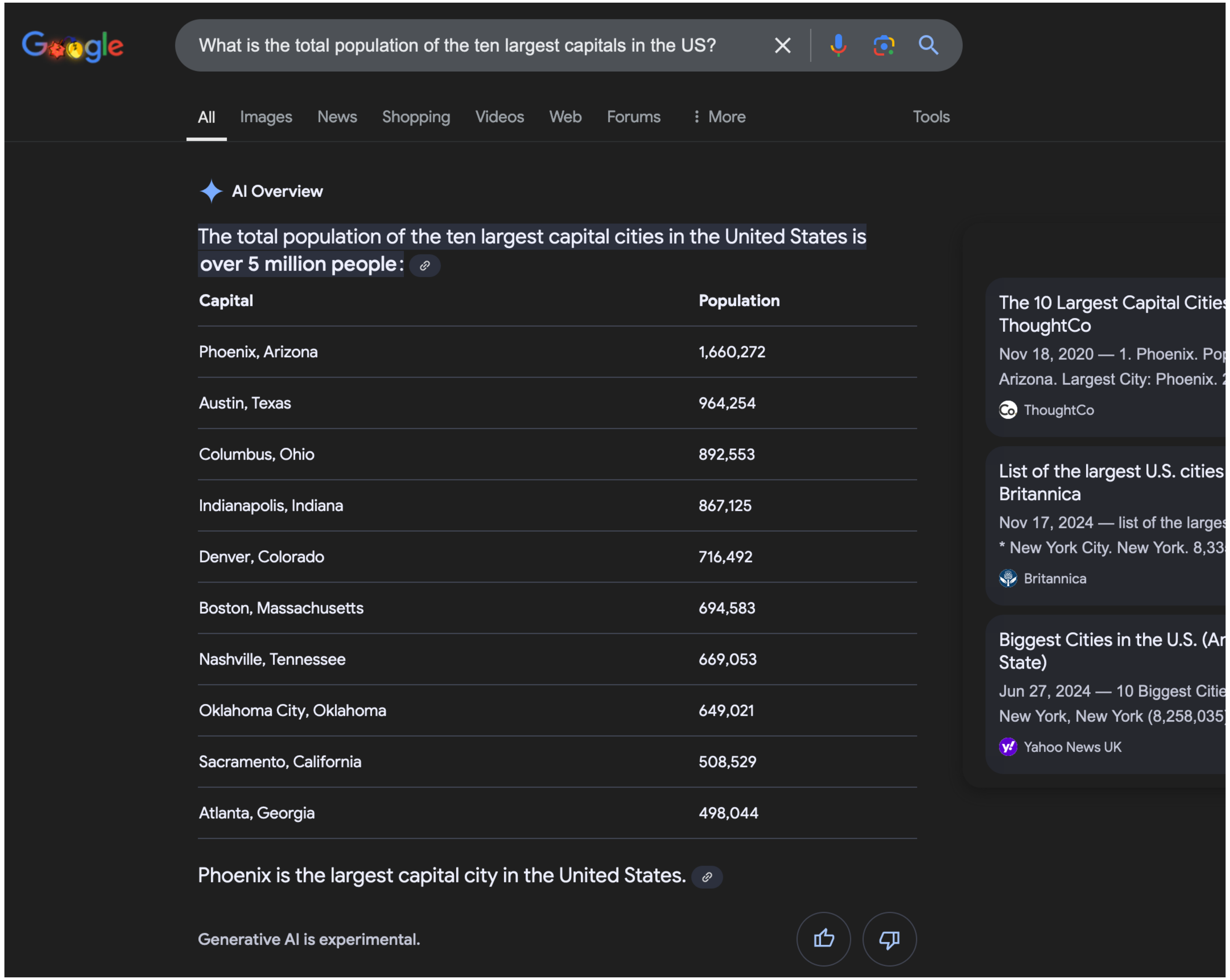
Re-visiting some questions



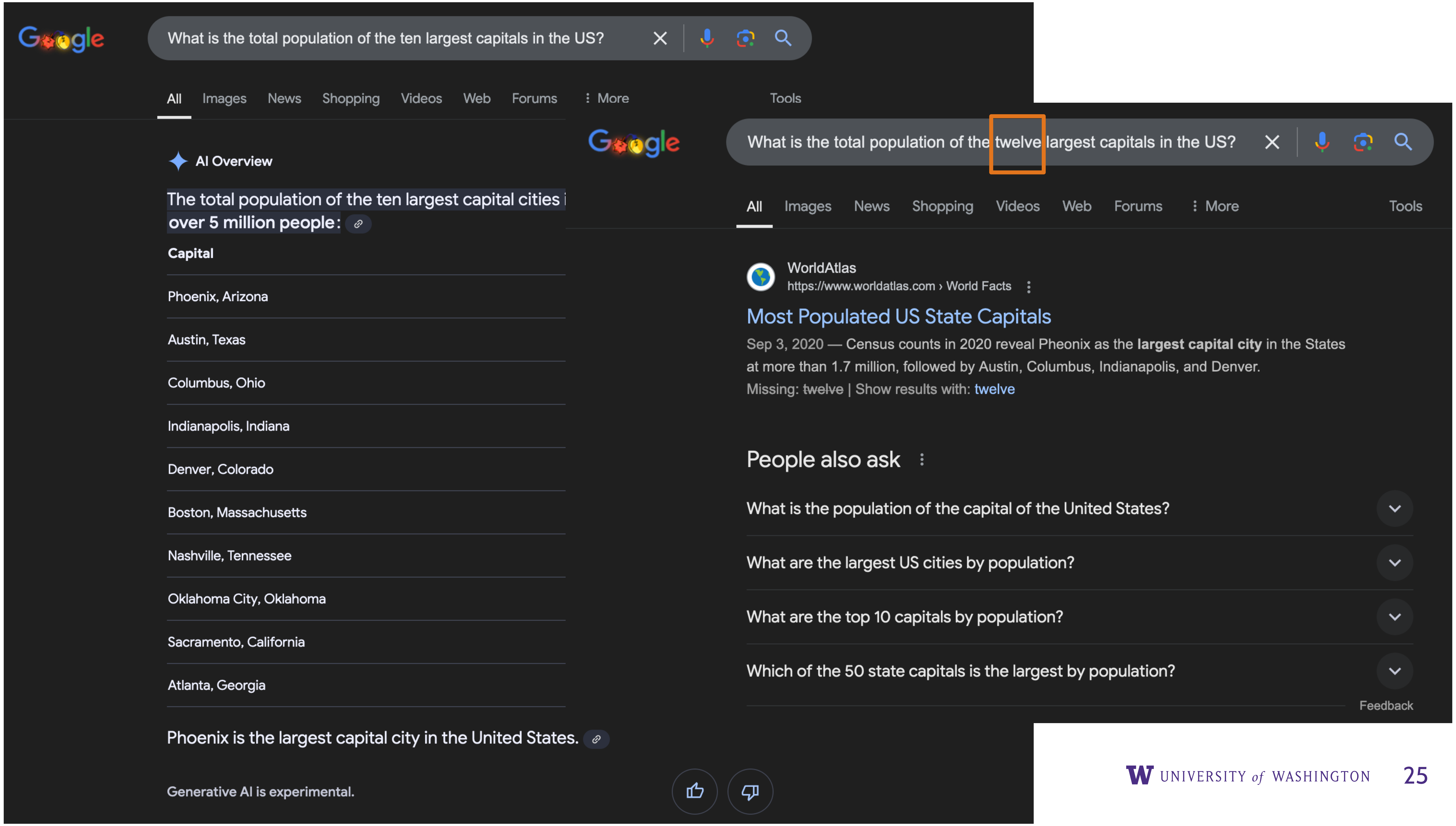
Notes:

- Metro area population, not city
- Because of a document it found
- Not connected to structured data source

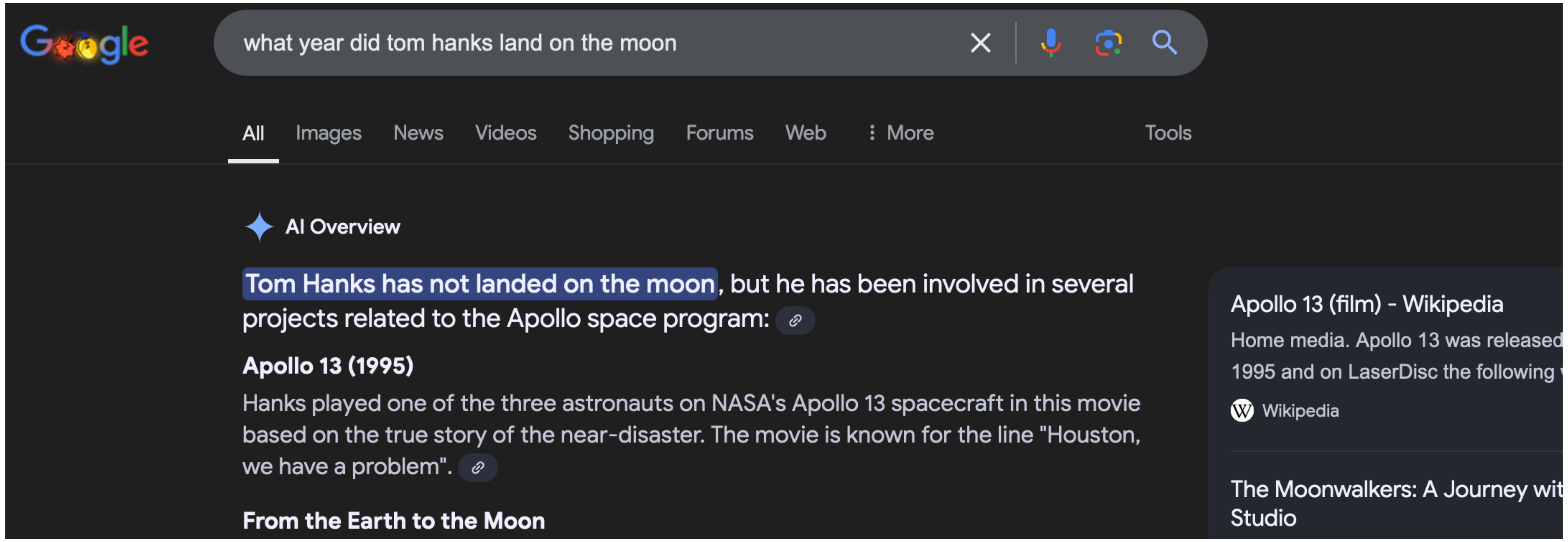
Re-visiting some questions



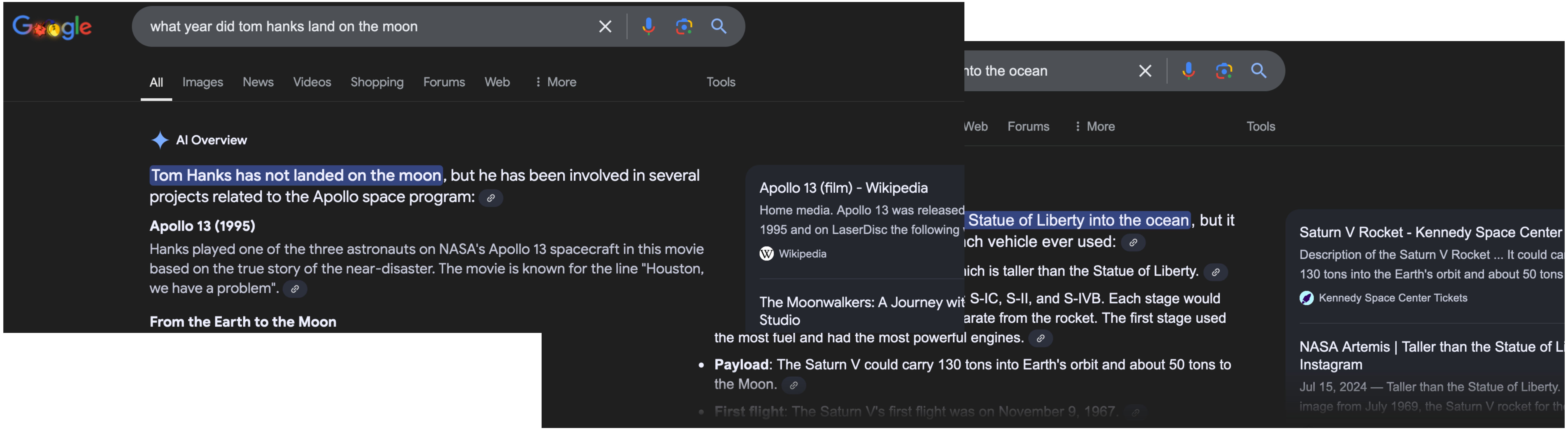
Re-visiting some questions



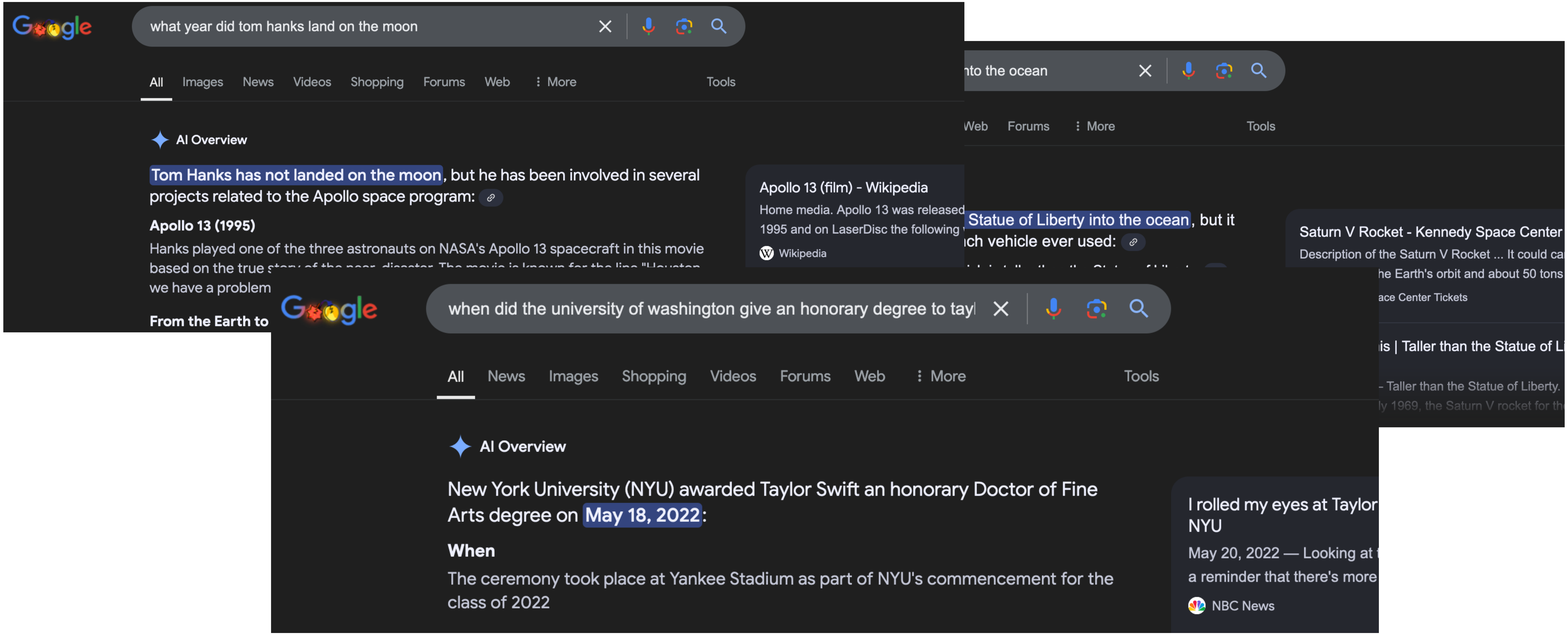
Re-visiting some questions



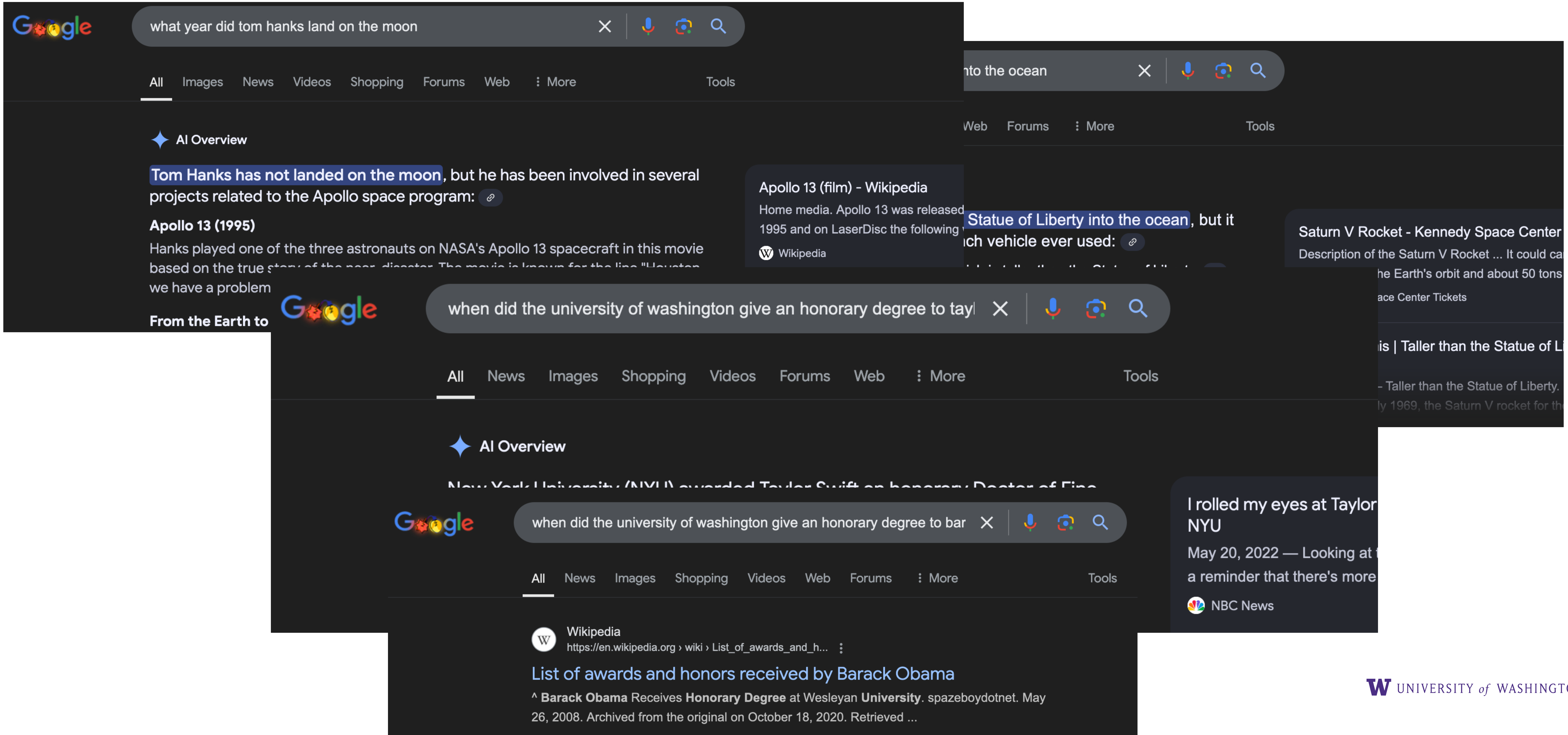
Re-visiting some questions



Re-visiting some questions



Re-visiting some questions



Conclusions

- Deep processing for QA
  - Exploits parsing, semantics, anaphora, reasoning
  - Computationally expensive
    - More tractable when applied only to questions and passages
- Systems trending toward greater use of:
  - Web resources: Wikipedia, answer repositories
  - Machine learning! Possible over-reliance on LLMs vs structured data (see also Shah and Bender 2022)
- But still: real use of deep representations and processing thereof, even in the LLM era

Next Time

Next Time

- More on current directions (e.g. unsupervised learning)

Next Time

- More on current directions (e.g. unsupervised learning)
- Summary / wrap-up

Next Time

- More on current directions (e.g. unsupervised learning)
- Summary / wrap-up
- AMA / general discussion

Next Time

- More on current directions (e.g. unsupervised learning)
- Summary / wrap-up
- AMA / general discussion
  - Submit questions here!

Next Time

- More on current directions (e.g. unsupervised learning)
- Summary / wrap-up
- AMA / general discussion
  - Submit questions here!
  - https://forms.gle/qiZ22sedhWXtyoGS9

Next Time

- More on current directions (e.g. unsupervised learning)
- Summary / wrap-up
- AMA / general discussion
  - Submit questions here!
  - https://forms.gle/qiZ22sedhWXtyoGS9
- Course evaluation!

Bonus Slides: Neural Approaches to Coreference and WSD

End-to-End Neural Coreference Resolution

Lee e. al, 2017

End-to-End Neural Coreference Resolution

Lee et al, 2017

- Begin with dataset with gold mention clusters (aka chains)

"General Electric said the Postal Service contacted the company."

End-to-End Neural Coreference Resolution

[Lee et al, 2017]

- Begin with dataset with gold mention clusters (aka chains)

"General Electric said the Postal Service contacted the company."

End-to-End Neural Coreference Resolution

Lee et al, 2017

- Can think of the coref problem as finding the maximally likely distribution:

$$P(y_1, \ldots, y_N \mid D) = \prod_{i=1}^{N} \frac{\exp\left(s\left(i, y_i\right)\right)}{\sum_{y' \in y(i)} \exp\left(s\left(i, y'\right)\right)}$$

End-to-End Neural Coreference Resolution

Lee et al, 2017

- Can think of the coref problem as finding the maximally likely distribution:

$$P(y_1, \ldots, y_N | D) = \prod_{i=1}^{N} \frac{\exp\left(s\left(i, y_i\right)\right)}{\sum_{y' \in y(i)} \exp\left(s\left(i, y'\right)\right)}$$

Where

$$s(i,j) = \begin{cases} 0 & j = \epsilon \\ s_m(i) + s_m(j) + s_a(i,j) & j \neq \epsilon \end{cases}$$

End-to-End Neural Coreference Resolution

Lee et al, 2017

- Can think of the coref problem as finding the maximally likely distribution:

$$P(y_1, \ldots, y_N \mid D) = \prod_{i=1}^{N} \frac{\exp\left(\boxed{s}(i, y_i)\right)}{\sum_{y' \in y(i)} \exp\left(\boxed{s}(i, y')\right)}$$

Where

$$\boxed{s(i,j)} = \begin{cases} 0 & j = \epsilon \\ s_m(i) + s_m(j) + s_a(i,j) & j \neq \epsilon \end{cases}$$

Coref Score

End-to-End Neural Coreference Resolution

Lee et al, 2017

- Can think of the coref problem as finding the maximally likely distribution:

$$P(y_1, \ldots, y_N | D) = \prod_{i=1}^{N} \frac{\exp\left(\boxed{s}(i, y_i)\right)}{\sum_{y' \in y(i)} \exp\left(\boxed{s}(i, y')\right)}$$

Where

$$\boxed{s(i,j)} = \begin{cases} 0 & j = \epsilon \\ \boxed{s_m}(i) + \boxed{s_m}(j) + s_a(i,j) & j \neq \epsilon \end{cases}$$

Coref Score Mention Score

End-to-End Neural Coreference Resolution

Lee et al, 2017

- Can think of the coref problem as finding the maximally likely distribution:

$$P(y_1, \ldots, y_N | D) = \prod_{i=1}^{N} \frac{\exp\left(\boxed{s}(i, y_i)\right)}{\sum_{y' \in y(i)} \exp\left(\boxed{s}(i, y')\right)}$$

Where

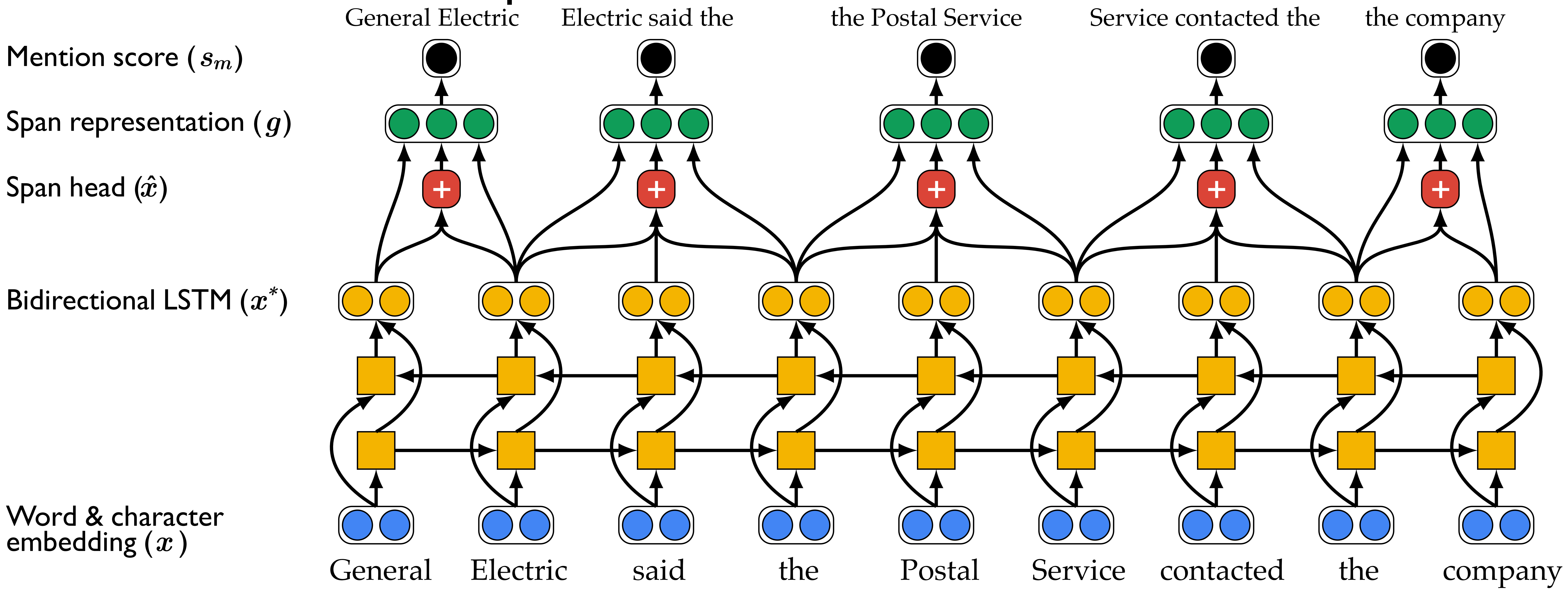
$$\boxed{s(i,j)} = \begin{cases} 0 & j = \epsilon \\ \boxed{s_m}(i) + \boxed{s_m}(j) + \boxed{s_a}(i,j) & j \neq \epsilon \end{cases}$$

Coref Score   Mention Score   Antecedent Score

End-to-End Neural Coreference Resolution

Lee et al, 2017

- **Step 1** — Train model to identify spans based on gold span labels
  - Use bi-LSTMs to model sequential information preceding/following/within spans
  - Include "headedness" of span with a learned ***attention*** mechanism



End-to-End Neural Coreference Resolution

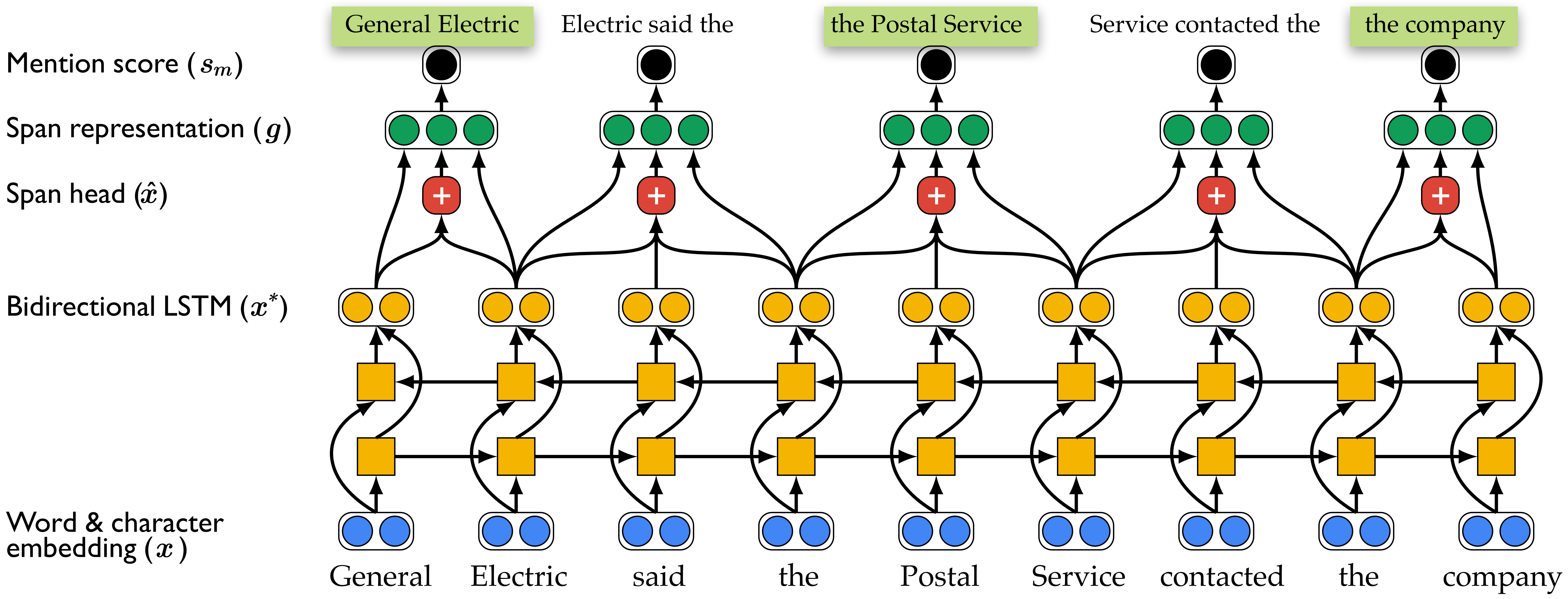
[Lee et al, 2017]

- ***Attention*** can be visualized by heatmap over spans:
  - (The flight attendants) have until 6:00 today to ratify labor concessions. (The pilots') union and ground crew did so yesterday.
  - (Prince Charles and his new wife Camilla) have jumped across the pond and are touring the United States making (their) first stop today in New York. It's Charles' first opportunity to showcase his new wife, but few Americans seem to care. Here's Jeanie Mowth. What a difference two decades make. (Charles and Diana) visited a JC Penney's on the prince's last official US tour. Twenty years later, here's the prince with his new wife.

End-to-End Neural Coreference Resolution

[Lee et al, 2017]

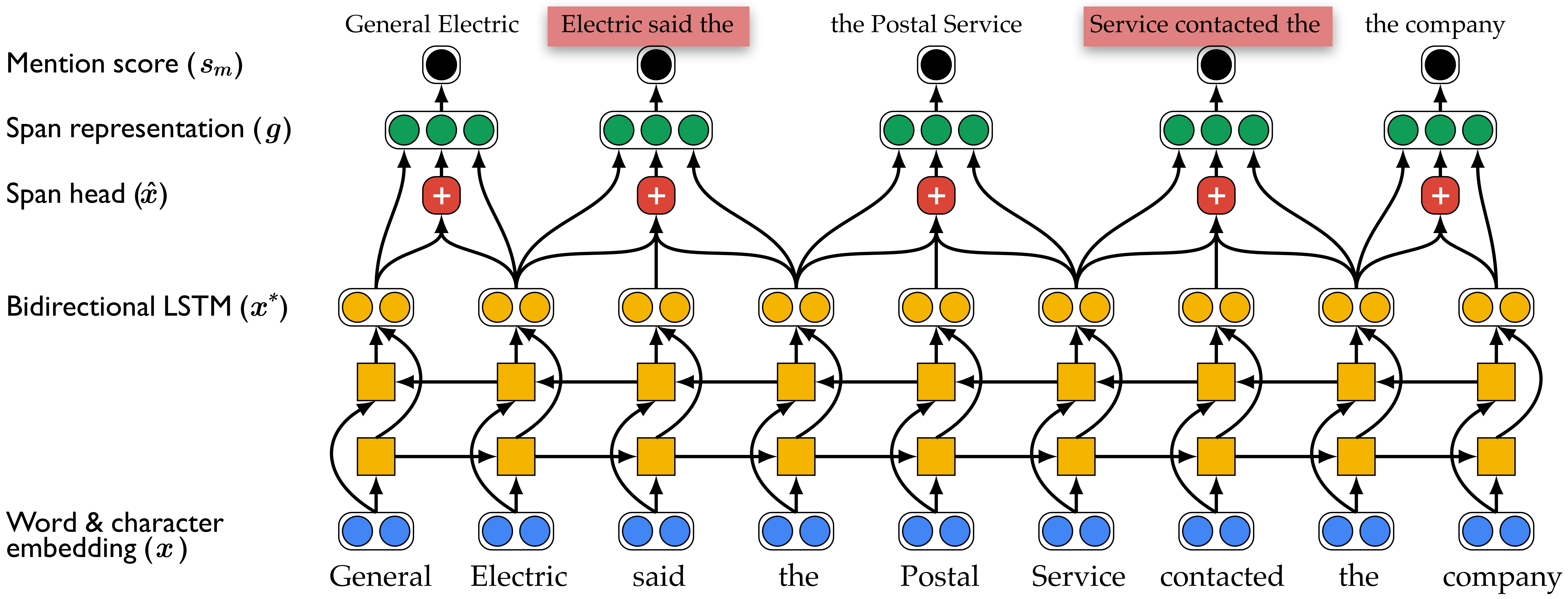
- **These** are valid gold mentions (network gets “reward” for getting these right)



End-to-End Neural Coreference Resolution

[Lee et al, 2017]

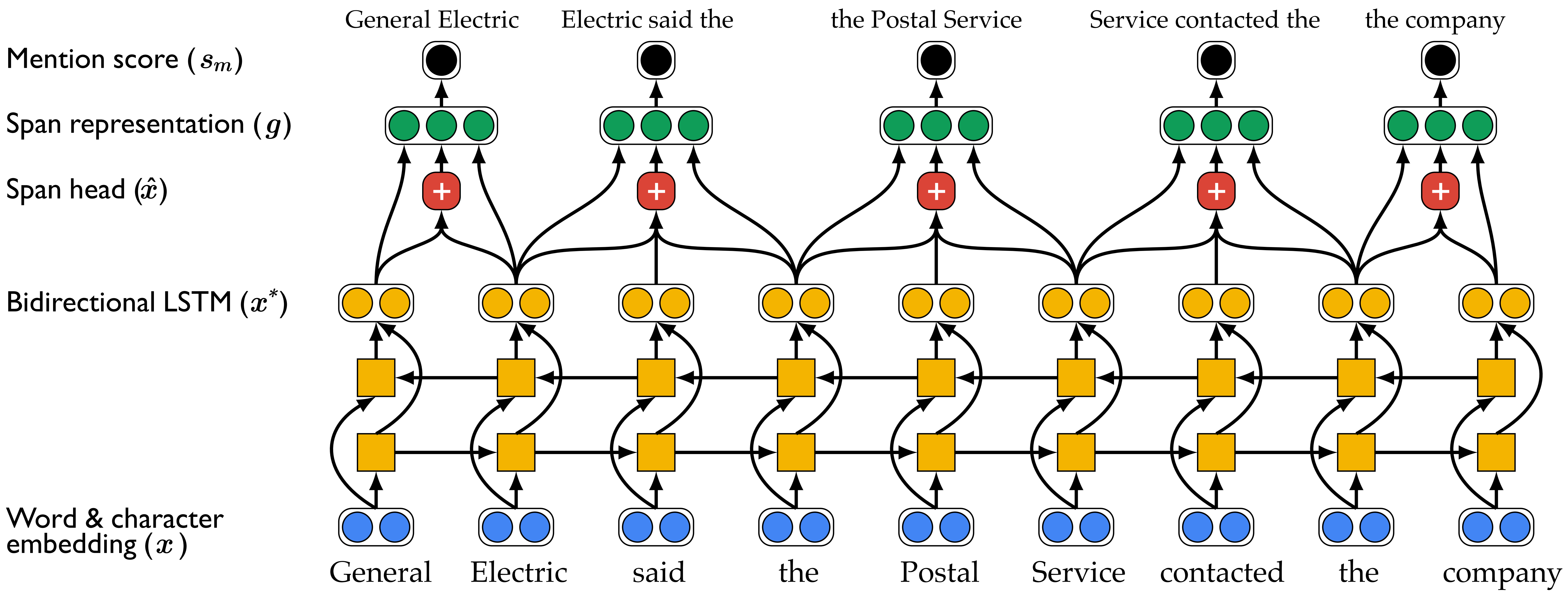
- **These** are invalid mentions (network accumulates error if these are selected)



End-to-End Neural Coreference Resolution

Lee et al, 2017

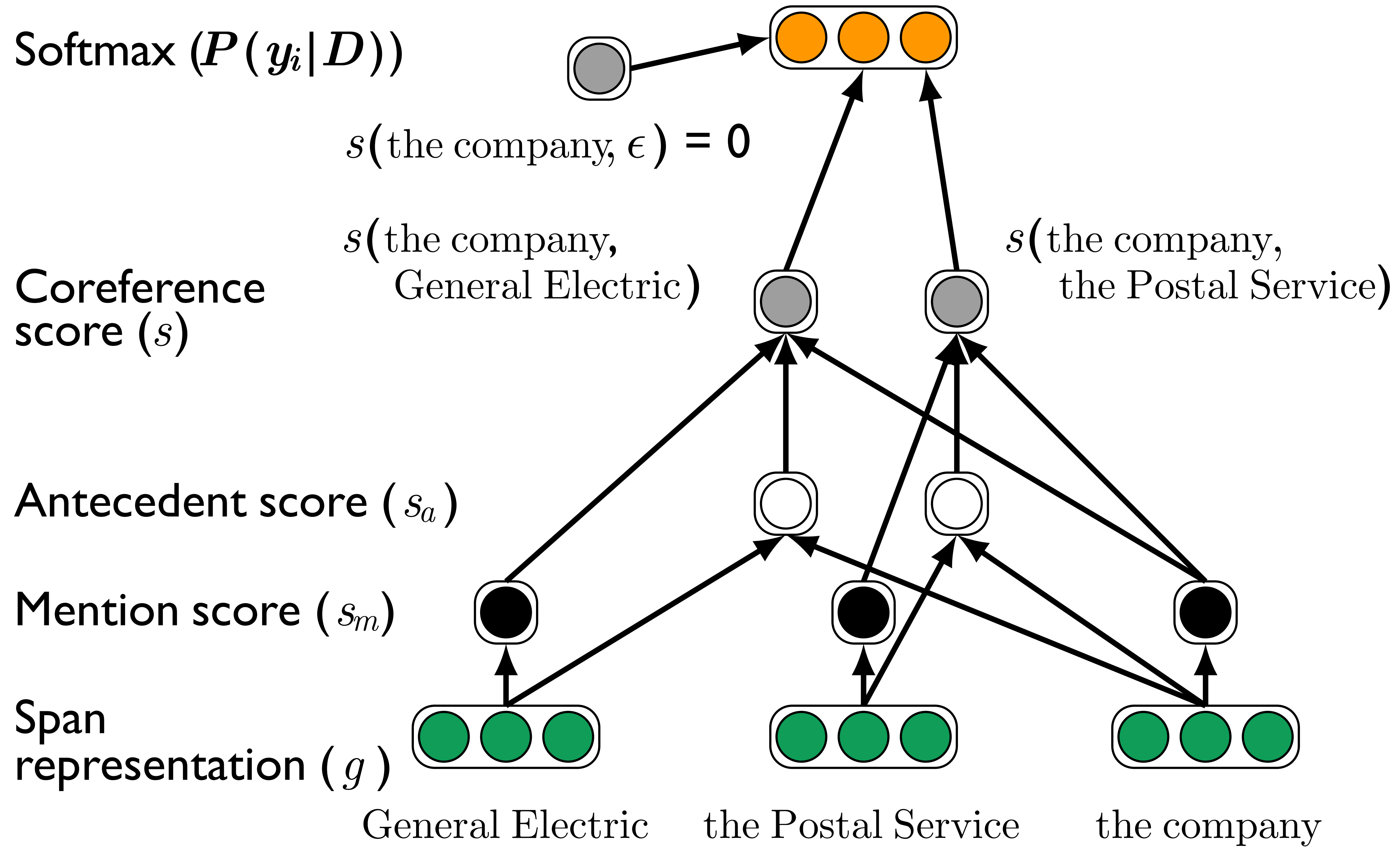
- Network thus learns to identify features from (embeddings → sequence) + head
  - As more or less likely to identify a span of words as a mention



End-to-End Neural Coreference Resolution

Lee et al, 2017

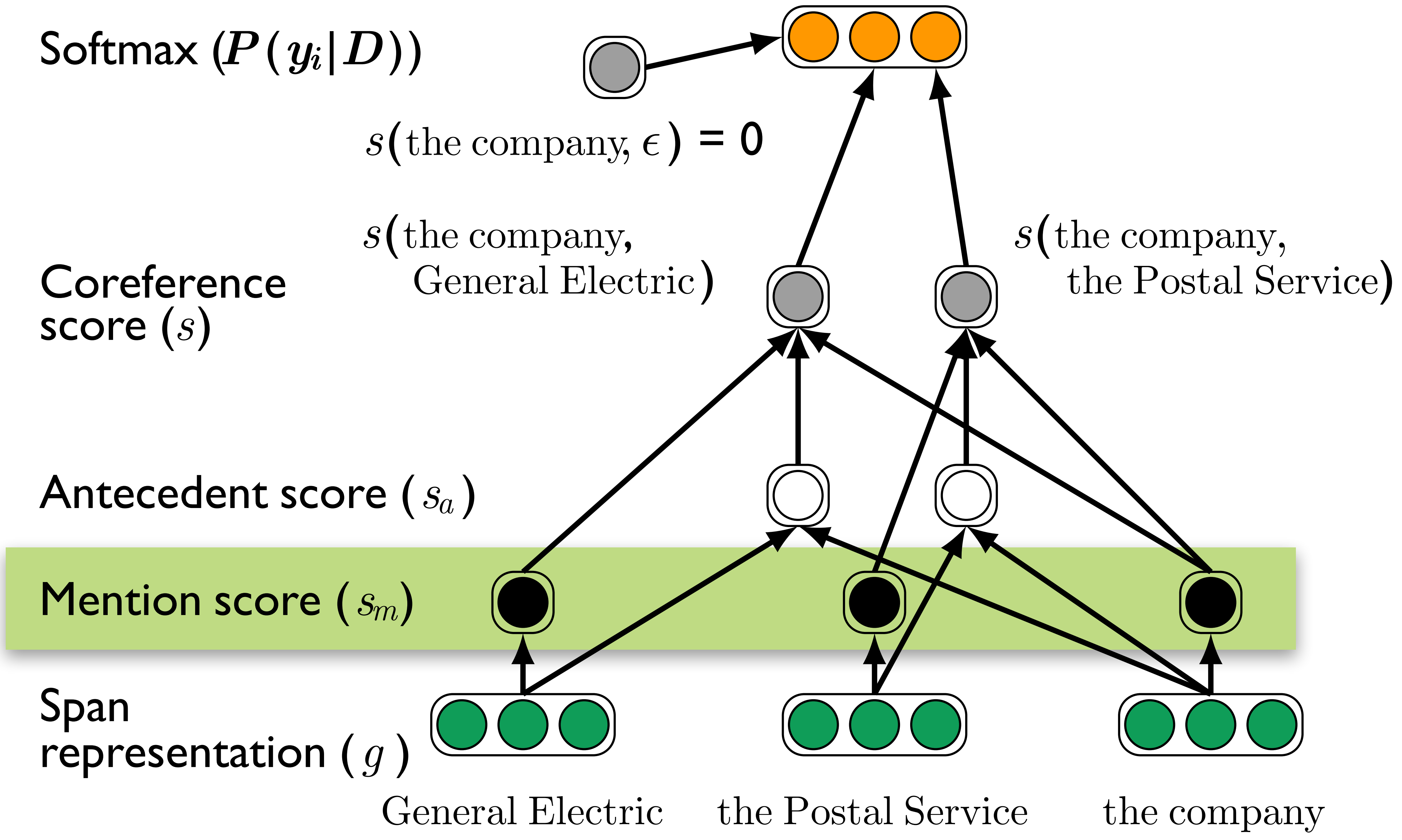
- **Step 2** — Learn Coref Clusters



End-to-End Neural Coreference Resolution

Lee et al, 2017

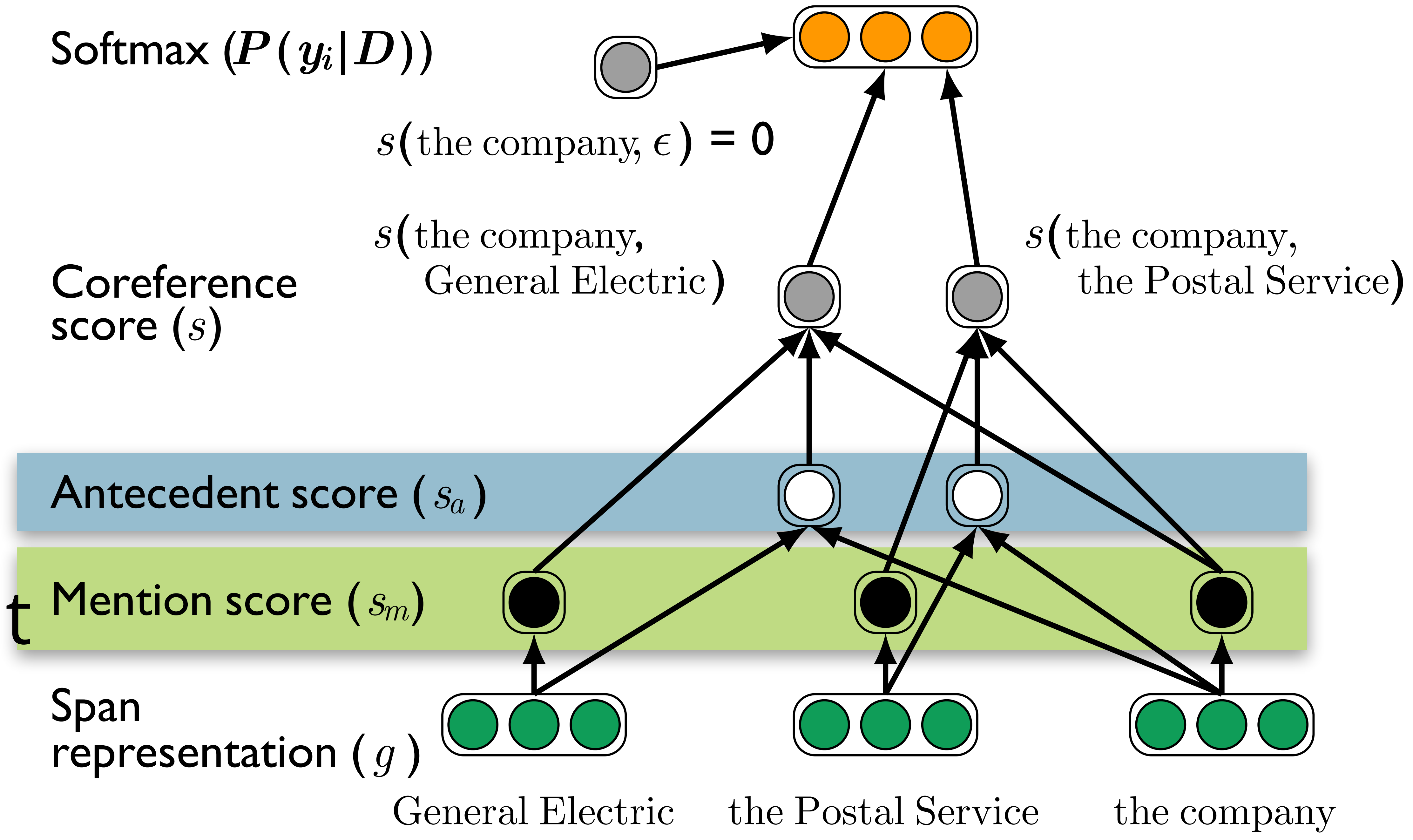
- **Step 2** — Learn Coref Clusters
- Mention Scores
  - Likelihood a given span is a mention
  - Unary over spans



End-to-End Neural Coreference Resolution

Lee et al, 2017

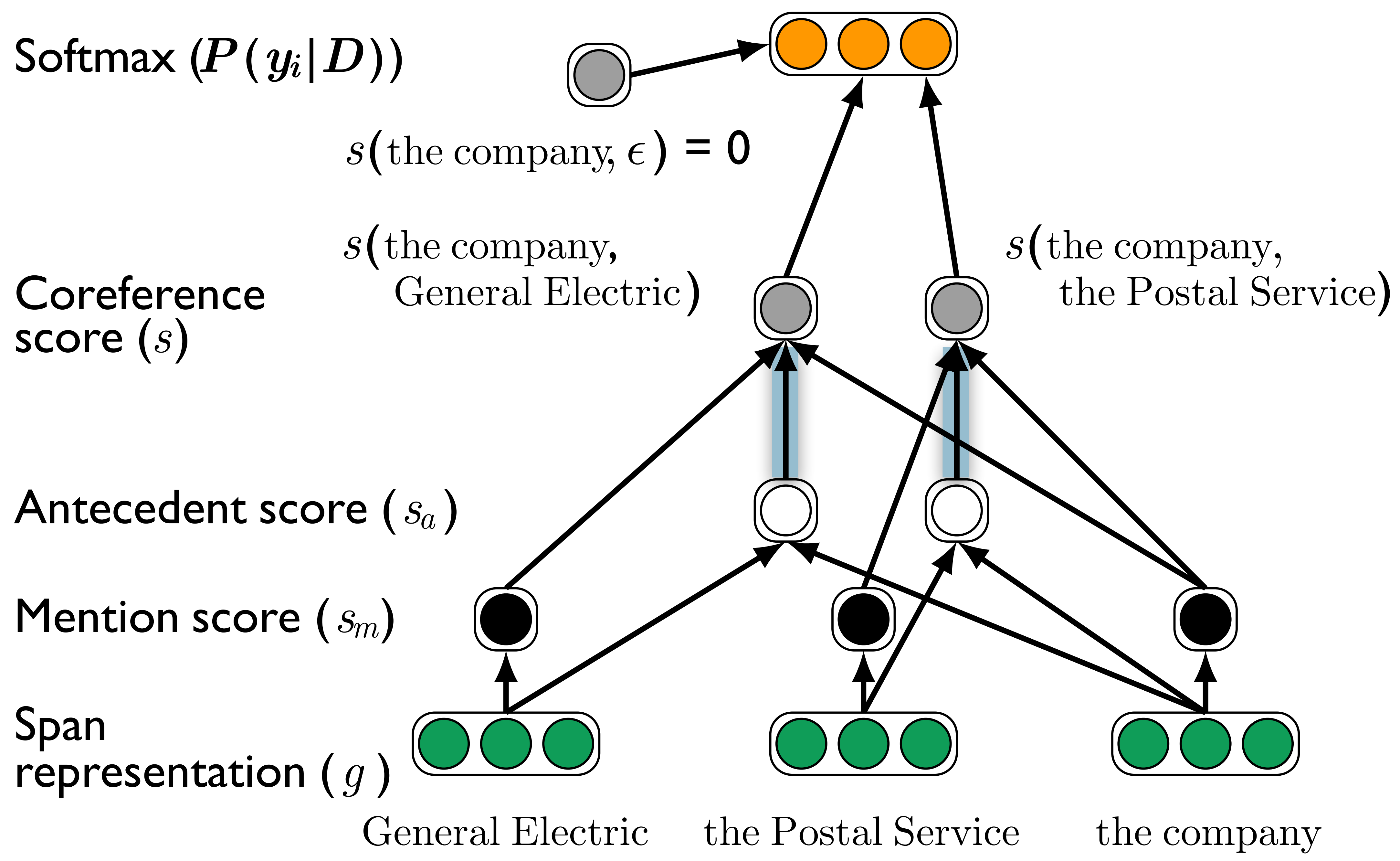
- **Step 2** — Learn Coref Clusters
- Mention Scores
  - Likelihood a given span is a mention
  - Unary over spans
- Antecedent scores
  - Likelihood another mention is antecedent
  - Pairwise between spans



End-to-End Neural Coreference Resolution

Lee et al, 2017

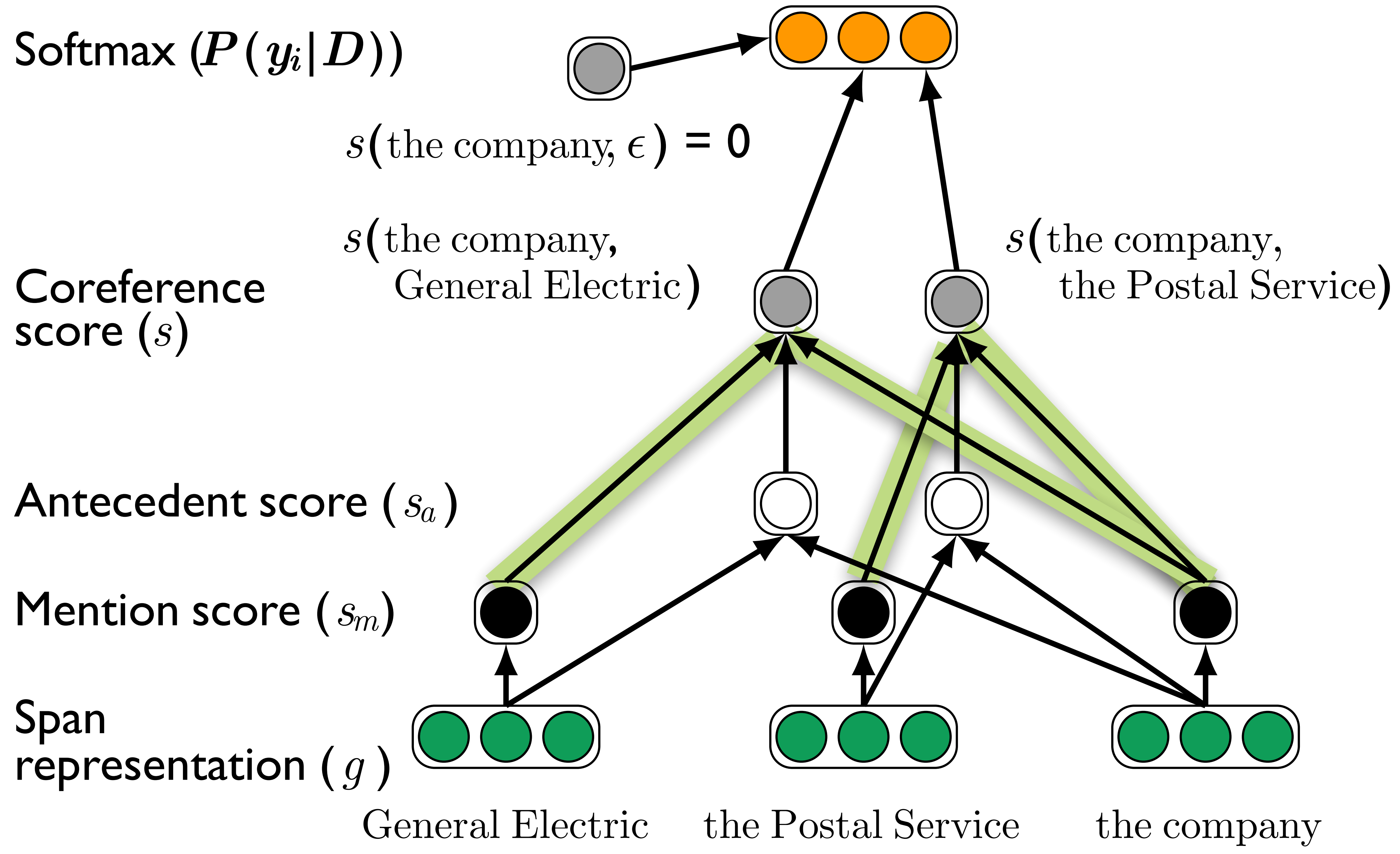
- The coref score is a combination of:
  - antecedent scores



End-to-End Neural Coreference Resolution

[Lee et al, 2017](#)

- The coref score is a combination of:
  - antecedent scores
  - mention scores



End-to-End Neural Coreference Resolution

Lee et al, 2017

- Other info:
  - Also implement pruning to avoid dealing with *all* spans
  - Also encode metadata, such as speaker and genre in mention representation

End-to-End Neural Coreference Resolution

Lee et al, 2017

- Data:
  - CoNLL-2012 Shared Task (Coref on OntoNotes)
  - 2802 training docs
  - 343 development docs
  - 348 test docs
  - 454 words/doc average

End-to-End Neural Coreference Resolution

Lee et al, 2017

- Positive:
  - State-of-the-art on CoNLL-2012 Test Data
- Errors:
  - Word embeddings tend to conflate paraphrasing with relatedness
    - e.g. (The flight attendants) have until 6:00 today to ratify labor concessions. (The pilots') union and ground crew did so yesterday.
  - (Prince Charles and his new wife Camilla) have jumped across the pond … What a difference two decades make. (Charles and Diana) visited a JC Penney's on the Prince's last official US tour. …

Neural Sequence Learning Models for Word Sense Disambiguation

Raganato et. al (2017b)

Neural Sequence Learning Models for WSD

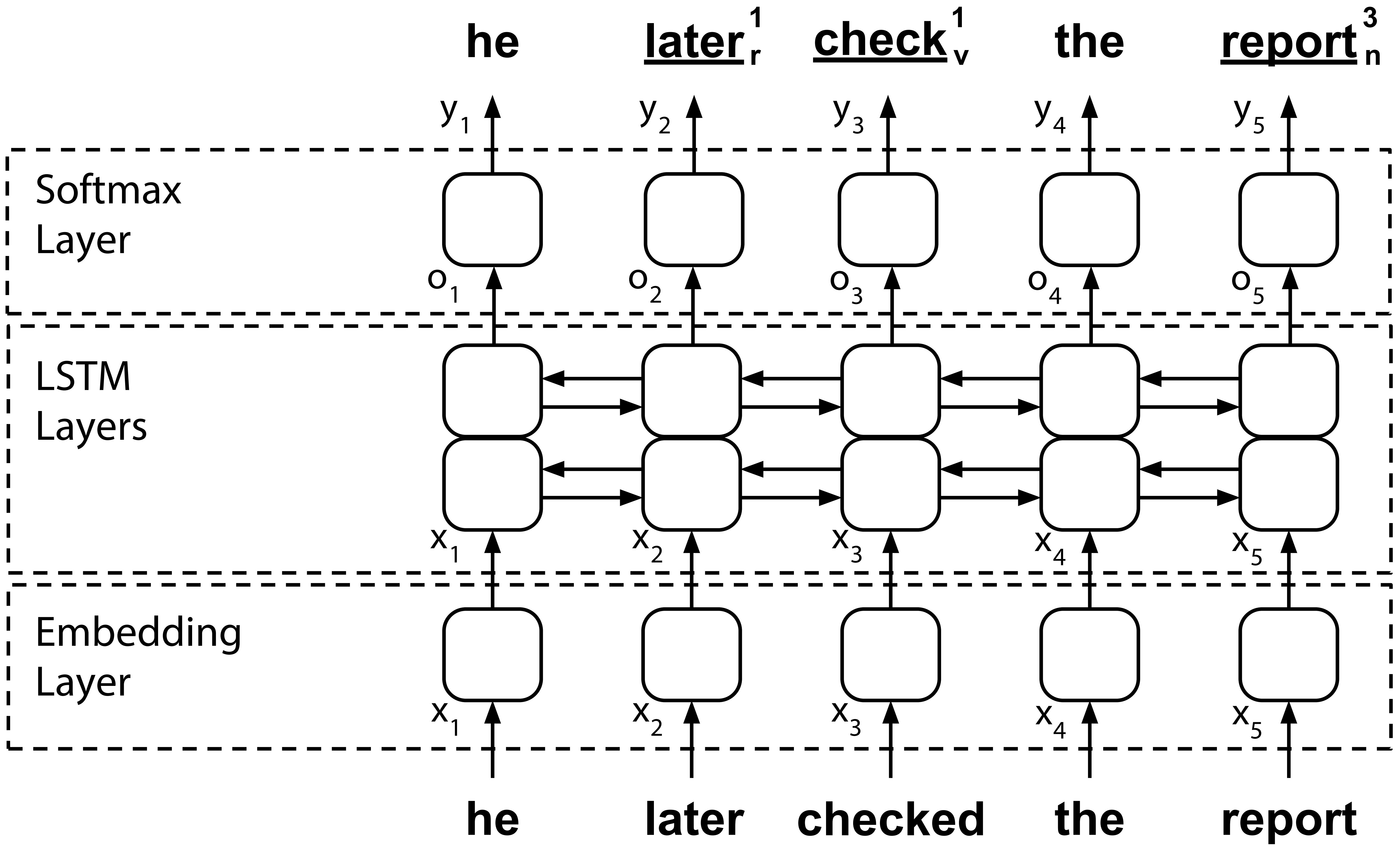
Raganato et. al (2017b)

- Authors propose several models for encoding words and senses
  - **bi-LSTM**
  - **bi-LSTM + Attention**
  - **Sequence to Sequence**
- All approaches are encoding sequential information
- All approaches use sense-tagged corpus

Neural Sequence Learning Models for WSD

Raganato et. al (2017b)

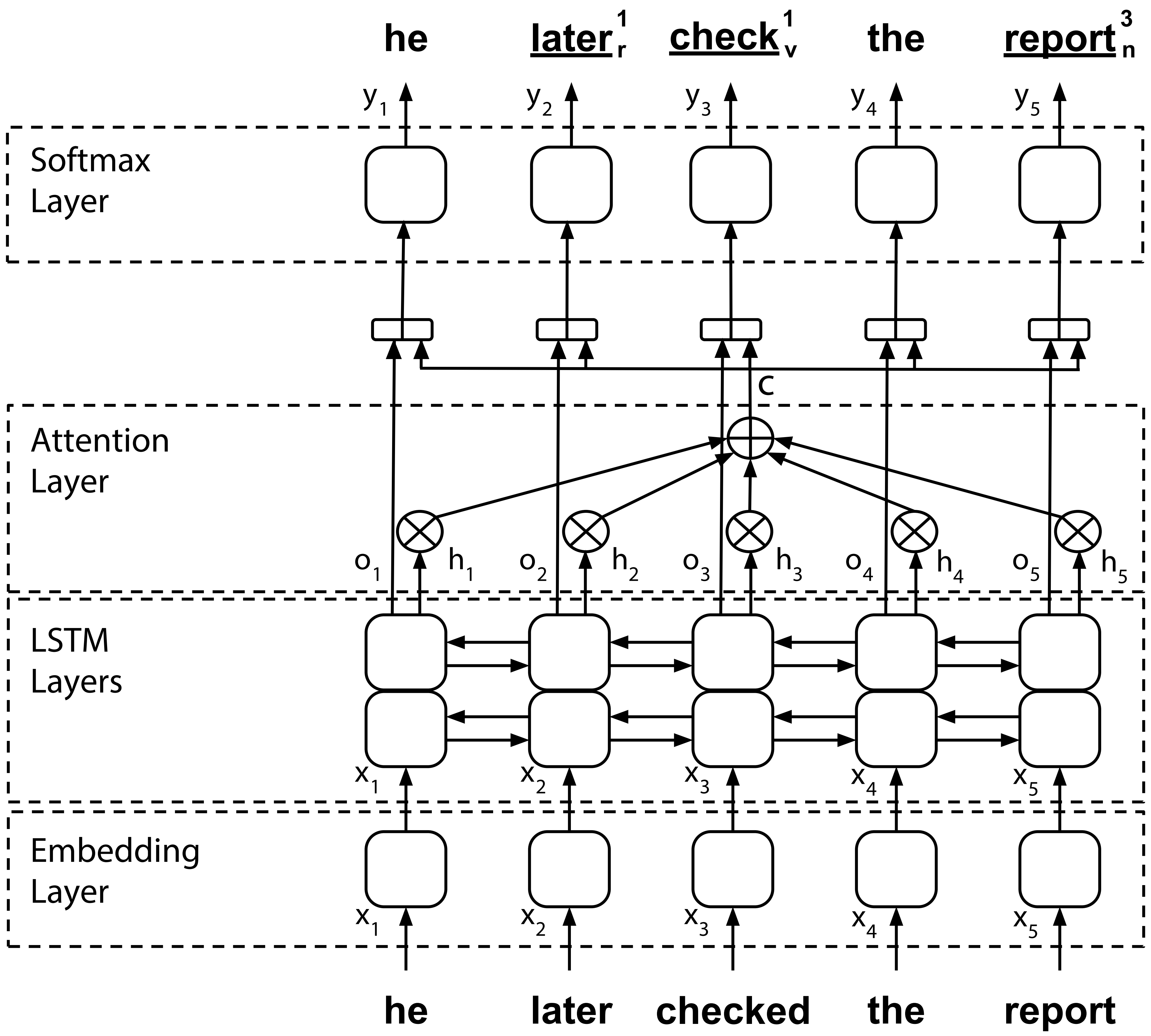
- **bi-LSTM**
  - Learn to label proper sense given word embedding and context (LSTM)



Neural Sequence Learning Models for WSD

Raganato et. al (2017b)

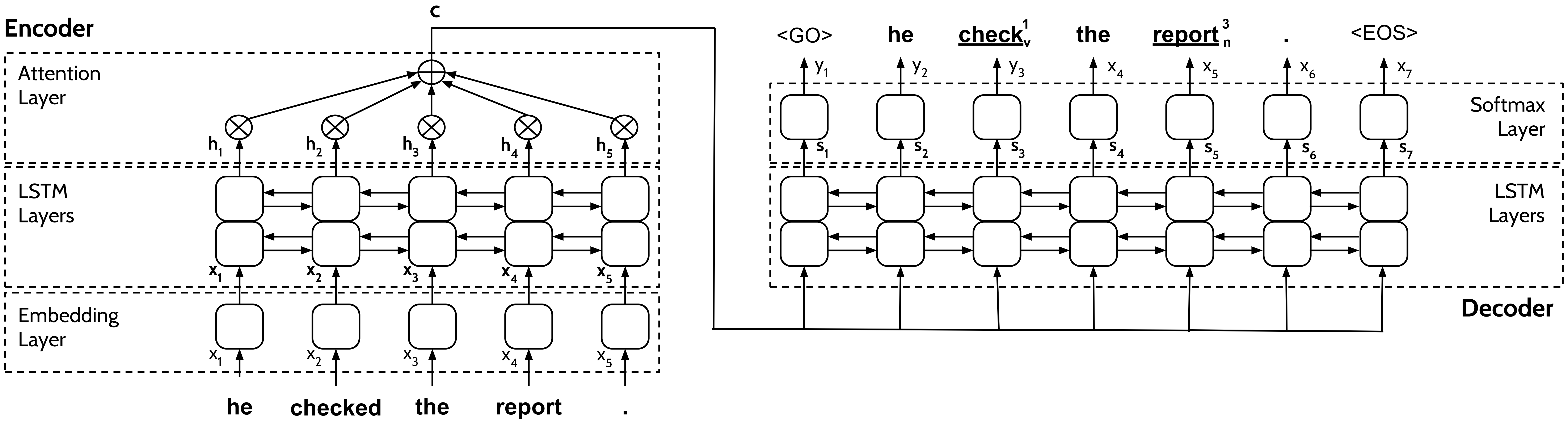
- **bi-LSTM + Attention**
  - Attention layer adds sentence-level representation *c* to guide the labels generate at each sequence time step by focusing on what part of the sentence may be relevant
    - (e.g. with *wicket* in focus, *match* might be influenced toward the game sense, rather than firestarter)



Neural Sequence Learning Models for WSD

Raganato et. al (2017b)

- **seq2seq**
  - Two-step task:
    - Memorization — Model is trained to replicate input token-by-token
    - Disambiguation — Model learns to replace surface forms with appropriate senses



Neural Sequence Learning Models for WSD

Raganato et. al (2017b)

- Also try models that jointly learn WSD and:
  - coarse semantic labels
    - e.g. *noun.location*, *verb.motion*
  - POS tags
  - Both

Neural Sequence Learning Models for WSD

Raganato et. al (2017b)

- Data:
  - Use SemCor 3.0 for training/evaluating word senses

Neural Sequence Learning Models for WSD

Raganato et. al (2017b)

- Results:

	Dev	Test Datasets				Concatenation of All Test Datasets				
	SE07	SE2	SE3	SE13	SE15	Nouns	Verbs	Adj.	Adv.	All
BLSTM	61.8	71.4	68.8	65.6	69.2	70.2	56.3	75.2	**84.4**	68.9
BLSTM + att.	62.4	71.4	**70.2**	66.4	70.8	71.0	**58.4**	75.2	83.5	69.7
BLSTM + att. LEX	63.7	**72.0**	69.4	66.4	**72.4**	**71.6**	57.1	**75.6**	83.2	**69.9**
BLSTM + att. LEX +POS	**64.8**	**72.0**	69.1	**66.9**	71.5	71.5	57.5	75.0	83.8	**69.9**
Seq2Seq	60.9	68.5	67.9	65.3	67.0	68.7	54.5	74.0	81.2	67.3
Seq2Seq + att.	62.9	69.9	69.6	65.6	67.7	69.5	57.2	74.5	81.8	68.4
Seq2Seq + att. LEX	64.6	70.6	67.8	66.5	68.7	70.4	55.7	73.3	82.9	68.5
Seq2Seq + att. LEX +POS	63.1	70.1	68.5	66.5	69.2	70.1	55.2	75.1	84.4	68.6
IMS	61.3	70.9	69.3	65.3	69.5	70.5	55.8	75.6	82.9	68.9
IMS+emb	**62.6**	**72.2**	**70.4**	65.9	71.5	**71.9**	56.6	**75.9**	**84.7**	**70.1**
Context2Vec	61.3	71.8	69.1	65.6	**71.9**	71.2	**57.4**	75.2	82.7	69.6
Lesk_{ext} +emb	56.7	63.0	63.7	66.2	64.6	70.0	51.1	51.7	80.6	64.2
UKB_{gloss} w2w	42.9	63.5	55.4	62.9	63.3	64.9	41.4	69.5	69.7	61.1
Babelfy	51.6	67.0	63.5	**66.4**	70.3	68.9	50.7	73.2	79.8	66.4
MFS	54.5	65.6	66.0	63.8	67.1	67.7	49.8	73.1	80.5	65.5

Neural Sequence Learning Models for WSD

Raganato et. al (2017b)

- Analysis:
  - Comparable to other supervised systems
  - Adding coarse-grained lexical tags appears to help
  - POS did not seem to help
- ***None of these systems substantially better than using the Most Frequent Sense***