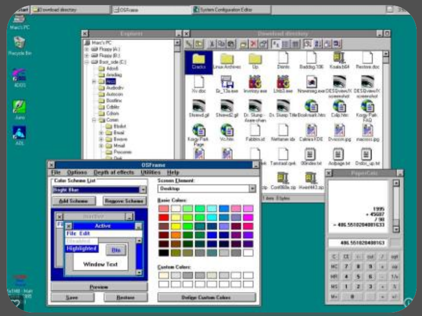
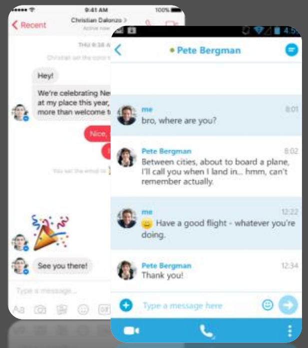
Building Information Bot (InfoBot) with Question Understanding, Answering & Asking

Nan Duan (段楠)
Natural Language Computing Group
Microsoft Research Asia

2017-11-08
Dalian

CCF ADL 第86期-自动问答、聊天机器人与自然语言理解

The world and technology are once again transforming.

	mid-80s	mid-90s	mid-00s	mid-10s
	PC	Web	Smartphone	Messaging
				
Platform Examples	Desktop OS Dos, Windows, Mac OS	Browser Mosaic, Explorer, Netscape	Mobile OS iOS, Android, WP	Messaging Apps Whatsapp, WeChat, Skype
Application Examples	Programs Excel, PPT, Lotus	Web Sites Yahoo, Amazon	Apps Angry Birds, Instagram	Bots / Agents Weather, Travel
UI	GUI Apps	Web Pages	Native Apps	Conversations

The Logical Architecture of Conversational Bot

Specific Knowledge

General Knowledge

Task-Oriented Dialogue System (TASK COMPLETION)

- Intent Detection/Slot Filling
- Dialogue Management
- Single-turn/Multi-turn

InfoBot (INFORMATION & ANSWERS)

- QA/QG
- Web Search
- Single-turn (Often)

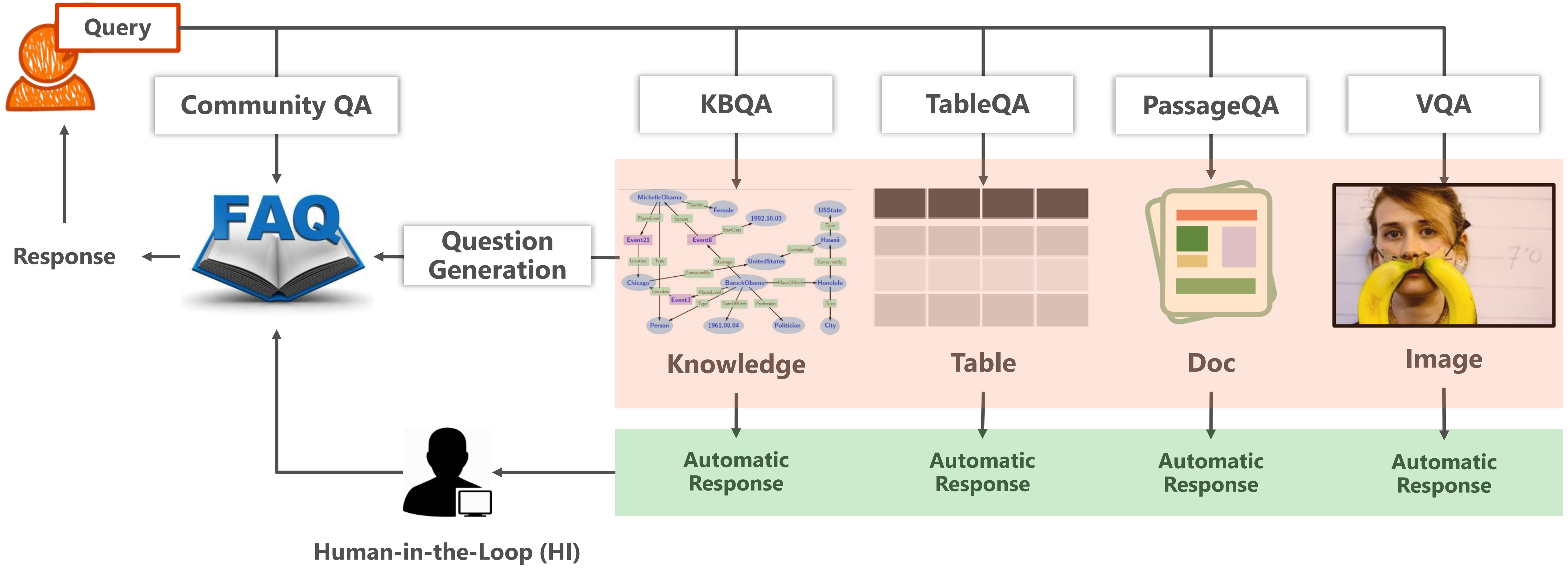
ChatBot (GENERAL CONVERSATION)

- Chit-chat
- Emotion/Sentiment Analysis
- Multi-turn (Most of time)

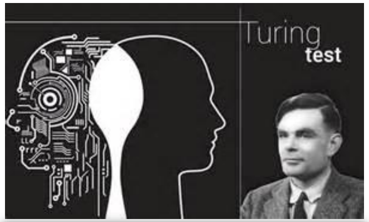
InfoBot Overview

Three core tasks in InfoBot

- Given a question, transform it into its logical form (**Question Understanding, QU**)
- Given a question, find its answer from existing database (**Question Answering, QA**)
- Given a piece of content, predict possible questions that can be answered by the content (**Question Generation, QG**)



Evolution of QA: Tasks & Systems (from 1950 to present)



Turing Test

Restricted Domain QA

Natural Language Interface to Database (NLIDB) – 1960s
- BASEBALL, 1961
- LUNAR, 1971

Reading Comprehension – 1970s
- QUALM, 1977
- …

Dialogue System – 1970s~1990s
- SHRDLU, 1972
- UNIX Consultant, 1988
- …

IBM Watson in Jeopardy!

Apple Siri

Amazon Echo

more names not in this list, due to space …

1950 | 1960s | 1970s | 1980s | 1990s | 1999-2007 | 2011 | 2014 | 2015 | 2016 | Present

Text REtrieval Conference (TREC), QA Task

TREC-8, 1999
- The first open domain QA task
- Answer document retrieval

TREC-9, 2000
- Answer span retrieval

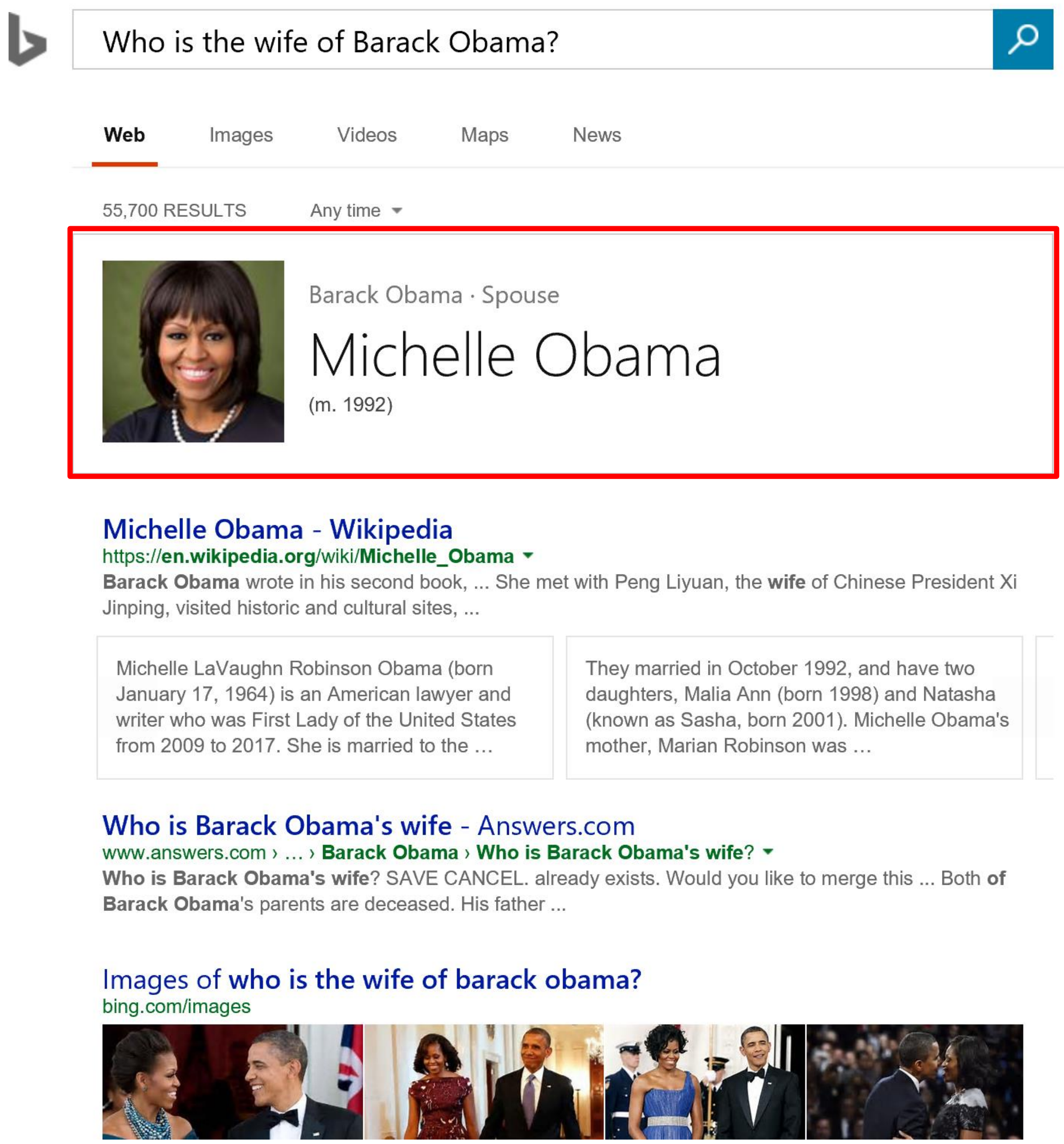
TREC, 2003
- Answer passage retrieval

…

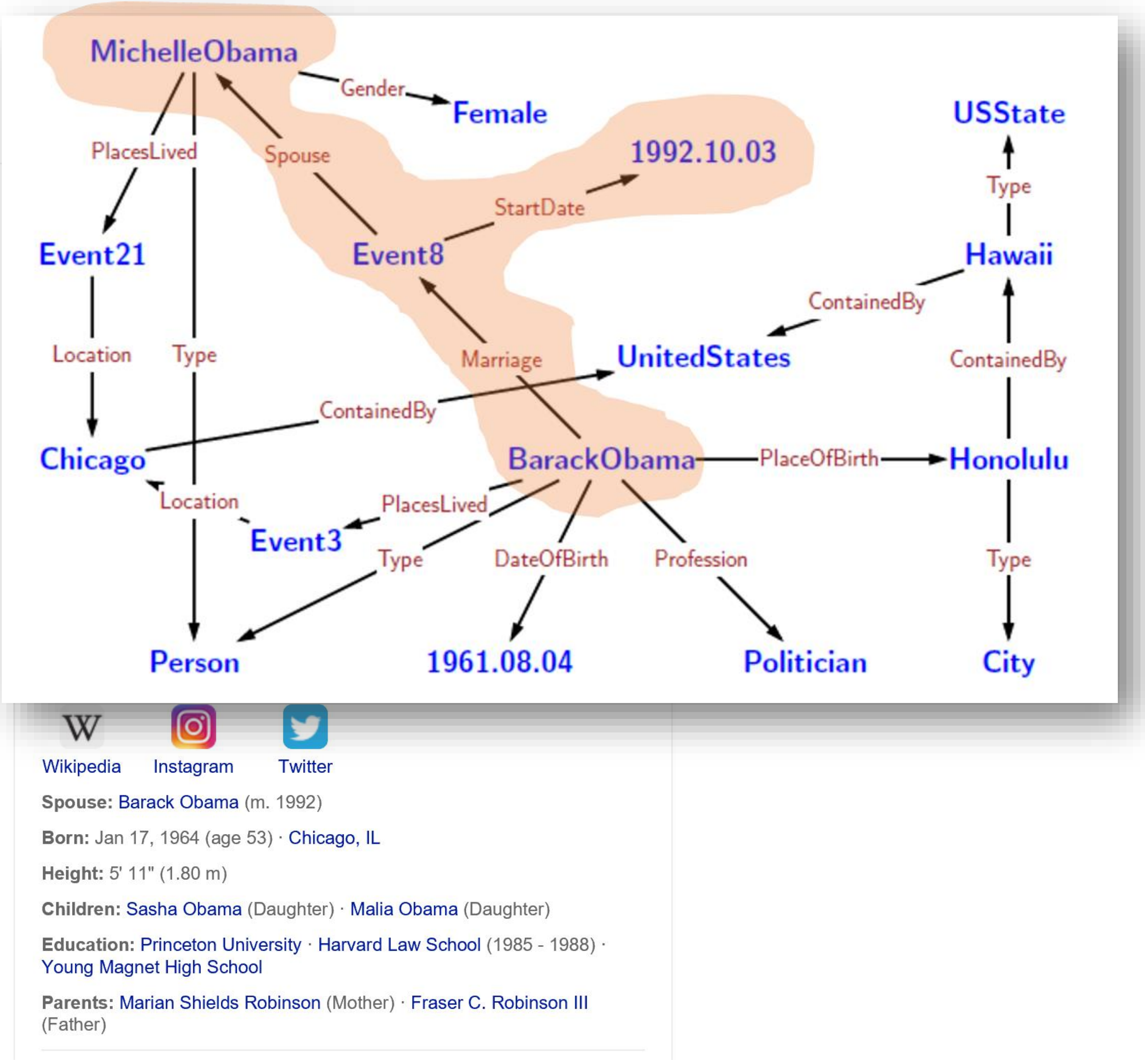
Microsoft Cortana & Xiaoice

Google Home

KBQA (知识图谱问答)

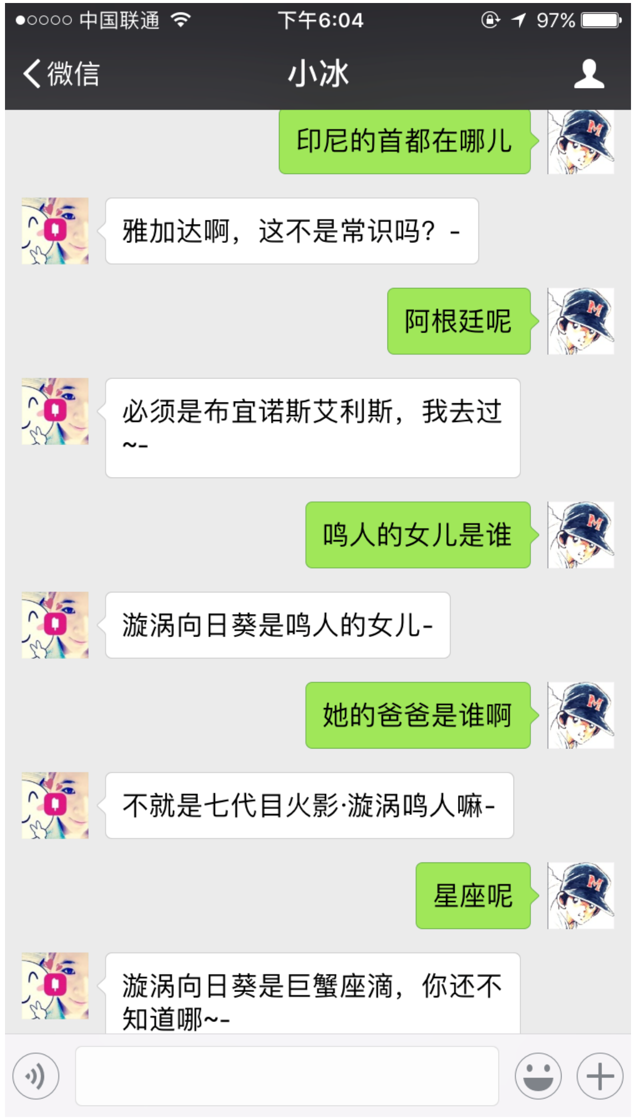


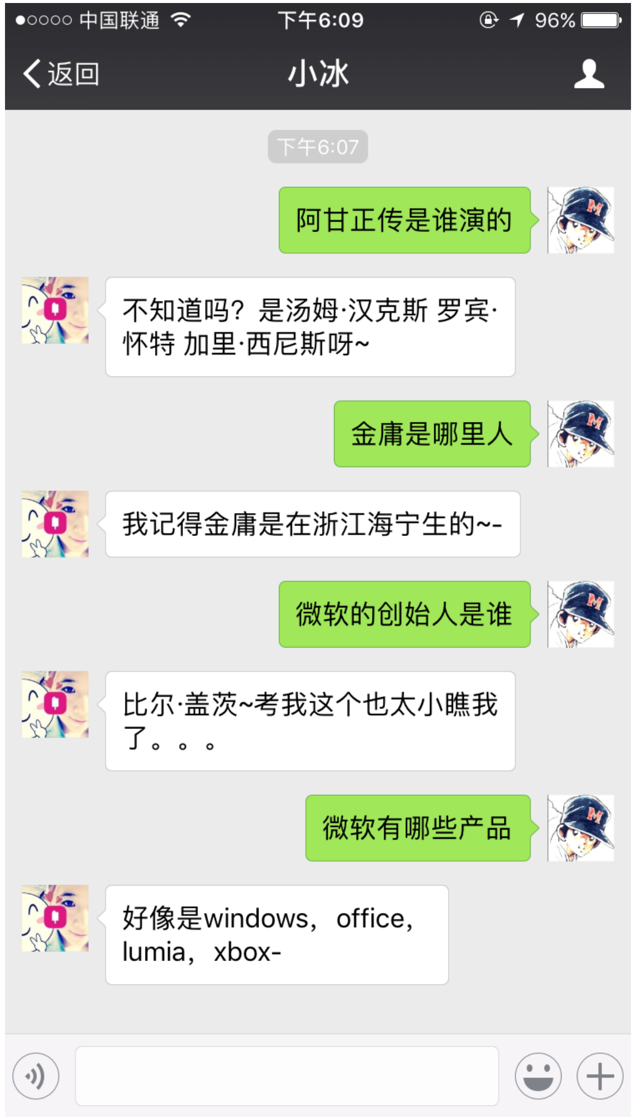
Who is the wife of Barack Obama?
Web
Images
Videos
Maps
News
55,700 RESULTS
Any time
Barack Obama · Spouse
Michelle Obama
(m. 1992)
Michelle Obama - Wikipedia
https://en.wikipedia.org/wiki/Michelle_Obama
Barack Obama wrote in his second book, ... She met with Peng Liyuan, the wife of Chinese President Xi Jinping, visited historic and cultural sites, ...
Michelle LaVaughn Robinson Obama (born January 17, 1964) is an American lawyer and writer who was First Lady of the United States from 2009 to 2017. She is married to the …
They married in October 1992, and have two daughters, Malia Ann (born 1998) and Natasha (known as Sasha, born 2001). Michelle Obama's mother, Marian Robinson was …
Who is Barack Obama's wife - Answers.com
www.answers.com › … › Barack Obama › Who is Barack Obama's wife?
Who is Barack Obama's wife? SAVE CANCEL. already exists. Would you like to merge this ... Both of Barack Obama's parents are deceased. His father ...
Images of who is the wife of barack obama?
bing.com/images



MichelleObama
Gender
Female
PlacesLived
Spouse
1992.10.03
USState
Type
StartDate
Event21
Event8
Hawaii
ContainedBy
Location
Type
Marriage
UnitedStates
ContainedBy
ContainedBy
Chicago
BarackObama
PlaceOfBirth
Honolulu
Location
PlacesLived
Event3
Type
DateOfBirth
Profession
Type
Person
1961.08.04
Politician
City
Wikipedia
Instagram
Twitter
Spouse: Barack Obama (m. 1992)
Born: Jan 17, 1964 (age 53) · Chicago, IL
Height: 5' 11" (1.80 m)
Children: Sasha Obama (Daughter) · Malia Obama (Daughter)
Education: Princeton University · Harvard Law School (1985 - 1988) · Young Magnet High School
Parents: Marian Shields Robinson (Mother) · Fraser C. Robinson III (Father)

KBQA (知识图谱问答)





Chinese Knowledge Graph (BingKnows)

微软 ||| 公司名称 ||| 微软公司
微软 ||| 外文名称 ||| Microsoft corporation.
微软 ||| 总部地点 ||| 美国华盛顿州雷德蒙市
微软 ||| 成立时间 ||| 1975年4月4日16时
微软 ||| 经营范围 ||| 操作系统，办公软件，手机
微软 ||| 公司性质 ||| 上市公司、外商独资
微软 ||| 公司口号 ||| 新效率(New Efficiency)
微软 ||| 年营业额 ||| 77,849百万美元（2014年）
微软 ||| 员工数 ||| 99,000人(2014年)
微软 ||| 联合创始人 ||| 比尔·盖茨、保罗·艾伦
微软 ||| 现任董事长 ||| 约翰·汤普森
微软 ||| 首席执行官 ||| 萨蒂亚·纳德拉
微软 ||| 首席运营官 ||| 凯文·特纳
微软 ||| 世界500强 ||| 第104位（2014年）
微软 ||| 成立地点 ||| 美国新墨西哥州阿尔伯克基市
微软 ||| 中国总部 ||| 中国北京海淀区知春路49号
微软 ||| 主要产品 ||| xbox，windows，office，lumia
董明珠(珠海格力集团有限公司原董事长) ||| 中文名 ||| 董明珠
董明珠(珠海格力集团有限公司原董事长) ||| 外文名 ||| Mingzhu Dong
董明珠(珠海格力集团有限公司原董事长) ||| 别名 ||| 东方明珠
董明珠(珠海格力集团有限公司原董事长) ||| 国籍 ||| 中华人民共和国
董明珠(珠海格力集团有限公司原董事长) ||| 民族 ||| 汉
董明珠(珠海格力集团有限公司原董事长) ||| 出生地 ||| 江苏南京
董明珠(珠海格力集团有限公司原董事长) ||| 出生日期 ||| 1954年8月
董明珠(珠海格力集团有限公司原董事长) ||| 职业 ||| 格力电器董事长兼总裁
董明珠(珠海格力集团有限公司原董事长) ||| 毕业院校 ||| 芜湖职业技术学院
董明珠(珠海格力集团有限公司原董事长) ||| 主要成就 ||| 全球100位最佳CEO
董明珠(珠海格力集团有限公司原董事长) ||| 代表作品 ||| 《棋行天下》

TableQA (表格问答)

star trek rts

Web Images Videos Maps News

240,000 RESULTS Any time

Star Trek Games for the PC

	Game	Genre	Year	Metascore
9	Star Trek: Armada II	RTS	2001	65
10	Star Trek: Away Team	RTS	2001	64
	Star Trek: Deep Space Nine: Dominion Wars	RTS	2001	64
16	Star Trek: New Worlds	RTS	2000	52

38 more rows, 2 more columns

Best and Worst Star Trek Videogames - Metacritic
www.metacritic.com/feature/best-and-worst-star-trek-videogames

Improve this answer · Is this answer helpful?

Star Trek: Armada - Wikipedia
https://en.wikipedia.org/wiki/Star_Trek:_Armada
Star Trek: Armada is a **real-time strategy** video game developed and published in 2000 by Activision. The game's look and feel is based primarily on **Star Trek**: The Next ...
Gameplay · Plot · Development · Reception

Find more within this Wikipedia page

Best and Worst Star Trek Videogames - Metacritic
www.metacritic.com/feature/best-and-worst-star-trek-videogames
The new **Star Trek** Online is only the latest in a storied line of **Star Trek**-related videogames dating back to the early 1970s. Voyage with us as we chart a course ...

Star Trek: Armada - IGN

http://www.metacritic.com/feature/best-and-worst-star-trek-videogames

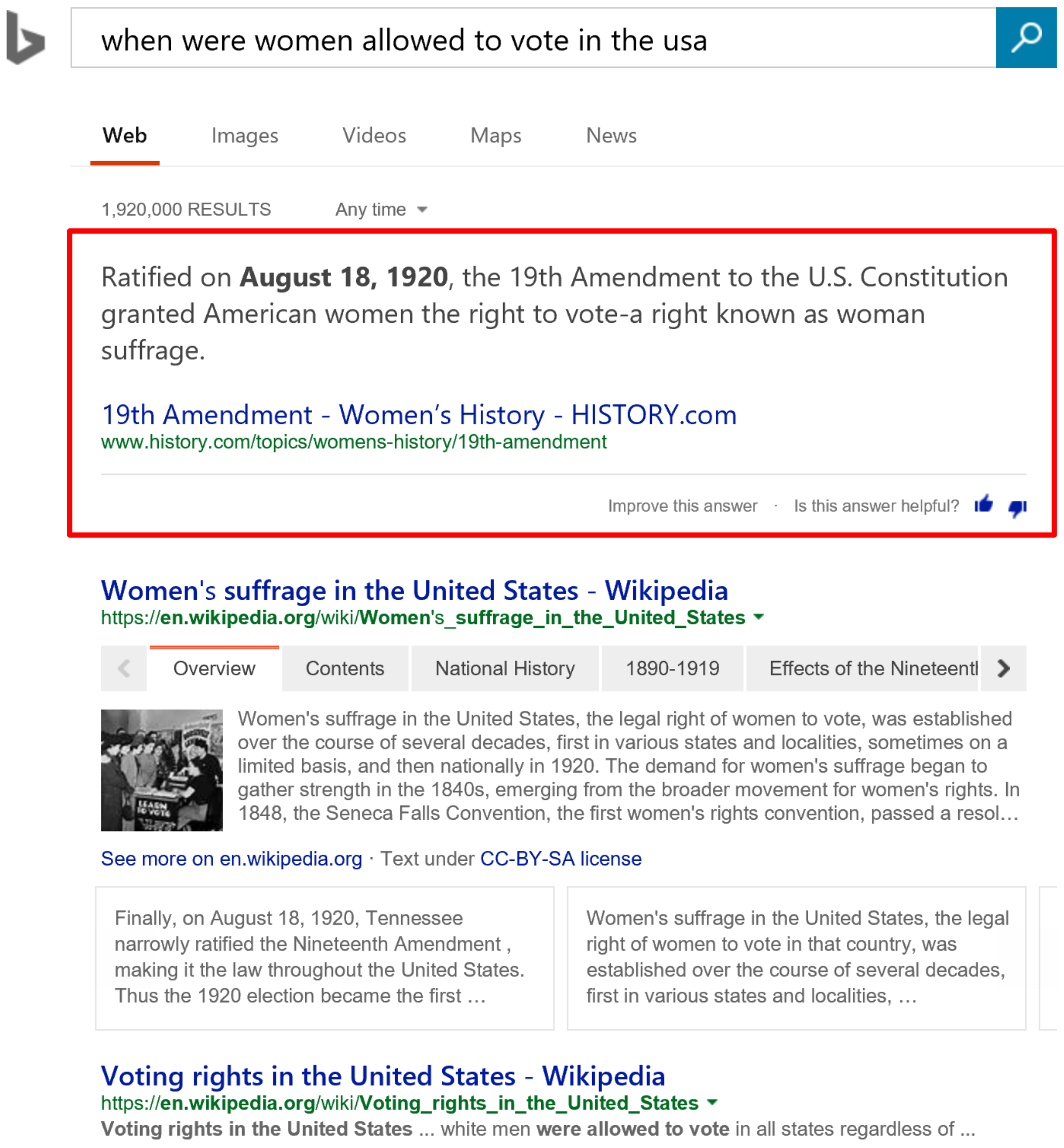
Best and

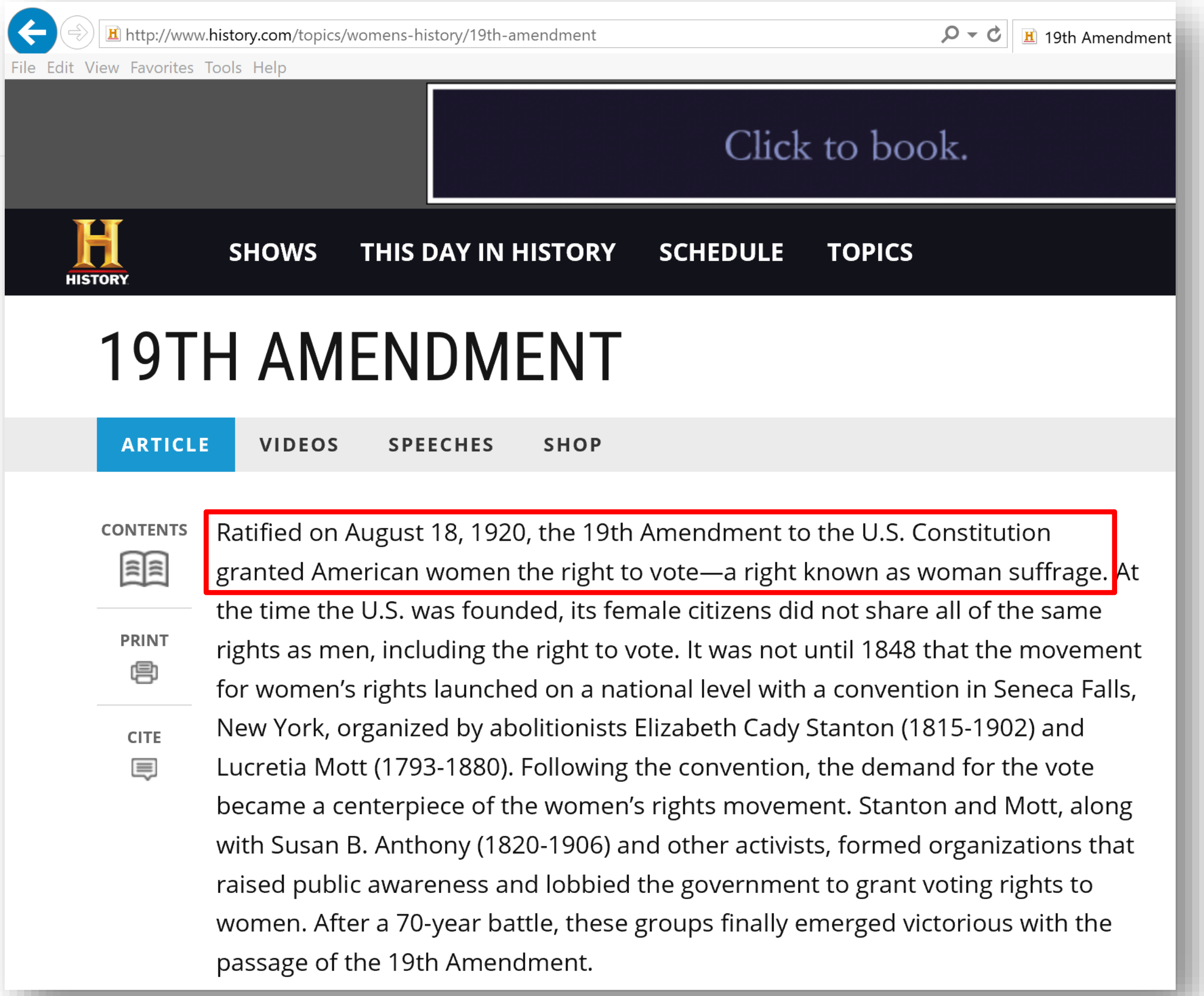
File Edit View Favorites Tools Help

STAR TREK GAMES FOR THE PC

	Game	Genre	Year	Metascore	Critic Grades	User Score
1	Star Trek Voyager: Elite Force	FPS	2000	86	25,0,0	9.3
2	Star Trek: Bridge Commander	Simulation	2002	82	20,4,0	8.1
3	Star Trek Deep Space Nine: The Fallen	3D Shooter	2000	81	18,5,0	8.5
4	Star Trek: Starfleet Command: Orion Pirates	Strategy	2001	78	7,2,0	7.5
	Star Trek: Starfleet Command III	RTS/Simulation	2002	78	11,7,0	9.2
	Star Trek Elite Force II	FPS	2003	78	21,7,0	7.4
7	Star Trek: Starfleet Command Volume II--Empires at War	RTS/Simulation	2000	77	9,7,0	9.3
8	Star Trek: Klingon Academy	Action/Sim	2000	74	5,9,0	10.0
9	Star Trek: Armada II	RTS	2001	65	4,7,2	8.0
10	Star Trek: Away Team	RTS	2001	64	3,16,3	8.5
	Star Trek: Deep Space Nine: Dominion Wars	RTS	2001	64	1,4,0	6.2
12	Star Trek: ConQuest Online	Strategy	2000	63	2,6,2	n/a
	Star Trek Online	MMORPG	2010	63	4,6,1	6.4
14	Star Trek Voyager: Elite Force Expansion Pack	Action	2001	62	2,6,2	8.6
15	Star Trek: Legacy	Action/RTS	2006	56	1,20,4	4.0
16	Star Trek: New Worlds	RTS	2000	52	4,9,10	8.1
17	Star Trek: D-A-C	Action	2009	50	4,19,0	4.1

PassageQA (文本问答)





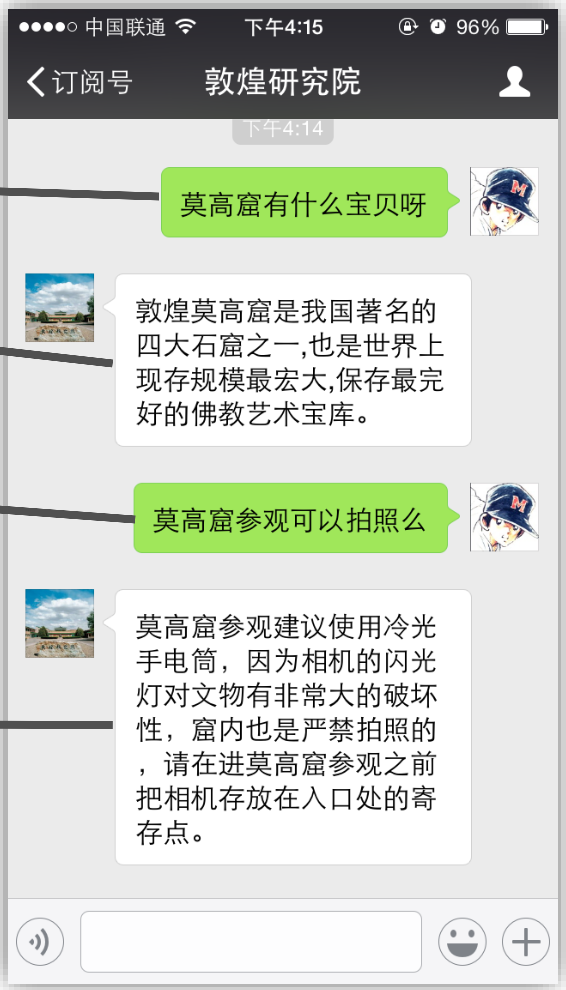
PassageQA (文本问答)

What is the Mogao Caves famous for?

The Mogao Caves are the best known of the Chinese Buddhist grottoes, and one of the four famous ancient Buddhist sculptural sites of China.

Can I take photos in the Mogao Caves?

Please use the cold-light torch within the Mogao caves, as the camera’s flashing is destructive to the wall paintings

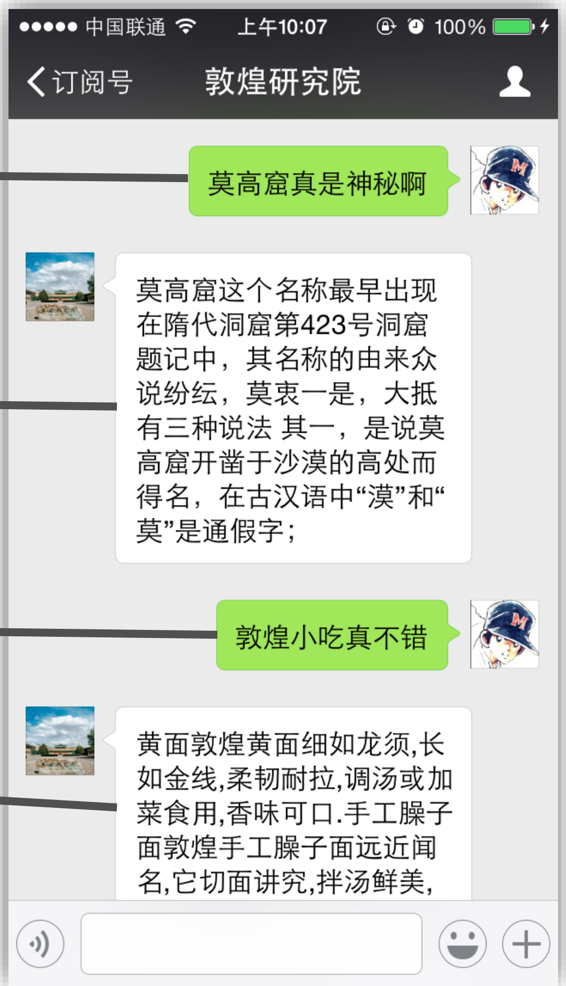


The history of Mogao Caves is so mysterious!

The construction of the Mogao Caves near Dunhuang is generally taken to have begun sometime in the fourth century CE, a Buddhist monk named Lè Zūn had a vision of a thousand Buddhas bathed in golden light at the site in 366 CE, inspiring him to build a cave here.

The snacks in Dunhuang is very delicious.

The meat of Donkey Meat Pasta is tender, sweet and delicious, and is also low in fat and high in protein.



CommunityQA (社区问答)

how to activate Office

日本のBingへ Nan Rewards

All Images Videos Maps News | My saves

Business results

176,000,000 Results Any time

1. Open an Office application, such as Word. (Can't find Office? Here's how to find Office in Windows 10, Windows 8, and Windows 7.)
2. View and accept the license agreement, if prompted.
3. Activation might take place automatically. If Office doesn't activate automatically, do one of the following: If the Activation Wizard appears, select ...
4. Under Step 1, select your country/region, and then call the Product Activation Center phone number that's listed under the country/region you selected. ...

Activate Office 365, Office 2016, or Office 2013 - Office ...
support.office.com/en-us/article/Activate-Office-365-Office-2016-or-Office-2013-1144e0de-e849-496e-8…

Is this answer helpful?

How To Activate Office 2013
www.intowindows.com › Microsoft **Office**
After completing the **Office** 2013 setup, just follow the given below steps to **activate** your **Office** 2013 copy. Step 1: Once installed, run **Office** Word, Excel or any ...

Where to enter your Office product key - Office Support
https://support.**office**.com/en-us/article/Where-to-enter-your...
Activate Office 365, **Office** 2016, or **Office** 2013; Deactivate an **Office** 365 install; Unlicensed Product error; ... Where to enter your **Office** product key.

Related searches

how to activate office **365**
how to activate **word 2016**
how to activate office **2016**
microsoft office **2016 activation**
microsoft office **professional plus 2013 product key**
how to activate office **online**
activate office **2016 without account**
activate office **professional plus 2016**

VisualQA (图片问答)

Result for Visual Question Answering

what is on the table

Submit

Predicted top-5 answers with confidence:

food	50.868%
donuts	6.663%
books	4.473%
cake	2.602%
laptop	2.523%

Agenda

- Deep Learning Basics for QA/QG
- QA based on Structured Data
- QA based on Unstructured Data
- QG and its Interaction with QA
- Summary, Latest Trends and Future Directions

Today's focus!

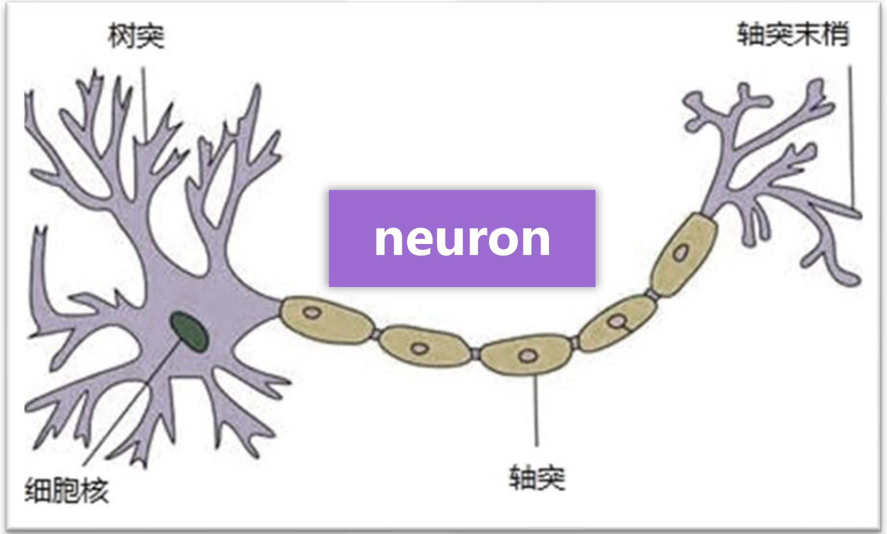
Deep Learning Basics for QA/QG

Neuron (神经元)

(McCulloch and Pitts, 1943)

accept input

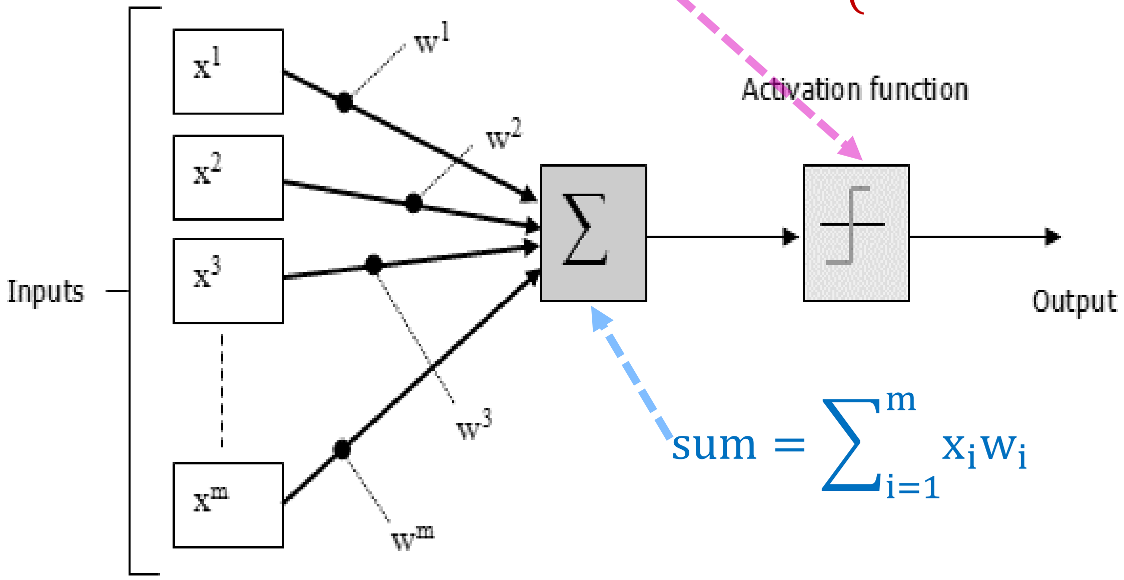
connected with other neurons



compute output

transfer output, if activated

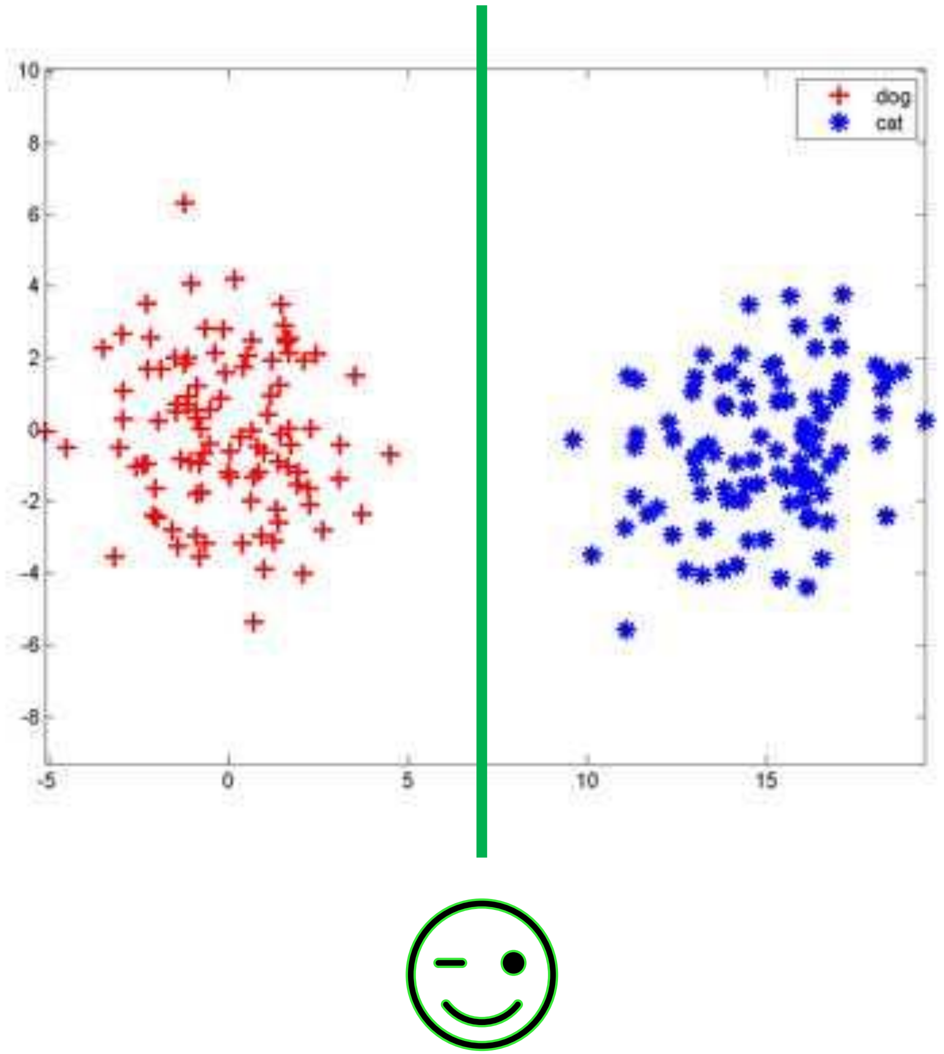
$$a(\text{sum}) = \begin{cases} +1 & \text{sum} \geq 0 \\ -1 & \text{sum} < 0 \end{cases}$$

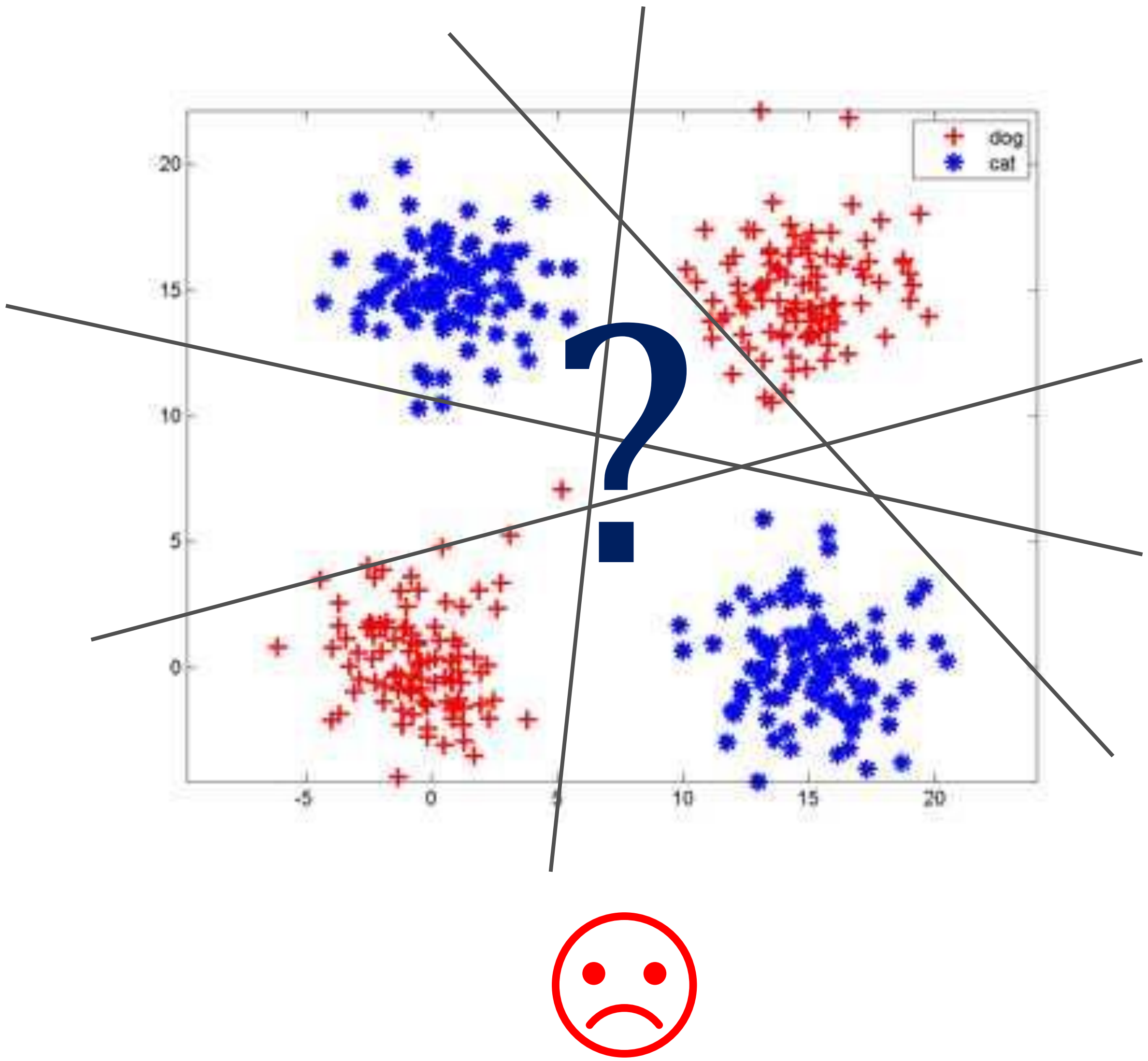


$$\text{sum} = \sum_{i=1}^{m} x_i w_i$$

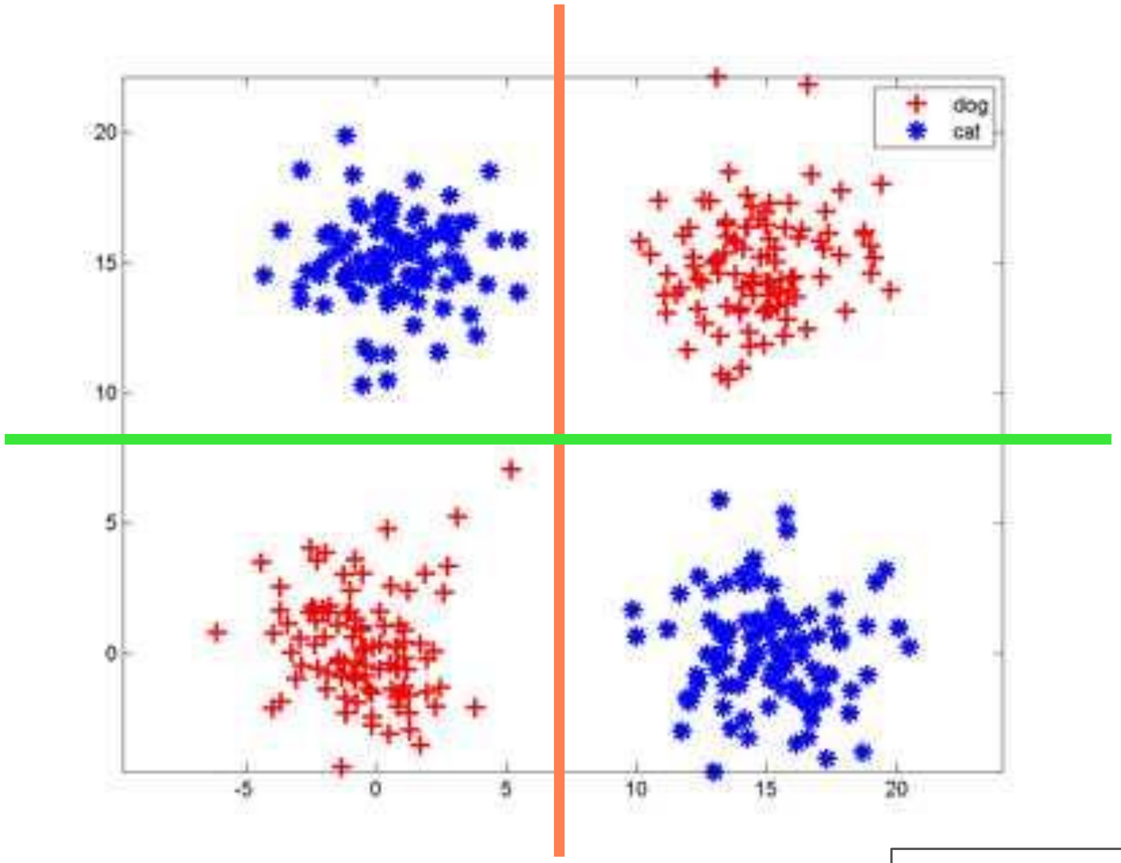
Perception (感知机)

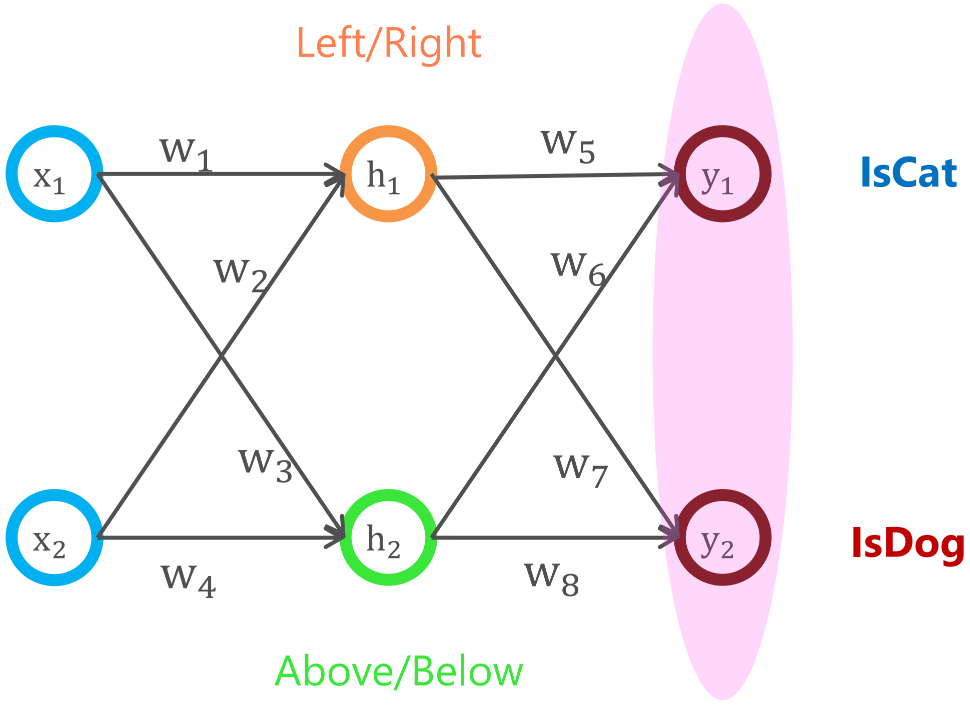
Separate Cats and Dogs with Neuron





Neural Network (神经网络)





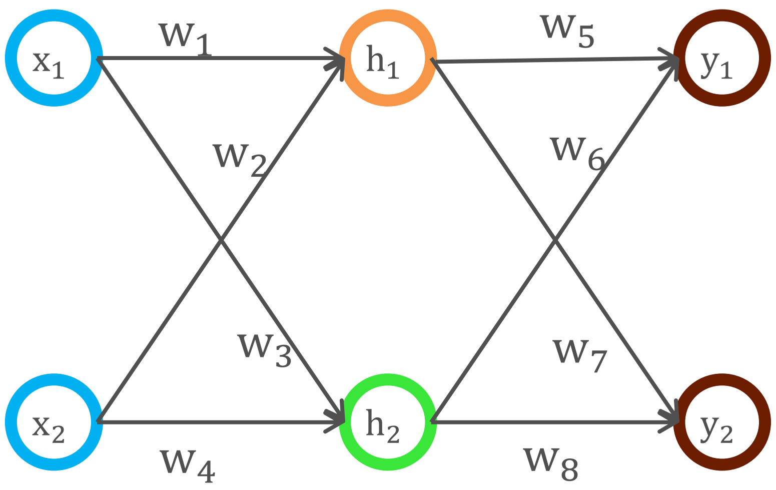
softmax function squashes a K-dimensional vector Z of arbitrary real values to a K-dimensional vector σ(Z) of real values in the range [0, 1] that add up to 1.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Feed Forward

$$\mathrm{net}_{h1} = w_1 * x_1 + w_2 * x_2$$

$$\mathrm{out}_{h1} = \frac{1}{1 + e^{-\mathrm{net}_{h1}}}$$



$$\mathrm{net}_{y1} = w_5 * \mathrm{out}_{h1} + w_6 * \mathrm{out}_{h2}$$

$$\mathrm{out}_{y1} = \frac{1}{1 + e^{-\mathrm{net}_{y1}}} \neq \mathrm{target}_{y1}$$

$$\mathrm{net}_{y2} = w_7 * \mathrm{out}_{h1} + w_8 * \mathrm{out}_{h2}$$

$$\mathrm{out}_{y2} = \frac{1}{1 + e^{-\mathrm{net}_{y2}}} \neq \mathrm{target}_{y2}$$

$$\mathrm{net}_{h2} = w_3 * x_1 + w_4 * x_2$$

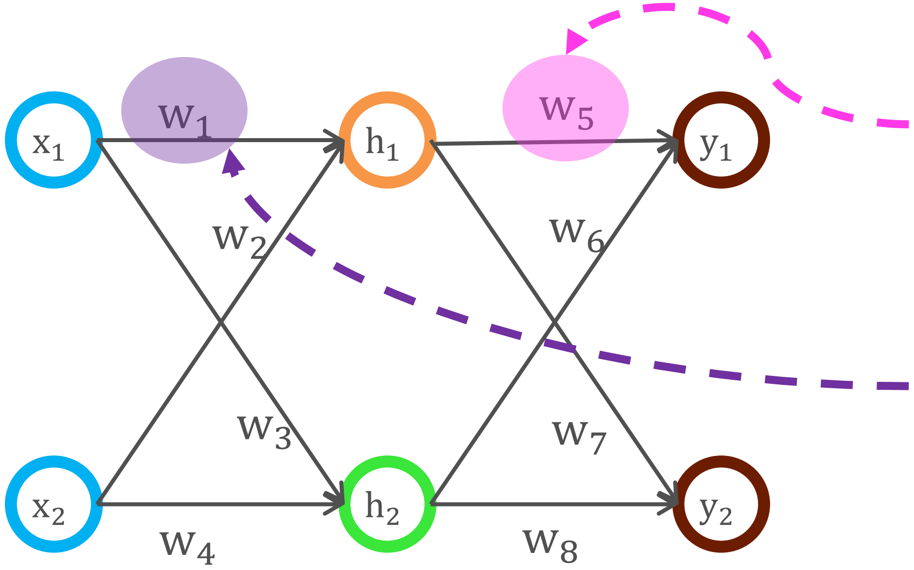
$$\mathrm{out}_{h2} = \frac{1}{1 + e^{-\mathrm{net}_{h2}}}$$

We want to adjust weights $\{w_j\}$ to ensure that each out_{y_i} is as close to target_{y_i} as possible.

Backward Propagation

$$w_i^+ = w_i - \alpha \frac{\partial E}{\partial w_i}$$

$$E = \frac{1}{2}(\text{target}_{y1} - \text{out}_{y1})^2 + \frac{1}{2}(\text{target}_{y2} - \text{out}_{y2})^2$$



$$\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial \text{out}_{y1}} * \frac{\partial \text{out}_{y1}}{\partial \text{net}_{y1}} * \frac{\partial \text{net}_{y1}}{\partial w_5}$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial \text{out}_{h1}} * \frac{\partial \text{out}_{h1}}{\partial \text{net}_{h1}} * \frac{\partial \text{net}_{h1}}{\partial w_1}$$

Chain Rule

Three Neural Network-based Technologies for QA and QG

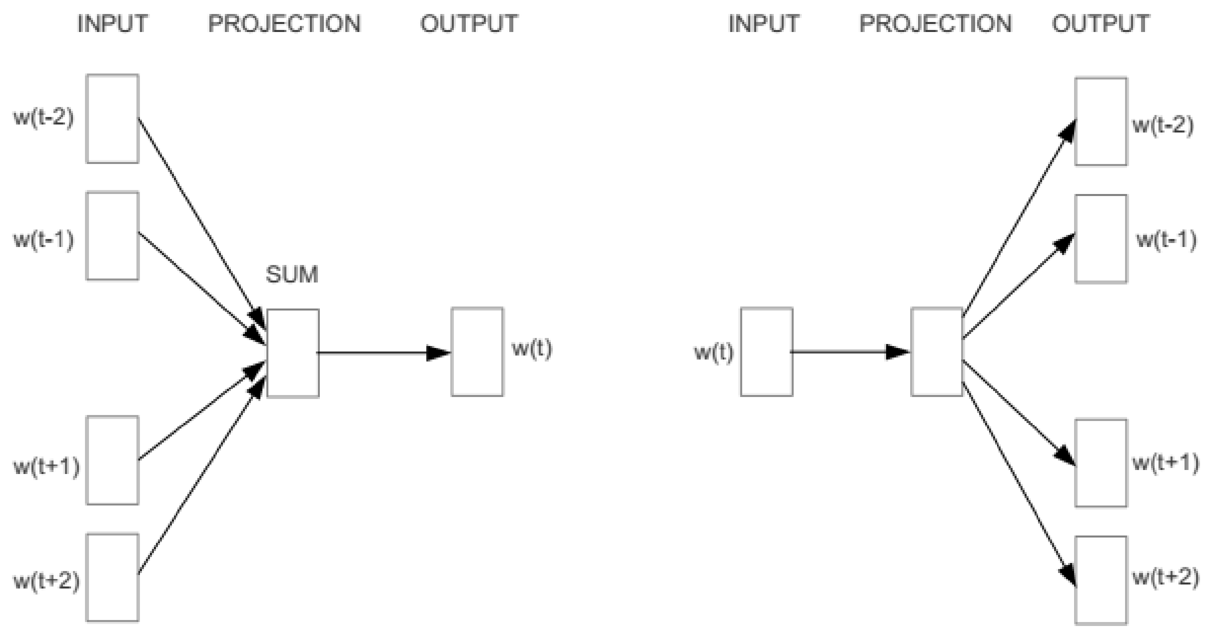
- Word Embedding
- Sequence Embedding
- Sequence Generation

Three Neural Network-based Technologies for QA and QG

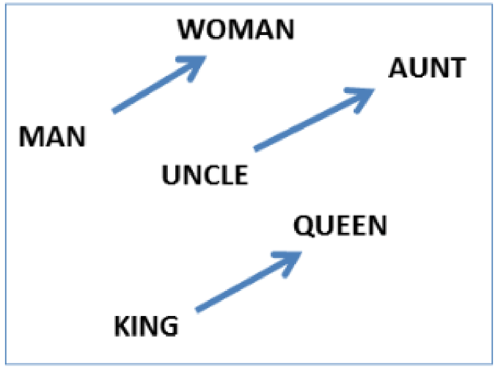
- Word Embedding
- Sequence Embedding
- Sequence Generation

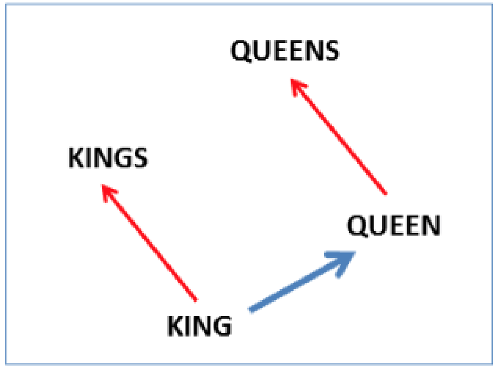
Word Embedding (词向量)

(Mikolov et al., 2013)



CBOW **Skip-gram**





$$J_\theta^{\mathrm{CBOW}} = -\frac{1}{T}\sum_{t=1}^{T}\log\left(p(w_t|w_{t-n},\dots,w_{t-1},w_{t+1},\dots,w_{t+n})\right)$$

$$p(w_t|w_{t-n},\dots,w_{t-1},w_{t+1},\dots,w_{t+n}) = \frac{\exp(h^T v'_t)}{\sum_{w_i\in V}\exp(h^T v'_{w_i})}$$

$$J_\theta^{\mathrm{Skip-gram}} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0}\log p(w_{t+j}|w_t)$$

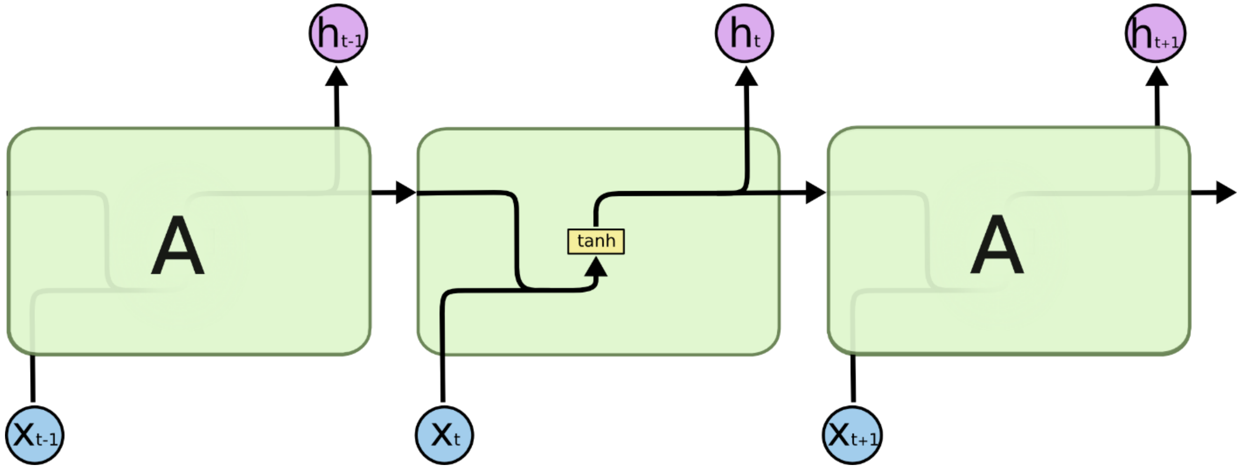
$$p(w_{t+j}|w_t) = \frac{\exp(v_{w_t}^T v'_{w_{t+j}})}{\sum_{w_i\in V}\exp(v_{w_t}^T v'_{w_i})}$$

Three Neural Network-based Technologies for QA and QG

- Word Embedding
- Sequence Embedding
- Sequence Generation

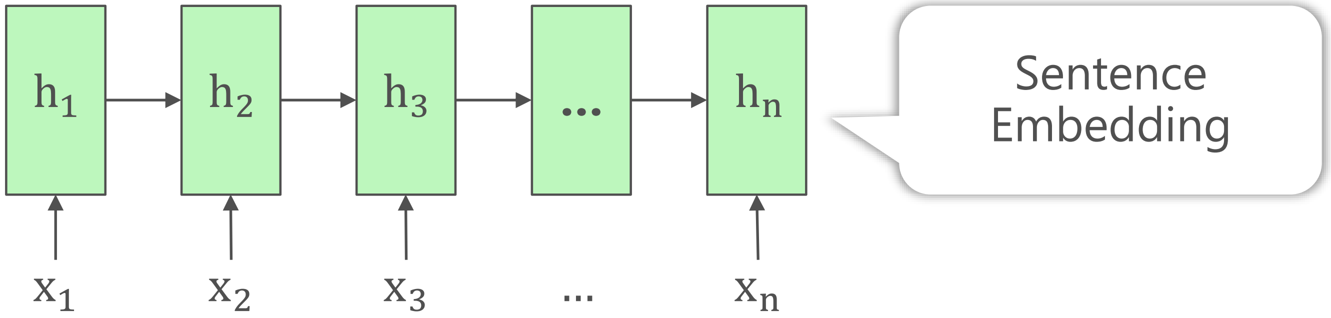
Recurrent Neural Network (RNN) (循环神经网络)

(Mikolov et al., 2010)

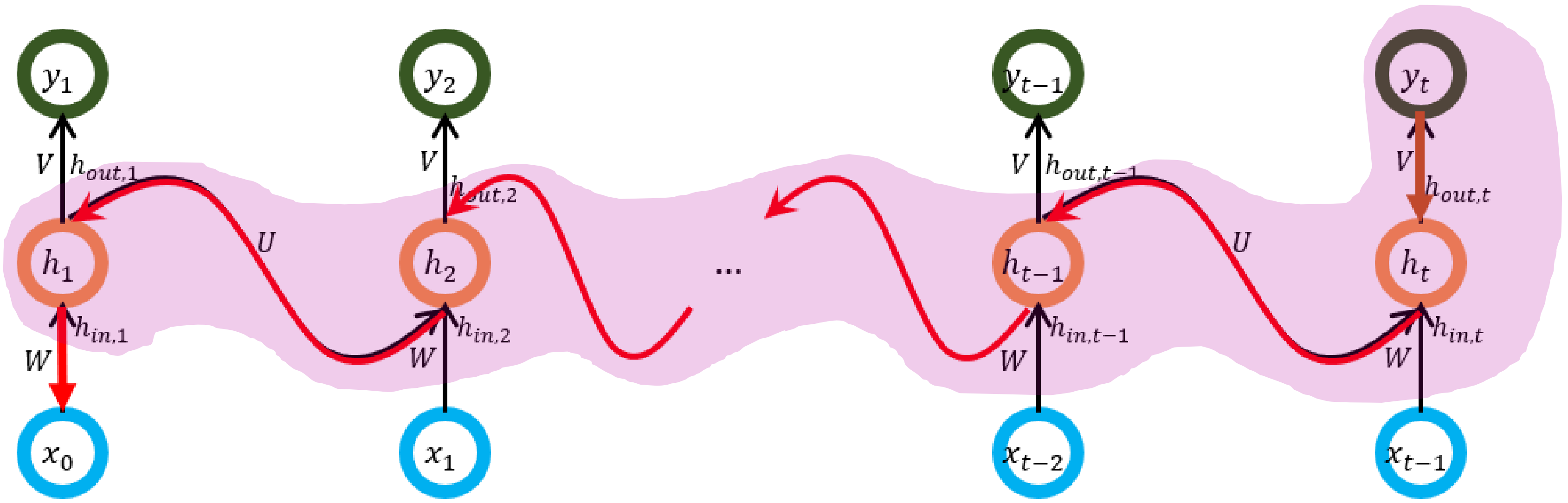


$$h_t = \tanh(W_h \cdot h_{t-1} + W_x \cdot x_t + b) = \tanh(W \cdot [h_{t-1}; x_t] + b)$$

- Goal
  - Represent an ordered sequence of words
- Input
  - A sequence of word embeddings
- Output
  - A sequence of hidden states

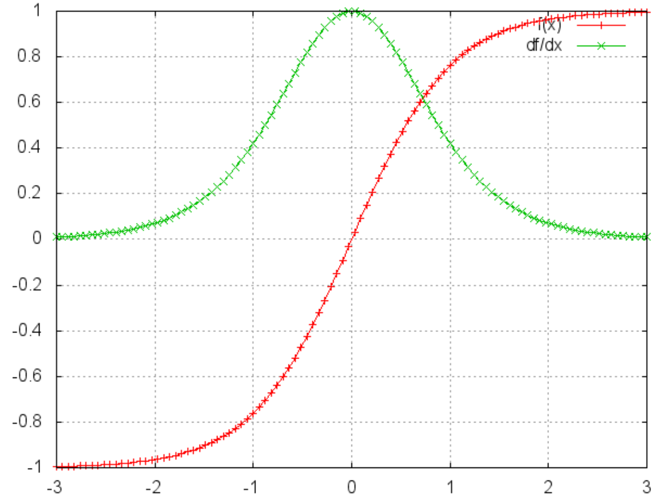


Vanishing Gradients



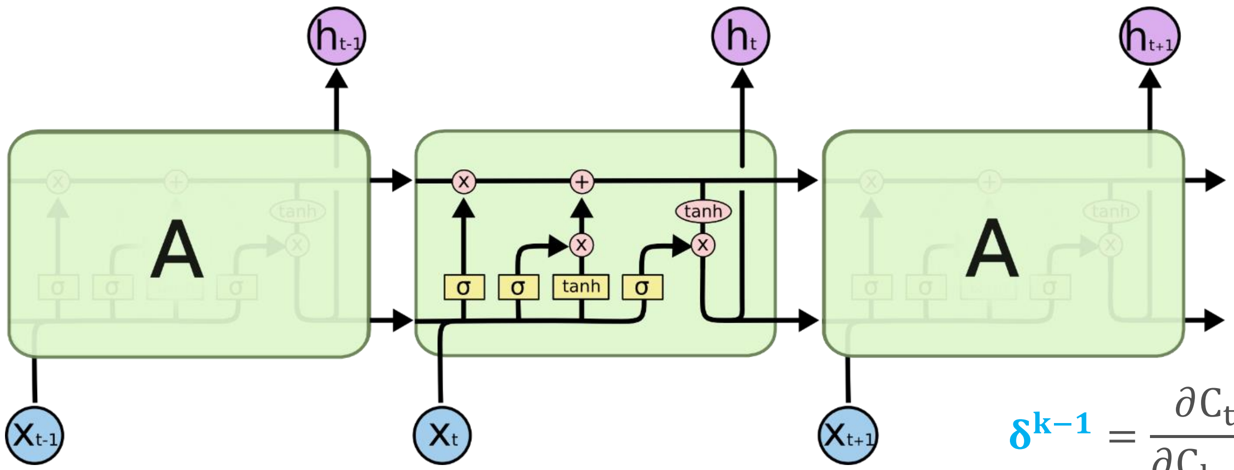
$$\delta_{\mathrm{in},1} = \delta_{\mathrm{out},t} * \frac{\partial h_{\mathrm{out},t}}{\partial h_{\mathrm{in},t}} * \frac{\partial h_{\mathrm{in},t}}{\partial h_{\mathrm{out},t-1}} * \cdots * \frac{\partial h_{\mathrm{in},2}}{\partial h_{\mathrm{out},1}} * \frac{\partial h_{\mathrm{out},1}}{\partial h_{\mathrm{in},1}}$$

Value range of
$\tanh(x)$ and $\tanh'(x)$



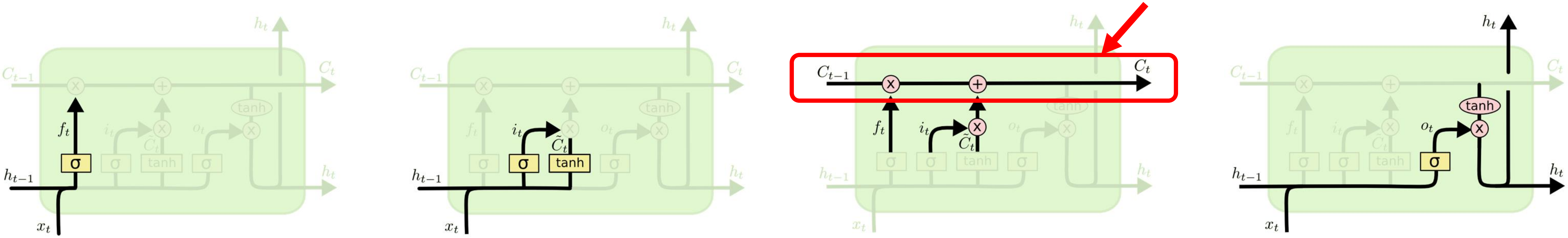
Long Short-Term Memory (LSTM) (长短期记忆)

(Hochreiter and Schmidhuber, 1997)



$$\delta^{k-1} = \frac{\partial C_t}{\partial C_{k-1}} = \frac{\partial C_t}{\partial C_k} * \frac{\partial C_k}{\partial C_{k-1}} = \delta^k * \frac{\partial C_k}{\partial C_{k-1}}$$

$$= \delta^k * (f_k + \cdots)$$



$\mathbf{f_t} = \sigma(W_f \cdot [h_{t-1}; x_t] + b_f)$

$\mathbf{i_t} = \sigma(W_i \cdot [h_{t-1}; x_t] + b_i)$

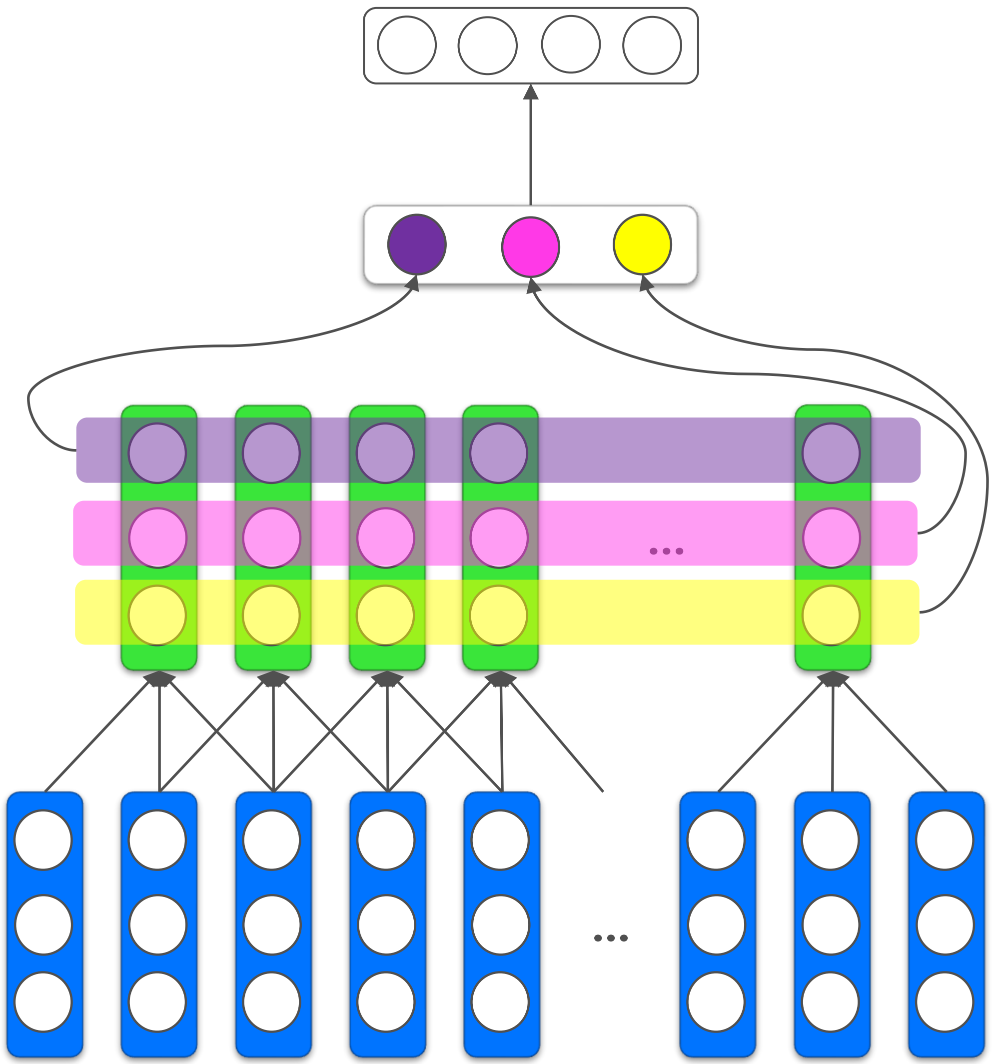
$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}; x_t] + b_C)$

$C_t = \mathbf{f_t} \cdot C_{t-1} + \mathbf{i_t} \cdot \tilde{C}_t$

$\mathbf{o_t} = \sigma(W_o \cdot [h_{t-1}; x_t] + b_o)$

$h_t = \mathbf{o_t} \cdot \tanh(C_t)$

Convolution Neural Network (CNN) (卷积神经网络)



Output Layer

$$y_O = \tanh(W_O \cdot \hat{c} + b_O)$$

Pooling Layer

$$\hat{c}^j = \max\{c_1^j, \dots, c_N^j\}$$

Convolution Layer

$$c_i = f(W_C \cdot x_{i:i+d-1} + b_C)$$

Local Receptive Field

Weight Sharing

Input Layer

An Example: CNN for Image Processing

Image

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

×

Conv-kernel

	0	1	0	
	0	0	0	
	0	0	0	

=

Processed Image

		42		

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

Sharpen

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Blur

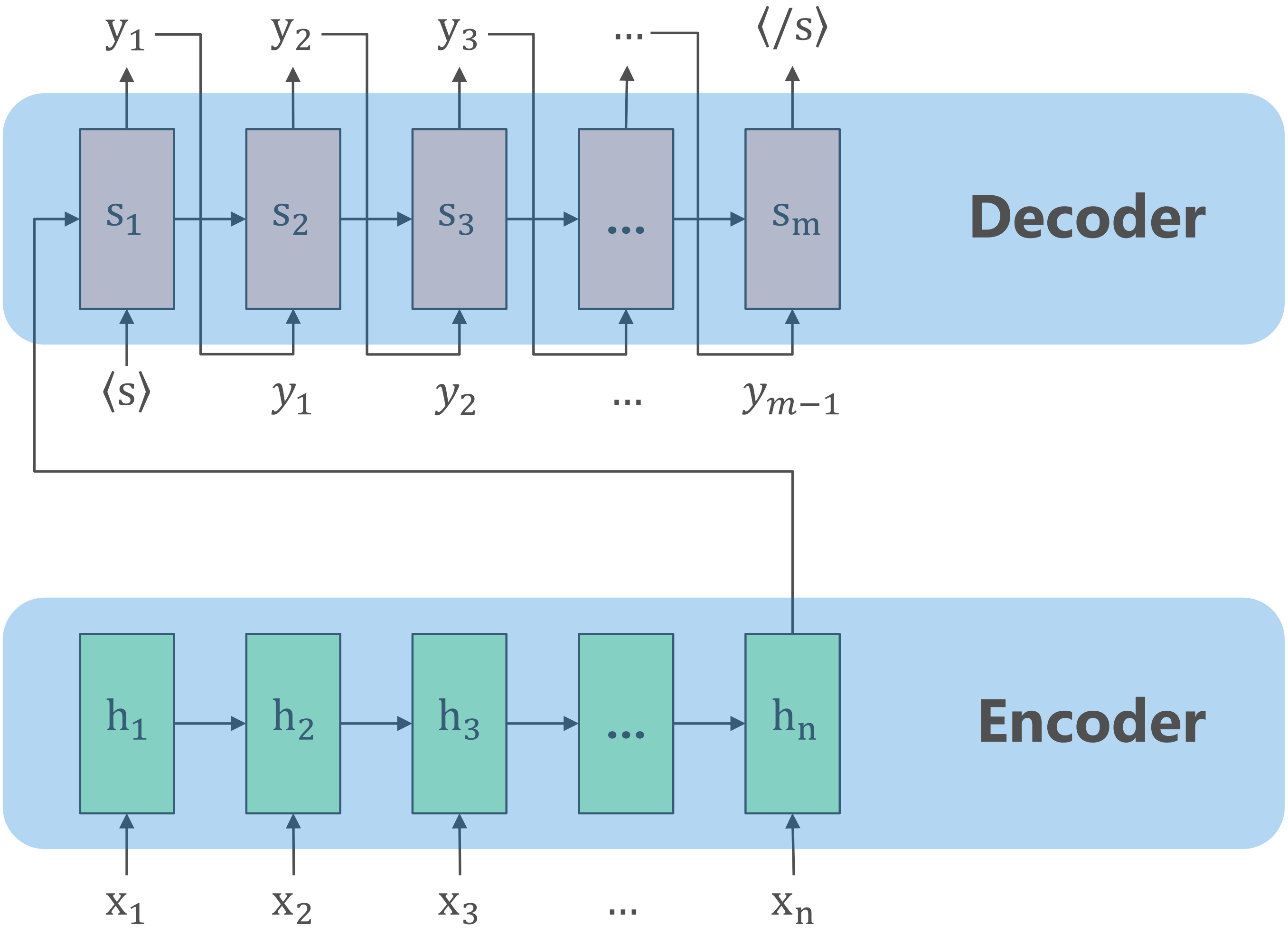
	0	0	0	
	-1	1	0	
	0	0	0	

Edge Enhance

Three Neural Network-based Technologies for QA and QG

- Word Embedding
- Sequence Embedding
- Sequence Generation

Encoder-Decoder Framework for Sequence Generation

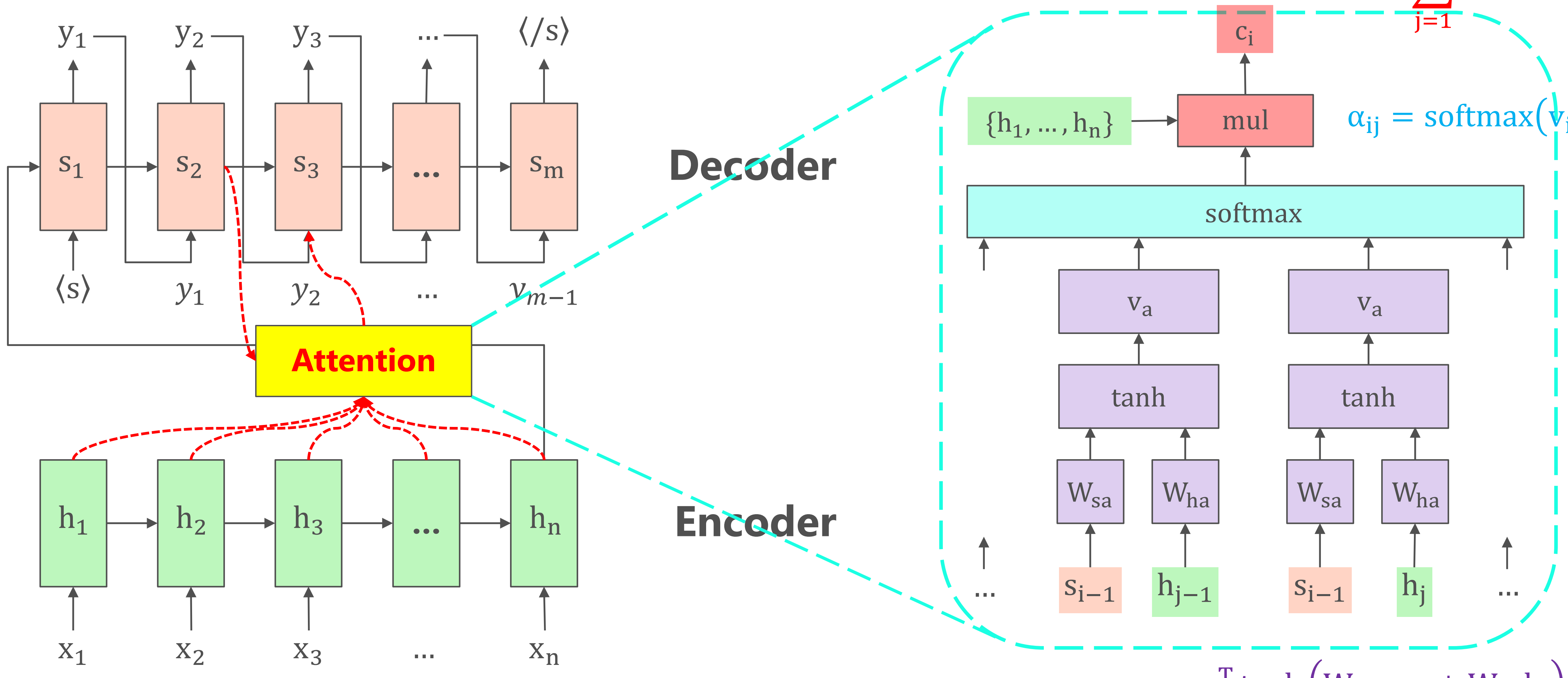


$$s_i = \tanh(W \cdot y_{i-1} + U \cdot s_{i-1})$$

Encoder-Decoder Framework with Attention

$$s_i = \tanh(W \cdot y_{i-1} + U \cdot s_{i-1} + \mathbf{C} \cdot \mathbf{c_i})$$

$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j$$



$$\alpha_{ij} = \mathrm{softmax}(v_{ij})$$

$$v_{ij} = v_a^T \tanh\left(W_{sa} s_{i-1} + W_{ha} h_j\right)$$

An Example: Attention in Caption Generation

a little girl sitting on a bench holding an umbrella.

a herd of sheep grazing on a lush green hillside.

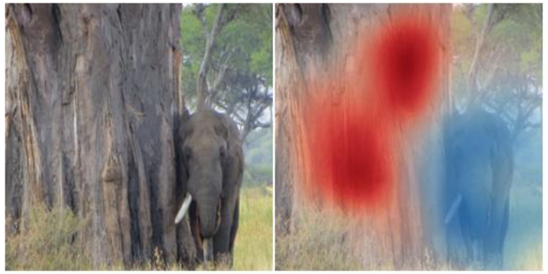
a close up of a fire hydrant on a sidewalk.

a yellow plate topped with meat and broccoli.

a zebra standing next to a zebra in a dirt field.

a stainless steel oven in a kitchen with wood cabinets.

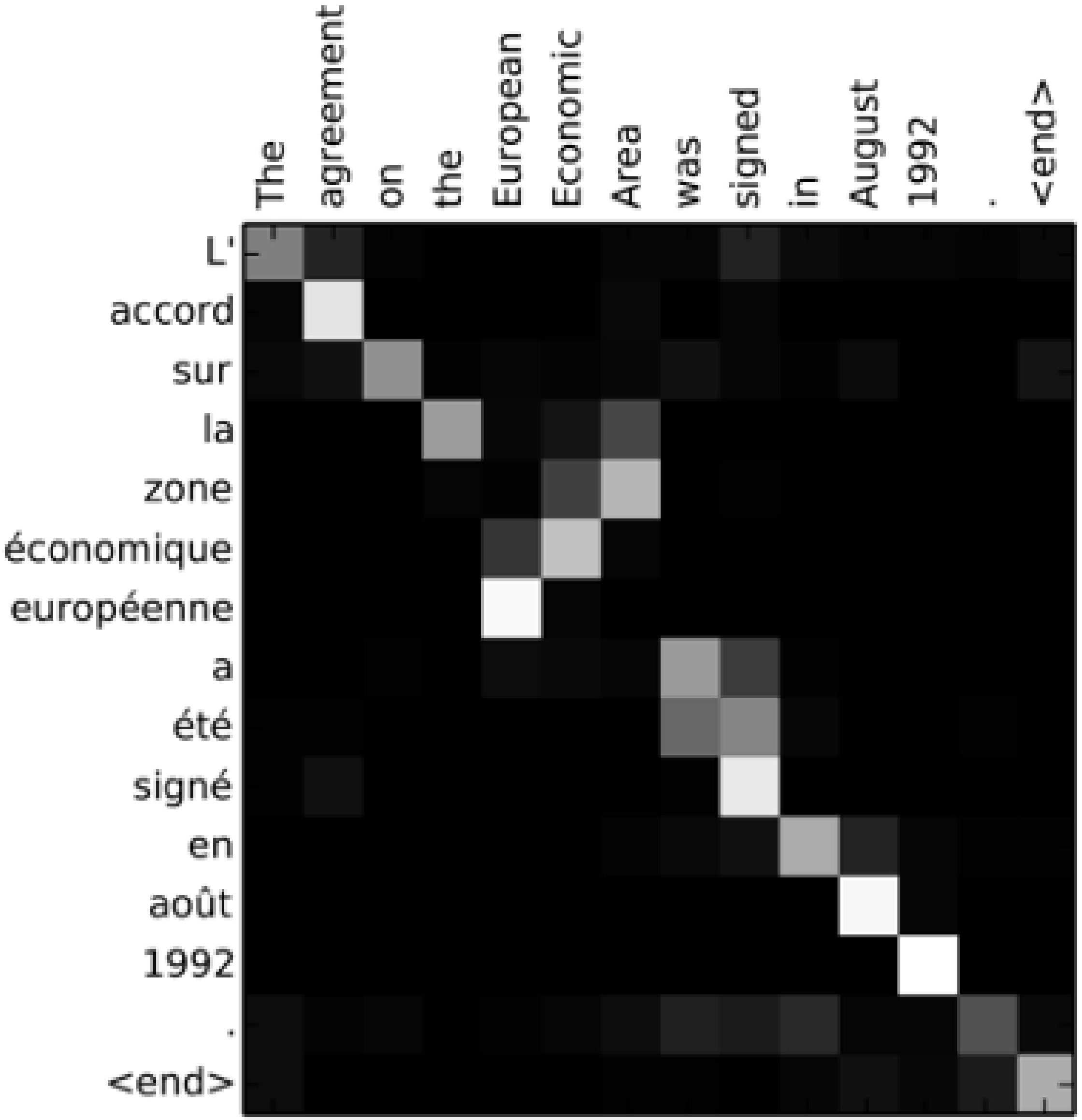
two birds sitting on top of a tree branch.



an elephant standing next to rock wall.

a man riding a bike down a road next to a body of water.

An Example: Attention in Machine Translation



It’s enough~

- Seems any task can be done with neural network, by

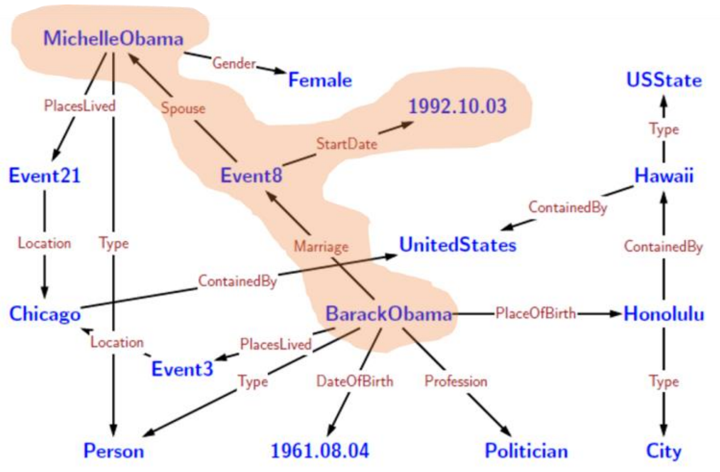
  1. Defining the task
  2. Designing the network structure
  3. Defining a loss function
  4. Labeling training data
  5. Tuning model parameters
  6. Applying the well-trained model to the task

- Let’s talk about QA/QG tasks now!

QA based on Structured Data

QA based on Structured Data

who is Michelle Obama married to in 1992 ?



Knowledge-based QA (KBQA)

Michelle Obama

which city hosted Summer Olympic in 2008 ?

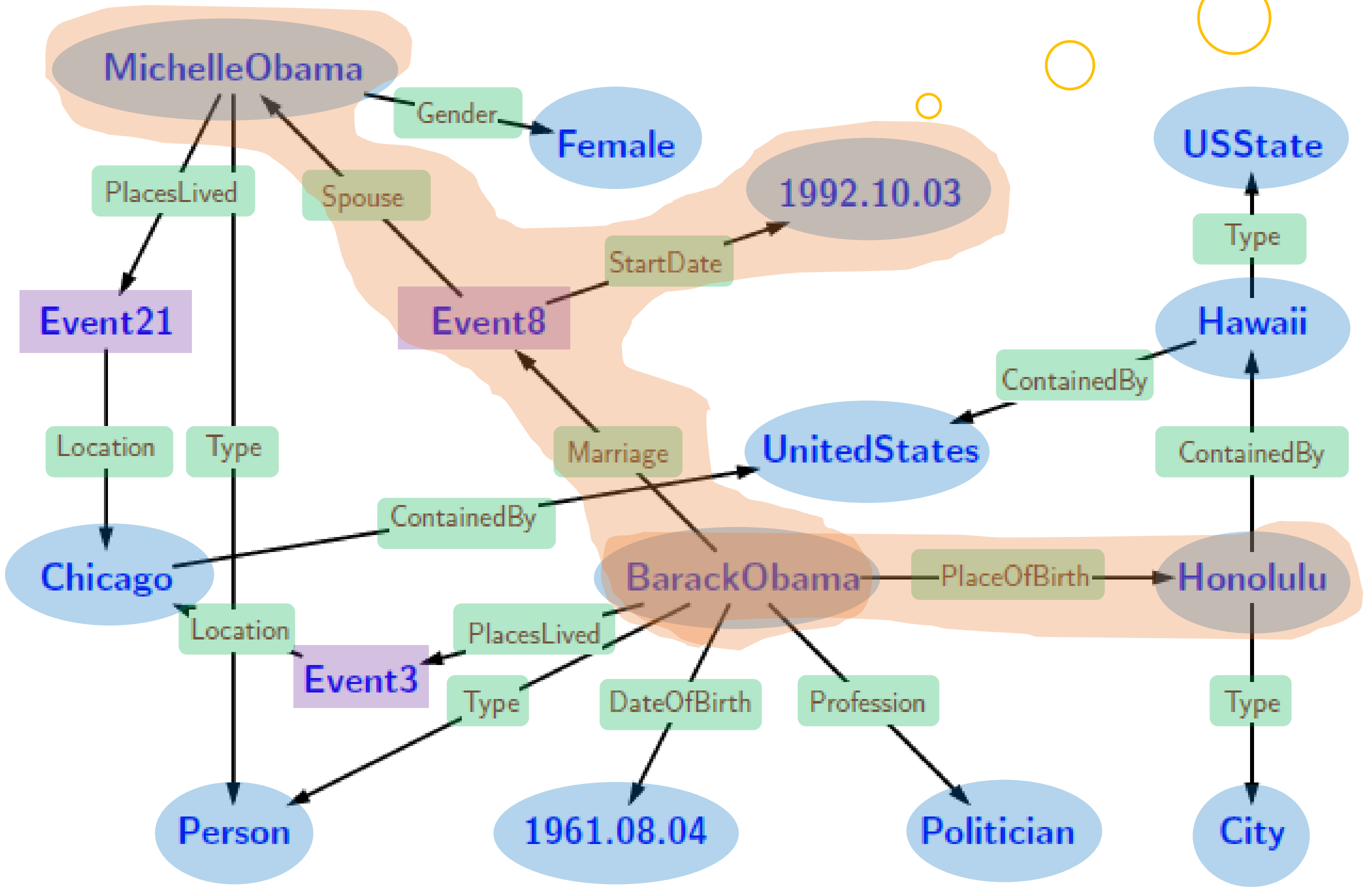
Year	City	Country	Nations
1896	Athens	Greece	14
1900	Pairs	France	24
…	…	…	…
2004	Athens	Greece	201
2008	Beijing	China	204

Table-based QA (TBQA)

Beijing

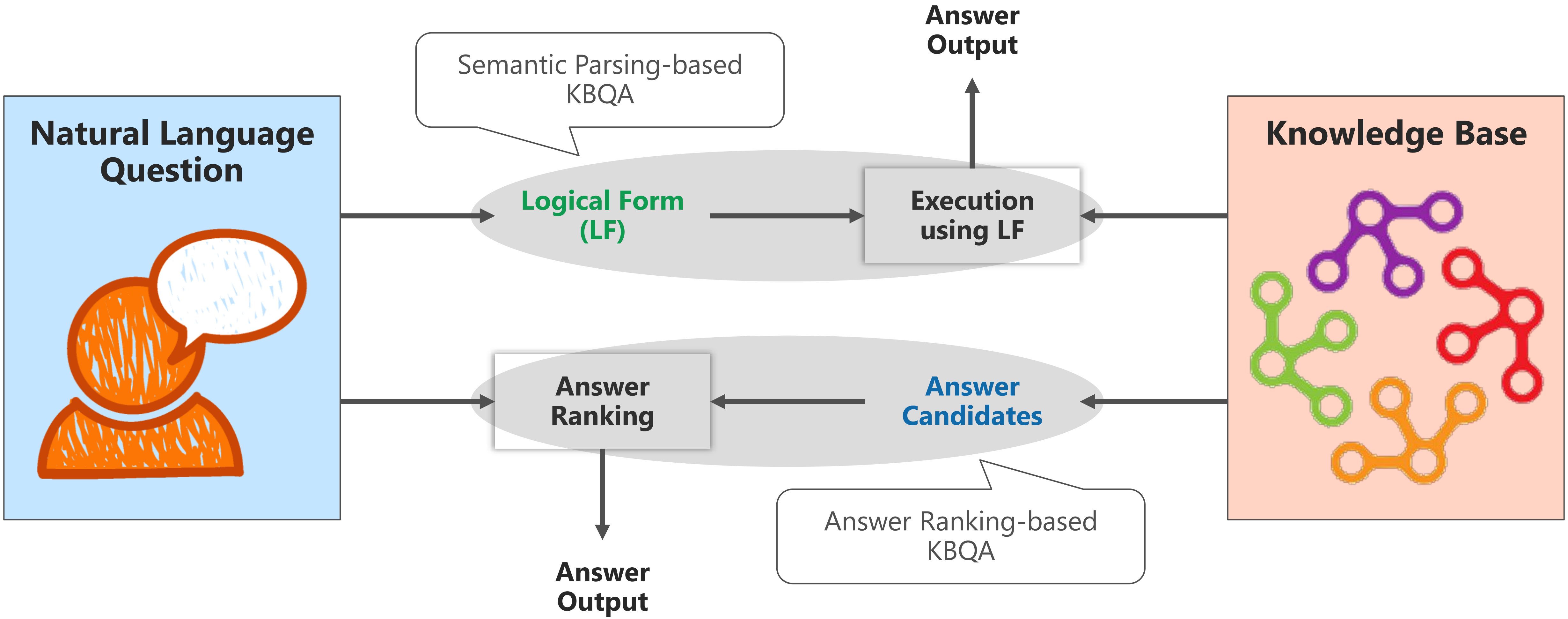
Knowledge Base (KB)

- Curated database with well-defined schema



- **Entity**
  Objects/Values in the world
- **Predicate**
  Relation between two connected entities
- **CVT** (Compound Value Type)
  Not a real-world entity, but is used to collect multiple fields of an event
- **Fact**
  Triple, which connects two entities
  Event, which connects multiple entities via a CVT node

Knowledge-based QA (KBQA): Methodology



Outline of KBQA

- Semantic Parsing-based KBQA
  1. What is **logical form**?
  2. How to **parse** a question into its logical form?
  3. How to **execute** a logical form against KB?
- Answer Ranking-based KBQA
  1. How to **select** answer candidates?
  2. How to **represent** answer candidates?
  3. How to **rank** answer candidates?

Outline of KBQA

- Semantic Parsing-based KBQA

  1. What is **logical form**?
  2. How to **parse** a question into its logical form?
  3. How to **execute** a logical form against KB?

- Answer Ranking-based KBQA

  1. How to **select** answer candidates?
  2. How to **represent** answer candidates?
  3. How to **rank** answer candidates?

Lambda Calculus (λ-Calculus) as Logical Form (LF)

- λ-Calculus was introduced by Alonzo Church in 1930s
- Any computable function can be expressed using this formalism
- The core concept in λ-Calculus is "expression"
- An **expression** is defined recursively as follows

$$\langle \text{expression} \rangle \coloneqq \langle \text{constant} \rangle \mid \langle \text{variable} \rangle \mid \langle \text{function} \rangle \mid \langle \text{application} \rangle$$

$$\langle \text{function} \rangle \coloneqq \lambda \, \langle \text{variable} \rangle . \langle \text{expression} \rangle$$

$$\langle \text{application} \rangle \coloneqq \langle \text{expression} \rangle \, \langle \text{expression} \rangle$$

Lambda Calculus: Constant

- Represent objects in the world

China, Bill Gates, Mount Everest, 2017, ...

Lambda Calculus: **Variable**

- Represent object variables

$$x, y, z, \ldots$$

Lambda Calculus: **Function**

- Represent a function, and return the output of the function

$$\boxed{\lambda x}\,.\,\boxed{\text{Place\_Of\_Birth}(\text{Barack Obama}, x)}$$

the **argument**
of the function

the **definition**
of the function

Lambda Calculus: Application

- Apply the first expression to the second expression

$$\lambda x \lambda y.\, \mathrm{Place\_Of\_Birth}(x, y) \quad \lambda x.\, (x = \mathrm{Barack\ Obama})$$

$$\Downarrow$$

$$\lambda y.\, \mathrm{Place\_Of\_Birth}(\mathrm{Barack\ Obama}, y)$$

Transforming Natural Language into Logical Form (λ-Calculus)

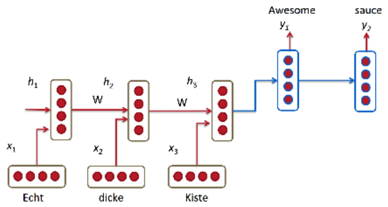
Natural Language

Semantic Parsing

Logical Form

What city was Obama born ?

Grammar-based Semantic Parsing



Neural Network-based Semantic Parsing

$$\lambda x.\, \mathrm{Type}(\mathrm{City}, x) \wedge \mathrm{Place\_of\_Birth}(\mathrm{Barack\ Obama}, x)$$

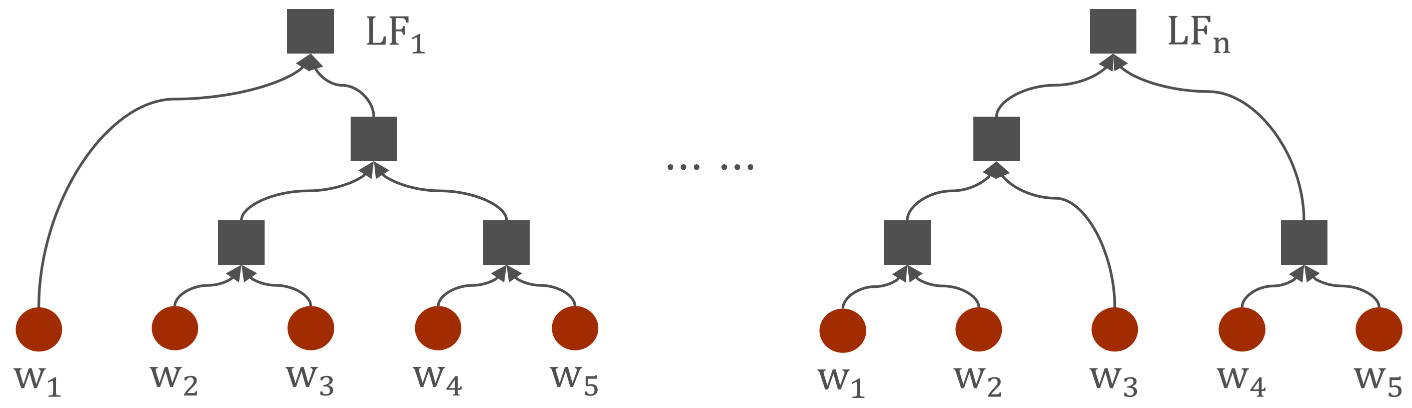
Generic Framework of Grammar-based Semantic Parsing

$$\widetilde{\mathrm{LF}} = \underset{\mathrm{LF}\in\{\mathrm{LF}_1,\ldots,\mathrm{LF}_n\}}{\operatorname{argmax}} \mathrm{P}(\mathrm{LF}|\mathrm{Q};\bar{\theta}) = \underset{\mathrm{LF}\in\{\mathrm{LF}_1,\ldots,\mathrm{LF}_n\}}{\operatorname{argmax}} \sum_{\mathrm{D}} \mathrm{P}(\mathrm{LF},\mathrm{D}|\mathrm{Q};\bar{\theta})$$

Logical Form Ranking

Rank different logical forms

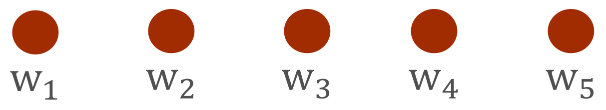
- Q: a question
- LF: a logical form of Q
- D: a semantic parsing derivation
- $\bar{\theta}$: model parameters



Semantic Rule-Driven CYK Parsing

Parse Q into possible logical forms $\{\mathbf{LF}_1, \ldots, \mathbf{LF}_n\}$ using semantic rules

- CCG
- SCFG
- λ-DCS
- …



Natural Language Question

Combinatorial Categorial Grammar (CCG)

- CCG captures syntactic and semantic information jointly

A CCG Rule Example

$$\underbrace{\mathrm{border}}_{\text{natural language}} \coloneqq \underbrace{(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}}_{\text{syntax}} : \underbrace{\lambda x \lambda y.\mathrm{Border}(x,y)}_{\text{semantics}}$$

- Match natural language input

- Syntactic symbols: S, N, NP, ADJ and PP
- Syntactic combinator: / and \
- Slashes specify combination orders and directions

- λ-Calculus expression
- Sematic types are the logical forms of the natural language parts

Semantic Parsing with CCG

(Zettlemoyer and Collins, 2007; Kwiatkowski et al., 2011)

State | borders | New Mexico

NP
$\lambda x.\mathrm{Type}(State, x)$

(S\NP)/NP
$\lambda x \lambda y.\mathrm{Border}(x, y)$

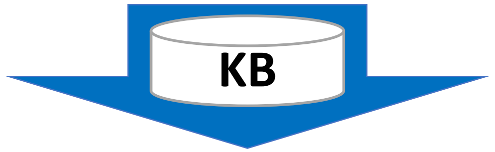
NP
New Mexico

>

S\NP
$\lambda x.\mathrm{Border}(x, \text{New Mexico})$

<

S
$\lambda x.\mathrm{Type}(\text{State}, x) \wedge \mathrm{Border}(x, \text{New Mexico})$



Arizona, Colorado, Oklahoma, Texas

CCG Rule Mining

- Input (<question, logical form> pairs)

Texas borders New Mexico
borders(texas, new_mexico)

use rules to extract all possible <Q, LF> pairs

- Output (CCG rules)

Texas := NP : texas
borders := (S \ NP) / NP : λx.λy.borders(y, x)
New Mexico := NP : new_mexico

Category Rules

Input Trigger	Output Category
constant c	NP : c
arity one predicate p	N : λx.p(x)
arity one predicate p	S \ NP : λx.p(x)
arity two predicate p	(S \ NP) / NP : λx.λy.p(y, x)
arity two predicate p	(S \ NP) / NP : λx.λy.p(x, y)
arity one predicate p	N / N : λg.λx.p(x) ∧ g(x)
arity two predicate p and constant c	N / N : λg.λx.p(x, c) ∧ g(x)
arity two predicate p	(N \ N) / NP : λx.λg.λy.p(y, x) ∧ g(x)
arity one function f	NP / N : λg.argmax/min(g(x), λx.f(x))
arity one function f	S / NP : λx.f(x)

1. **maximize the likelihood:** $\prod_i P_w(LF_i|Q_i) = \prod_i \sum_d P_w(LF_i, d|Q_i)$

2. **keep CCG rules that occur in the highest scoring derivations of training data**

Synchronous Context Free Grammar (SCFG)

- SCFG captures lexical and semantic information jointly
- A SCFG consist of a set of production rules

Category

$$X \rightarrow \langle \alpha \quad \beta \rangle$$

a sequence of terminal and non-terminal symbols

the logical form of α

Non-terminal symbols in α and β should have 1-to-1 correspondence.

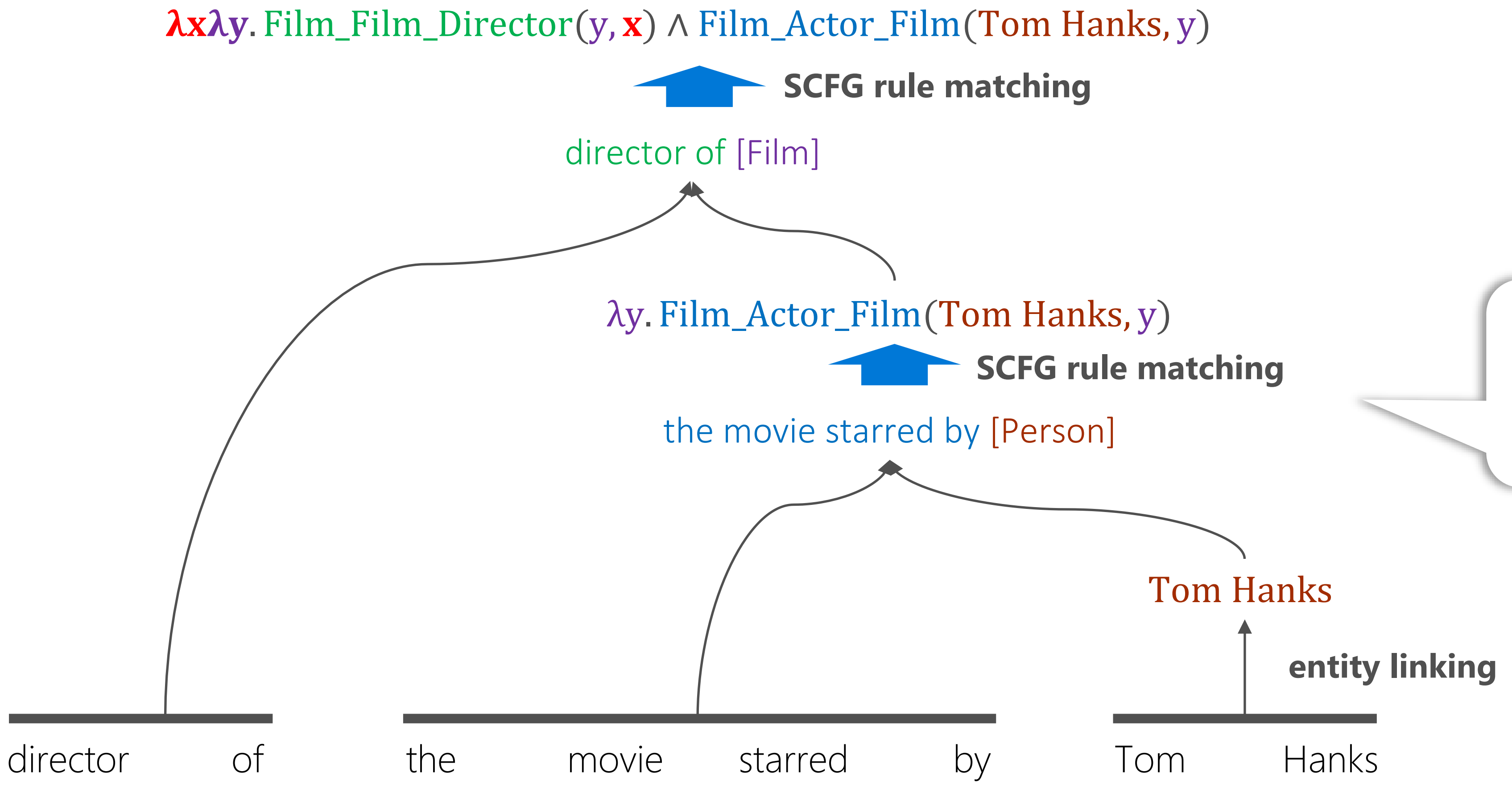
- Examples

$[\text{Person}] \rightarrow \langle \text{Tom Hanks} \quad \text{Tom Hanks} \rangle$

$[\text{Film}] \rightarrow \langle \text{the moive starred by } [\text{Person}]_1 \quad \lambda x.\,\text{Film\_Actor\_Film}([\text{Person}]_1, x) \rangle$

$[\text{Film}] \rightarrow \langle \text{the moive starred by Tom Hanks} \quad \lambda x.\,\text{Film\_Actor\_Film}(\text{Tom Hanks}, x) \rangle$

Semantic Parsing with SCFG
(Bao et al., 2014; Wong and Mooney, 2007)



Lots of semantic derivations will be generated during this procedure.

SCFG Rule Mining

Paired Entities of a given KB Predicate

Film.Film.Director

<Forrest Gump, Robert Zemeckis>
<Titanic, James Cameron>
<Rain Man, Barry Levinson>
…

Passage Retrieval from Raw Text

Robert Zemeckis is director of Forrest Gump

Titanic was a movie directed by James Cameron

Barry Levinson was famous as the director of Rain Man

Relation Patterns

Film.Film.Director

[Director] *is director of* **[Film]**	0.84
[Film] *was a movie directed by* **[Director]**	0.81
[Director] *was famous as the director of* **[Film]**	0.77
…	…

$$[\text{Director}] \rightarrow \langle \text{is director of } [\text{Film}]_1 \quad \lambda x.\, \text{Film\_Director\_Film}(x, [\text{Film}]_1) \rangle$$

film.actor.film — Query!

Results	
[film] starred [actor]	0.210230664938471
[film] starring [actor]	0.210230664938471
[film] stars [actor]	0.210230664938471
[actor] starred in [film]	0.0322559719953288
[actor] stars in [film]	0.0322559719953288
[film] is played by [actor]	0.0240061833253798
[film] was played by [actor]	0.0240061833253798
[film] were played by [actor]	0.0240061833253798
[actor] played [film]	0.00972055029550689
[actor] plays [film]	0.00972055029550689

Lambda Dependency-based Compositional Semantics (λ-DCS)

- λ-DCS is another formal language
- λ-DCS attempts to remove explicit use of variables, so it is simpler than λ-Calculus
- λ-DCS is specially designed for Freebase Knowledge Graph

Question	people who have lived in Seattle ?
λ-Calculus	$\lambda x. \exists y. \mathrm{PlaceLived}(x, y) \wedge \mathrm{Location}(y, \mathrm{Seattle})$
λ-DCS	$\mathrm{PlaceLived}.\mathrm{Location}.\mathrm{Seattle}$

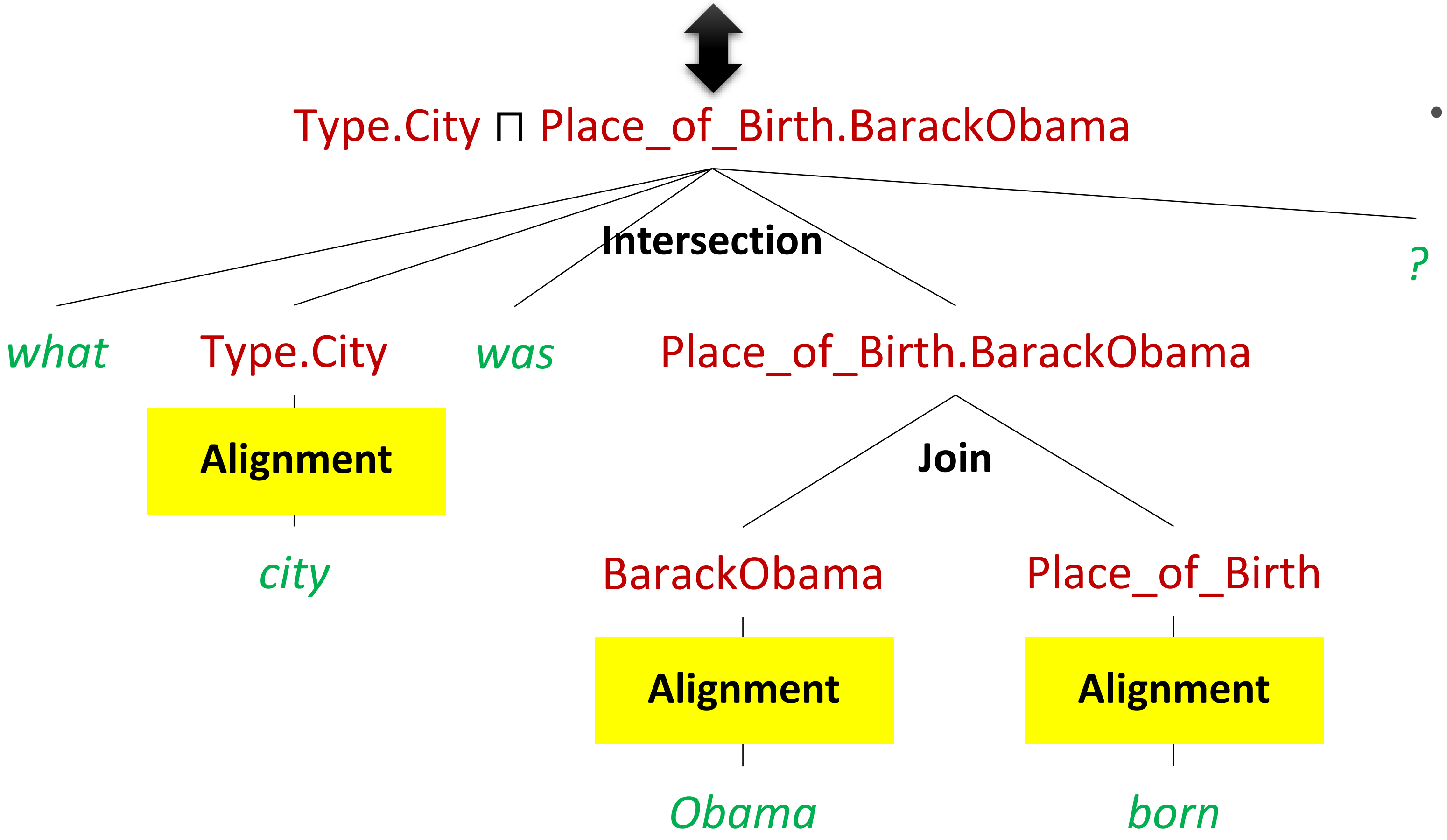
Building Blocks of λ-DCS

- **Entity**
  - NL: Seattle
  - LF: $\mathrm{Seattle} \leftrightarrow \lambda x.\,[x = \mathrm{Seattle}]$
- **Predicate**
  - NL: birthday
  - LF: $\mathrm{PlaceOfBirth} \leftrightarrow \lambda x \lambda y.\,\mathrm{PlaceOfBirth}(x, y)$
- **Join Operator (.)**
  - NL: people who was born in Seattle
  - LF: $\mathrm{PlaceOfBirth.Seattle} \leftrightarrow \lambda x.\,\mathrm{PlaceOfBirth}(x, \mathrm{Seattle})$
- **Intersection Operator (⊓)**
  - NL: people who are scientist and born in Seattle
  - LF: $\mathrm{Profession.Scientist} \sqcap \mathrm{PlaceOfBirth.Seattle} \leftrightarrow \lambda x.\,\mathrm{Profession}(x, \mathrm{Scientist}) \wedge \mathrm{PlaceOfBirth}(x, \mathrm{Seattle})$
- **Union Operator (⊔)**
  - NL: Movie directed or played by Tom Hanks
  - LF: $\mathrm{Directed\_By.TomHanks} \sqcup \mathrm{Starred\_By.TomHanks} \leftrightarrow \lambda x.\,\mathrm{Directed\_By}(x, \mathrm{Tom\ Hanks}) \vee \mathrm{Starred\_By}(x, \mathrm{TomHanks})$
- **…**

Semantic Parsing with λ-DCS

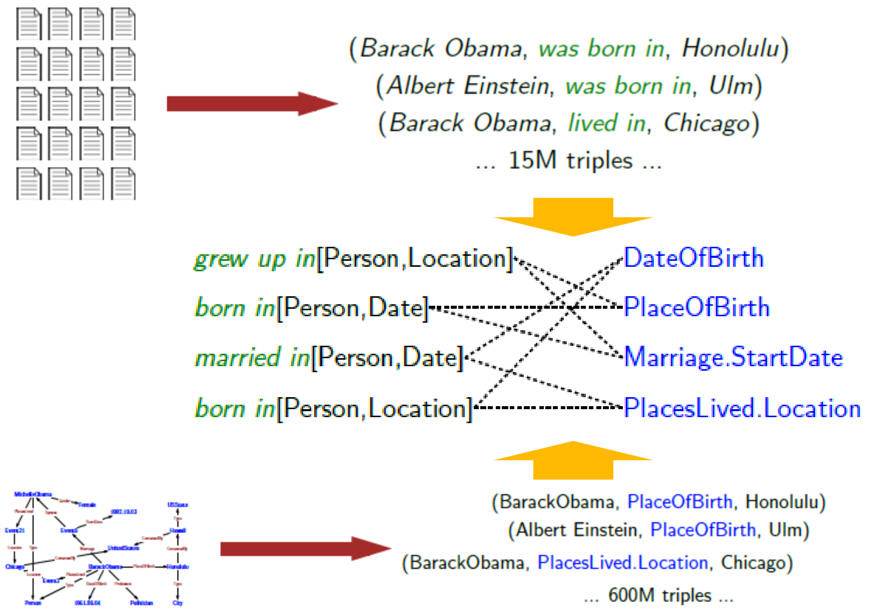
(Berant et al., 2013)

$$\lambda x.\, \mathrm{Type}(\mathrm{City}, x) \wedge \mathrm{Place\_of\_Birth}(\mathrm{Barack\ Obama}, x)$$



Alignment

- Map mentions to KB entities\predicates



Training Semantic Parser with Q-LF Pairs as Supervision

(Zettlemoyer and Collins, 2007)

$$\widehat{LF} = \underset{LF \in \{LF_1,\ldots,LF_n\}}{\operatorname{argmax}} P(LF|Q; \bar{\mathbf{w}}) = \underset{LF \in \{LF_1,\ldots,LF_n\}}{\operatorname{argmax}} \sum_i \mathbf{w_i} \cdot \Phi_i(LF, \mathcal{Q})$$

But labeling logical forms for natural language is very expensive.

- Remember **Perception**?
- Update weight if there is a mistake

$$w_i^+ = w_i - \alpha \frac{\partial E}{\partial w_i} = w_i + \Phi_i(LF^{true}, \mathcal{Q}) - \Phi_i(\widehat{LF}, \mathcal{Q})$$

  - increase score of positive examples
  - decrease score of negative examples
  - no change, if highest scoring answer is correct

Structured Perceptron Algorithm

```
create map w
for I iterations
    for each labeled pair X, Y_prime in the data
        Y_hat = HMM_VITERBI(w, X)
        phi_prime = CREATE_FEATURES(X, Y_prime)
        phi_hat = CREATE_FEATURES(X, Y_hat)
        w += phi_prime - phi_hat
```

Training Semantic Parser with QA Pairs as Weak Supervision

(Bao et al., 2014; Berant et al., 2013)

- Use answers as guides to train parameters in the semantic parser

$$\widehat{\mathrm{LF}} = \underset{\mathrm{LF} \in \{\mathrm{LF}_1, \ldots, \mathrm{LF}_n\}}{\operatorname{argmax}} \mathrm{P}(\mathrm{LF}|\mathrm{Q}; \overline{\mathbf{w}}) = \underset{\mathrm{LF} \in \{\mathrm{LF}_1, \ldots, \mathrm{LF}_n\}}{\operatorname{argmax}} \sum_{\mathrm{i}} \mathbf{w}_{\mathrm{i}} \cdot \Phi_{\mathrm{i}}(\mathrm{LF}, \mathcal{Q})$$

initial weights

$(1, 1, \ldots, 1) \xrightarrow{1^{st} \text{ round parsing}}$

Logical Form N-best		***Answer N-best***	
T1	✗	A1	✗
T2	✗	A2	✗
T3	✓	A3	✓
T4	✗	A4	✗
T5	✓	A5	✓

Training Semantic Parser with QA Pairs as Weak Supervision

(Bao et al., 2014; Berant et al., 2013)

- Use answers as guides to train parameters in the semantic parser

$$\widehat{\mathrm{LF}} = \underset{\mathrm{LF} \in \{\mathrm{LF}_1, \dots, \mathrm{LF}_n\}}{\operatorname{argmax}} P(\mathrm{LF} | Q; \overline{\mathbf{w}}) = \underset{\mathrm{LF} \in \{\mathrm{LF}_1, \dots, \mathrm{LF}_n\}}{\operatorname{argmax}} \sum_i \mathbf{w}_i \cdot \Phi_i(\mathrm{LF}, Q)$$

updated weights

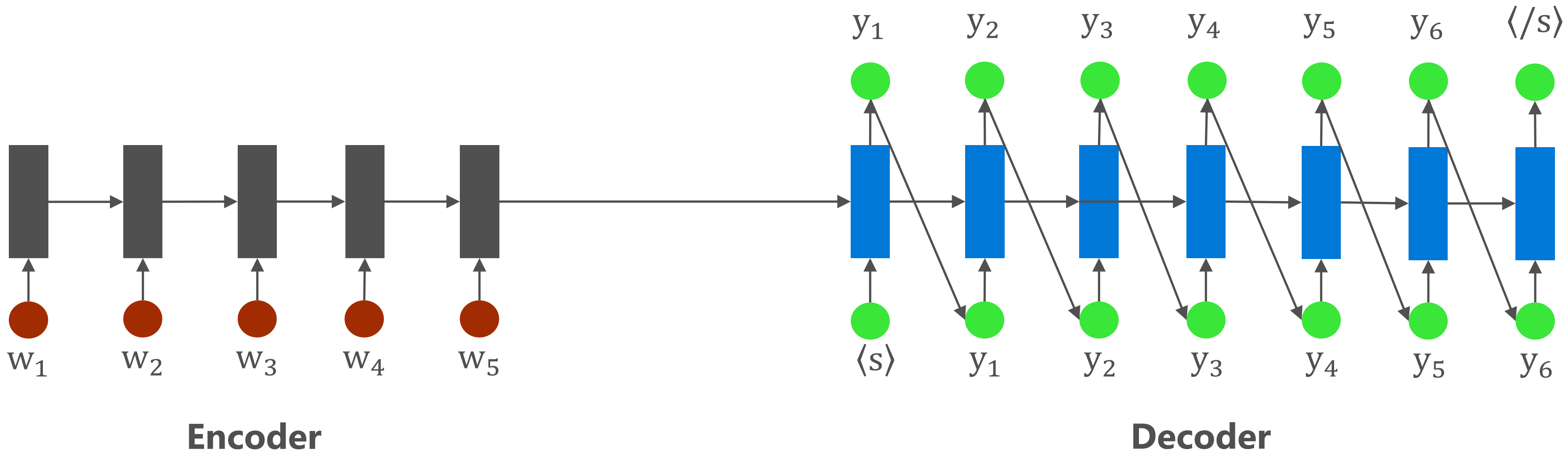
(0.2, -1.3, . . . , 0.7)

← 1st round optimization

Logical Form N-best		***Answer N-best***	
T3	✓	A3	✓
T5	✓	A5	✓
T1	✗	A1	✗
T4	✗	A4	✗
T2	✗	A2	✗

Generic Framework of Neural Network-based Semantic Parsing

- Perform semantic parsing as neural machine translation
  - **Encoder** considers a question as source language, encodes it into hidden states using RNN
  - **Decoder** considers a logical form as target language, generate it word-by-word based on question encoding using RNN

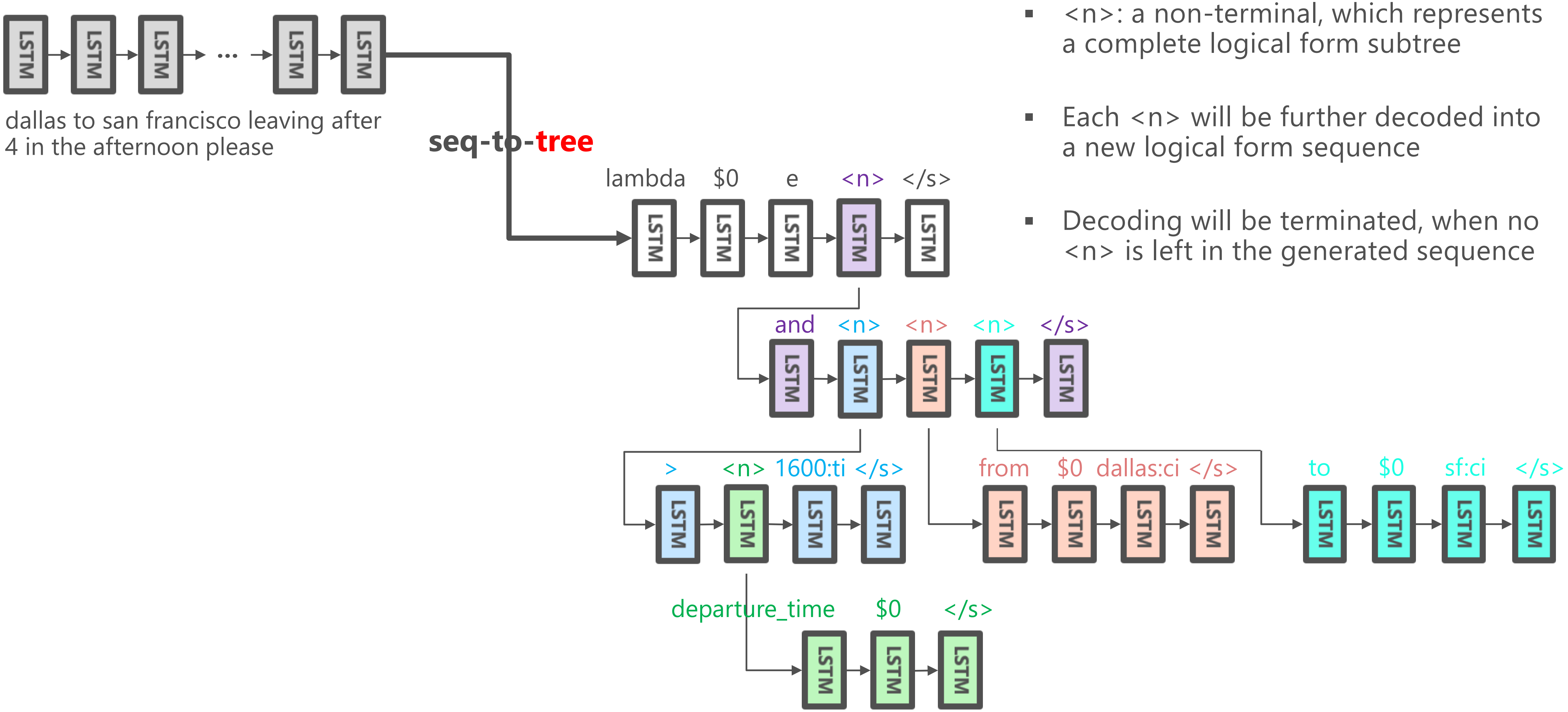


- Issue
  - Seq-to-Seq model ignores the **hierarchical structure** of logical forms

dallas to san francisco leaving after 4 in the afternoon please ➡ (Lambda $0 e (and (>(departure_time $0) 1600:ti) (from $0 dallas:ci) (to $0 sf:ci)))

Semantic Parsing with Seq-to-Tree Neural Network

(Dong and Lapata, 2016; Jia and Liang, 2016)



- <n>: a non-terminal, which represents a complete logical form subtree
- Each <n> will be further decoded into a new logical form sequence
- Decoding will be terminated, when no <n> is left in the generated sequence

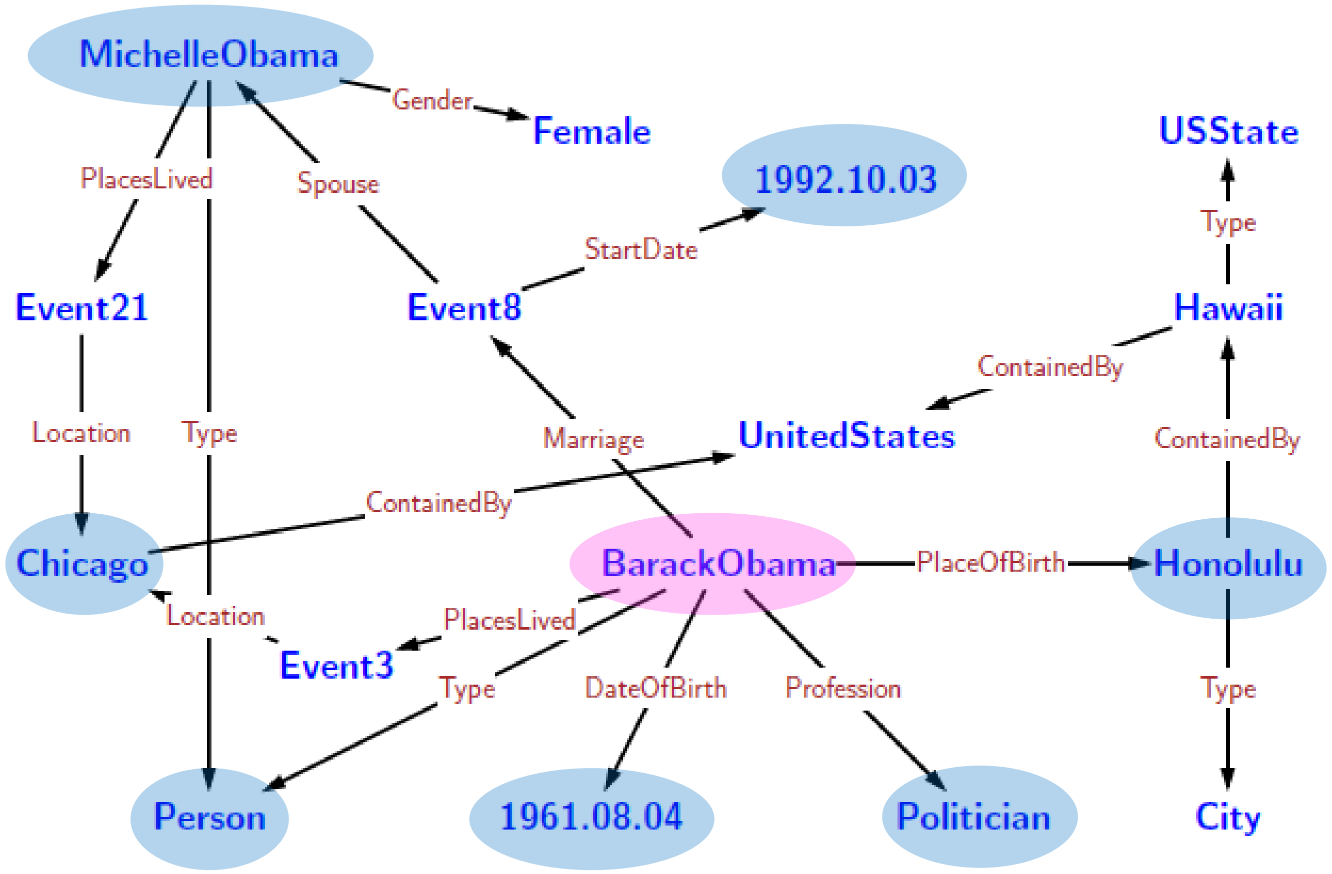
Answer Lookup

- Find answers by executing a logical form against KB is straightforward…

Outline of KBQA

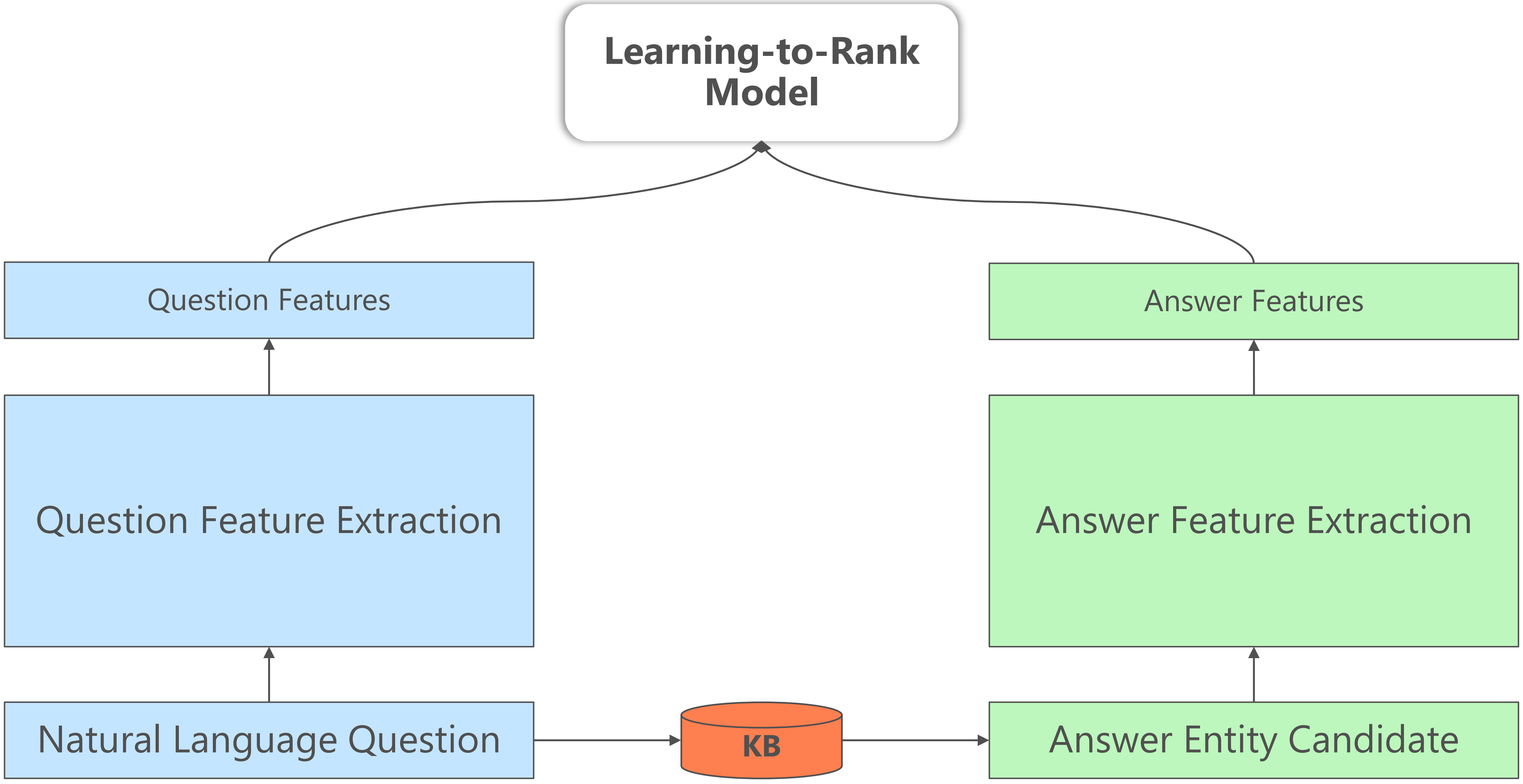
- Semantic Parsing-based KBQA
  1. What is **logical form**?
  2. How to **parse** a question into its logical form?
  3. How to **execute** a logical form against KB?
- Answer Ranking-based KBQA
  1. How to **select** answer candidates?
  2. How to **represent** answer candidates?
  3. How to **rank** answer candidates?

Answer Candidate Selection



- **Input question**
  - Where was Obama born?
- **Question entity detection**
  - Obama → Barack Obama
- **Answer candidate selection**
  - Entities connected to the question entity within n hops
  - Usually, $n = 1$ or 2

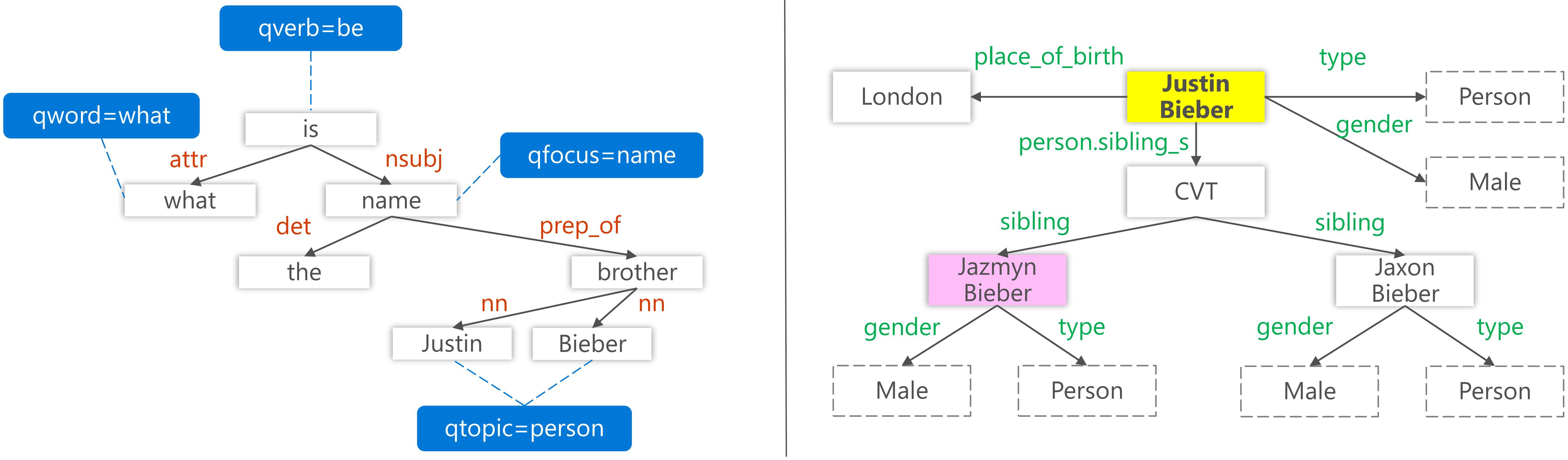
Answering Ranking with Features



An Example

(Yao and Durme, 2014)

what is the name of Justin Bieber brother

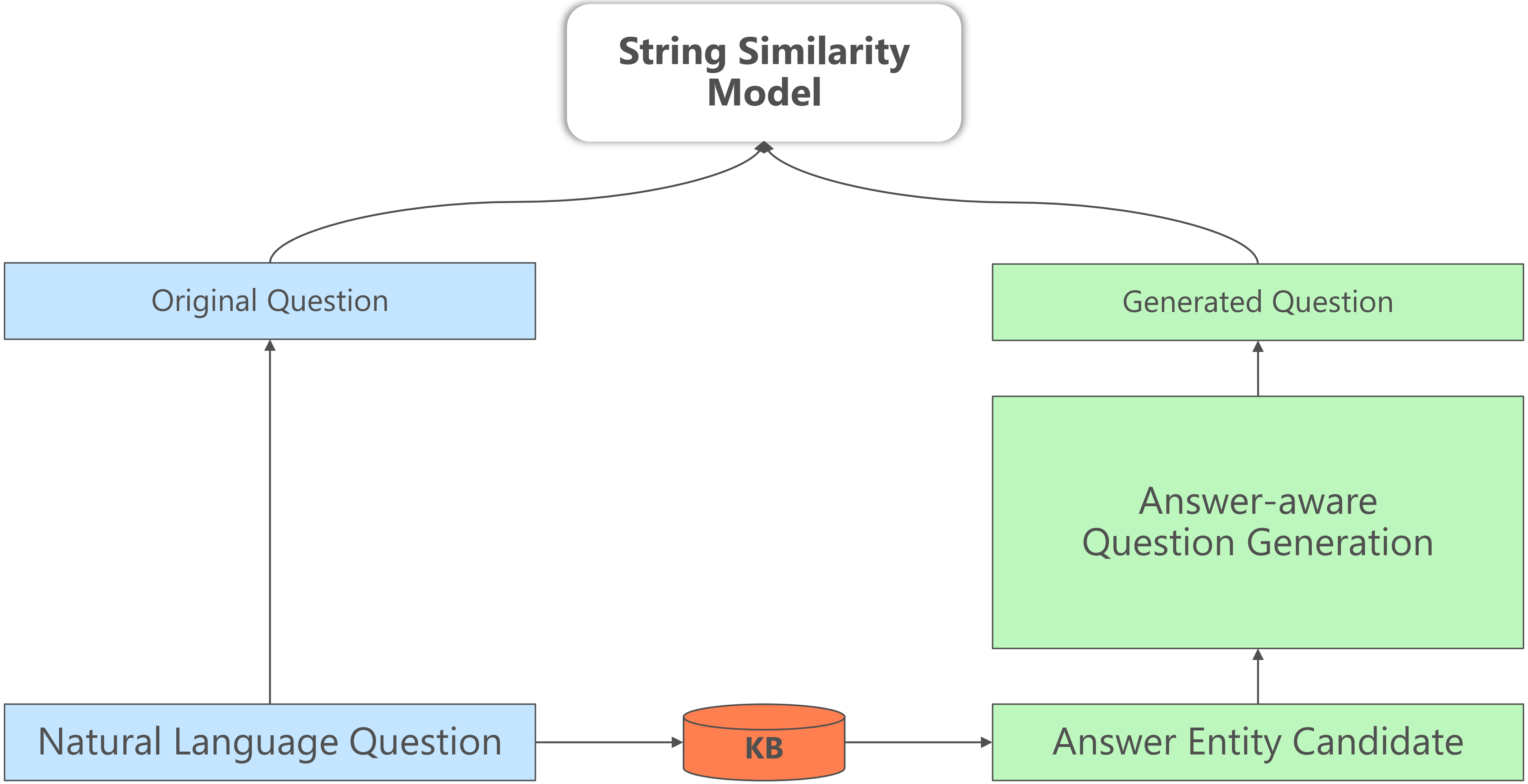


$$\tilde{A} = \underset{A}{\operatorname{argmax}} \sum_i \lambda_i \cdot h_i(QG, TG; A)$$

Each feature is a pairwise concatenation of a question graph feature and a topic graph feature of a specific answer node candidate.

Question Graph (QG) **Topic Graph (TG)**

Answer Ranking with Question Generation



An Example
(Berant and Liang, 2014)

which city was Obama born ?

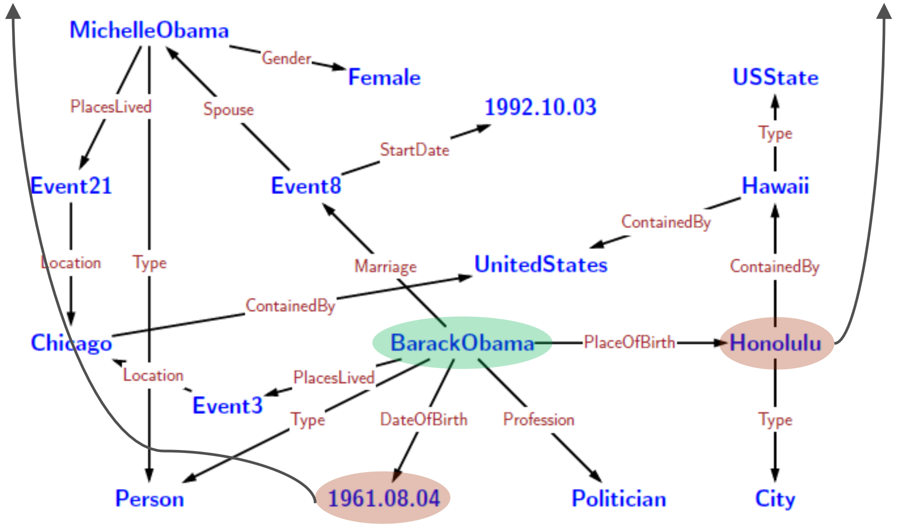
Ranking Model
(question-generated question)

When is the date of birth of Barack Obama ? … What city is the place of birth of Barack Obama ?

Pattern-based Question Generation **Pattern-based Question Generation**

DataOfBirth.BarackObama … Type.City ∩ PlaceOfBirth.BarackObama

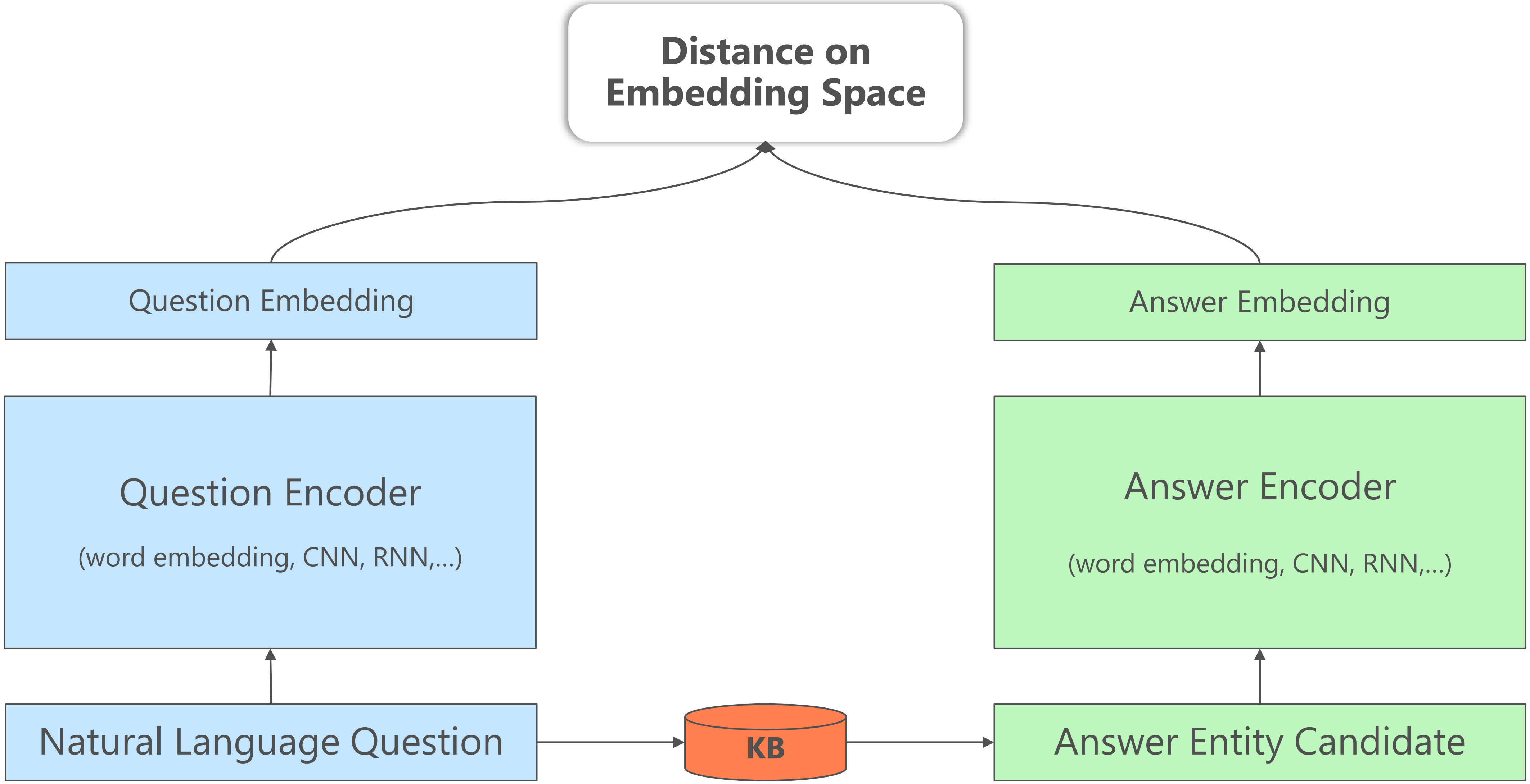
Predicate POS	Question Generation Pattern
NP	What **TYPE** is the **NP** of **ENTITY** ?
NP VP	What **NP** is **VP** by **ENTITY** ?
…	…



Ranking Model

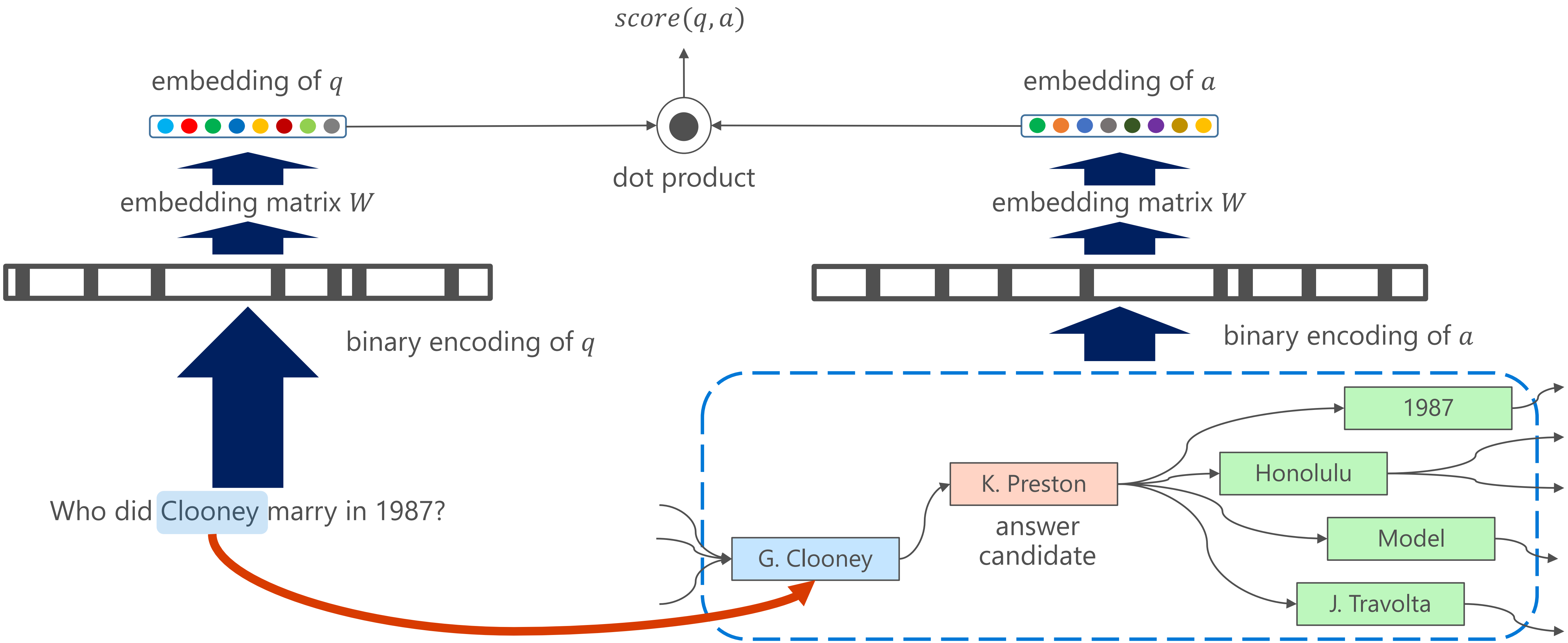
- Compute similarity between input question and each generated question
- Features
  - Association Model (paraphrasing)
  - Vector Space Model (word embedding)

Answering Ranking with Embedding

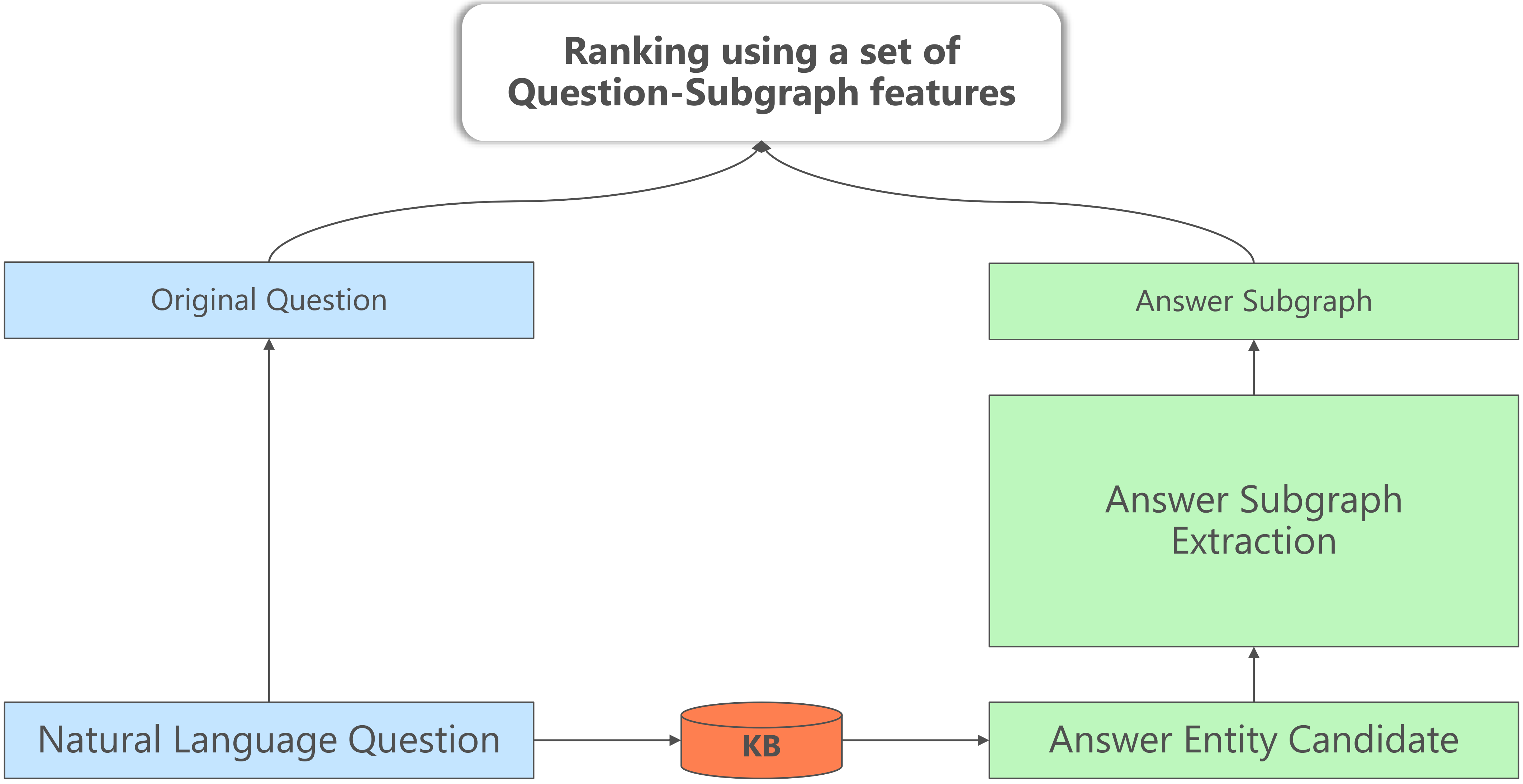


An Example

(Bordes et al., 2014)



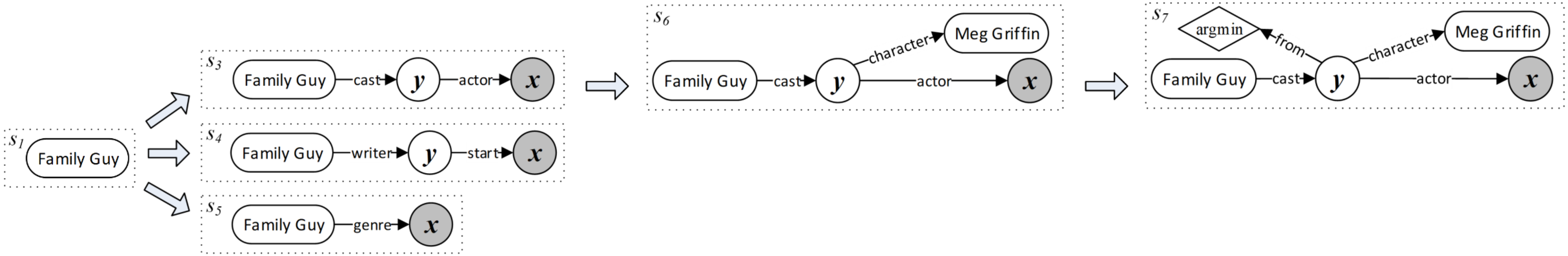
Answering Ranking with Subgraph



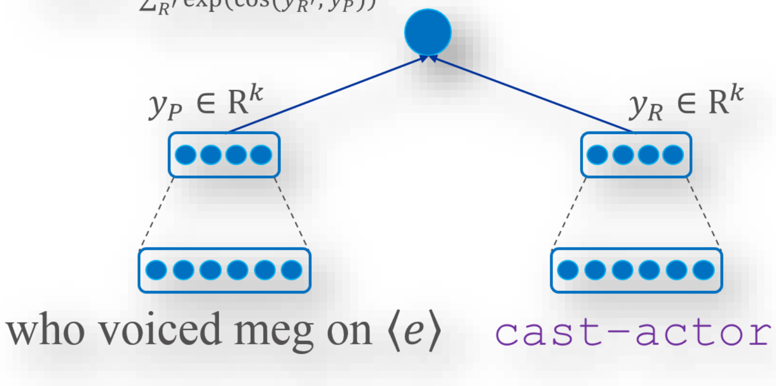
An Example

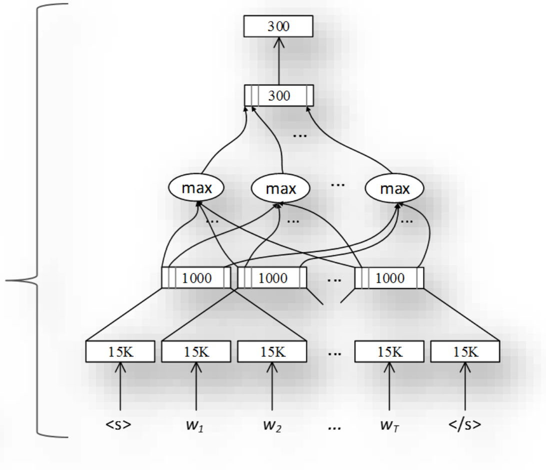
(Yih et al., 2015)

Who first voiced Meg on Family Guy ?



$$P(R|P) = \frac{\exp(\cos(y_R, y_P))}{\sum_{R'} \exp(\cos(y_{R'}, y_P))}$$





Semantic Parsing-based KBQA vs. Answer Ranking-based KBQA

- Compared on **WebQuestions dataset**
  - http://nlp.stanford.edu/software/sempre/

- Data statistic
  - 5,810 Q-A pairs (English) (questions are sampled from Google query log)
  - Most of them are one-hop factoid questions

- Citation
  - Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang. Semantic Parsing on Freebase from Question-Answer Pairs. EMNLP, 2013

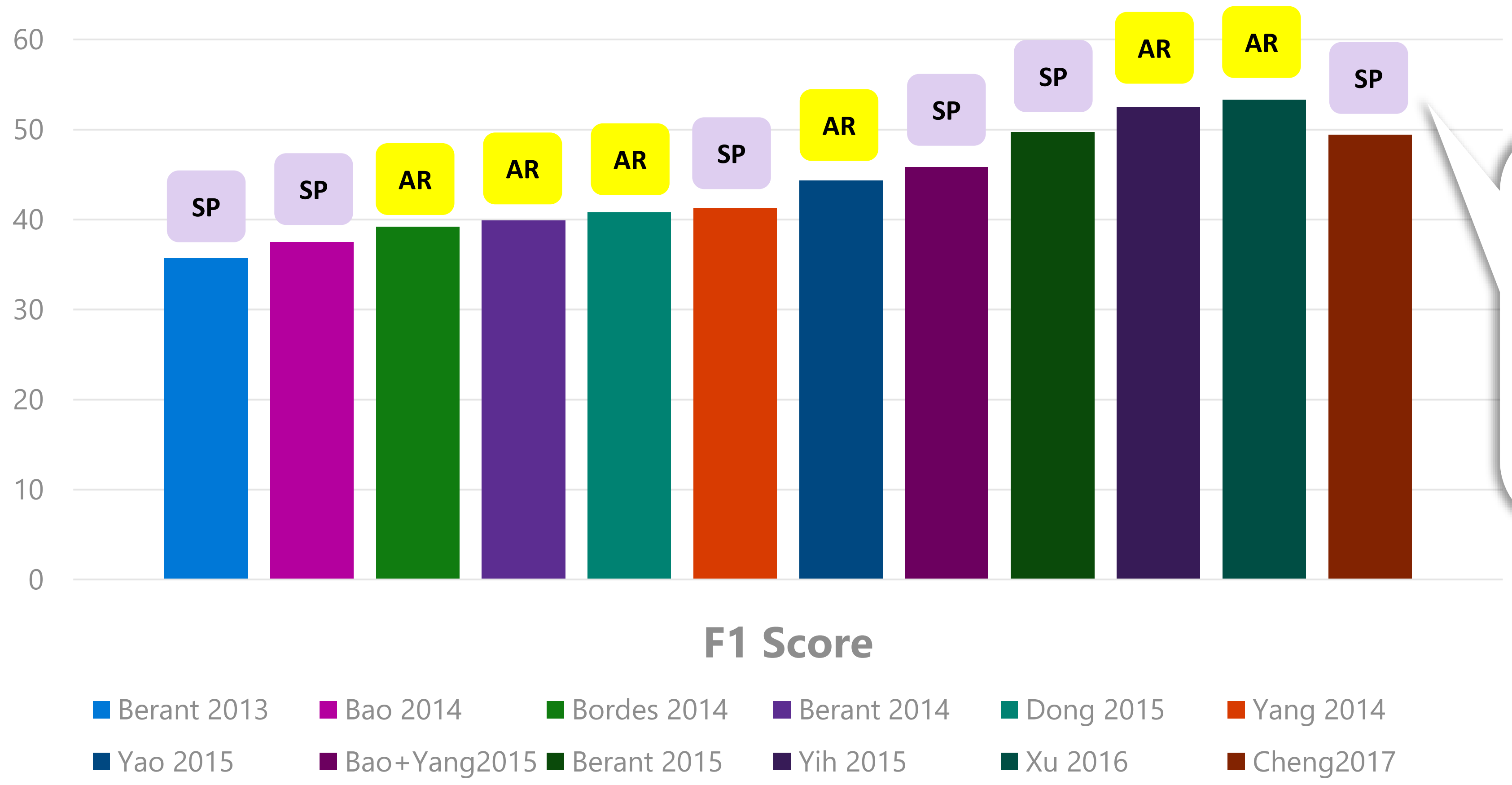
- Data example

what is the name of Justin Bieber brother ?
http://www.freebase.com/view/en/justin_bieber
Jazmyn Bieber

what state does Salena Gomez ?
http://www.freebase.com/view/en/selena_gomez
New York City

Semantic Parsing-based KBQA vs. Answer Ranking-based KBQA

- Evaluation results on WebQuestions using F1 Score as metric



- **AR**: Answer Ranking-based KBQA
- **SP**: Semantic Parsing-based KBQA
- AR-based methods perform better than SP-based methods

TableQA

- Two types of approaches
  - Semantic parsing-based approaches
  - Answer ranking-based approaches
- Similar to KBQA approaches
- Skip here, due to time limitation

which city hosted Summer Olympic in 2008 ?

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Pairs	France	24
…	…	…	…
2004	Athens	Greece	201
2008	Beijing	China	204

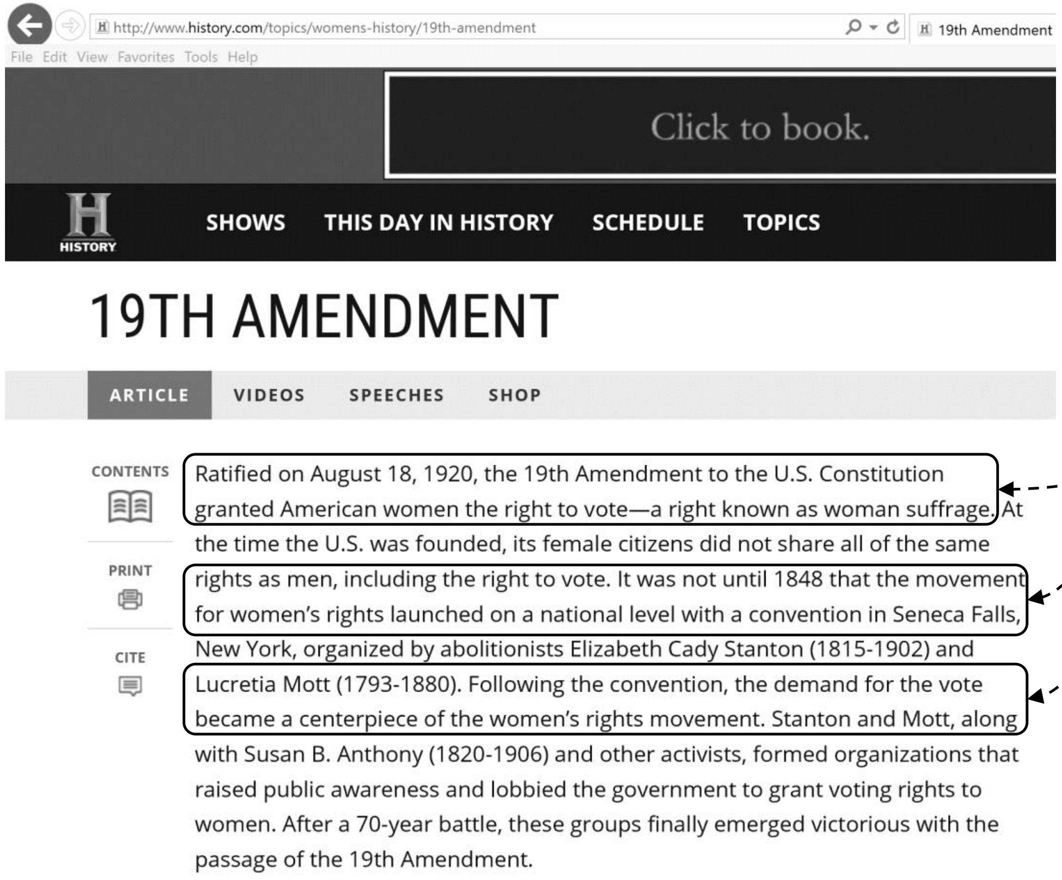
Table-based QA (TBQA)

Beijing

QA based on Unstructured Data

PassageQA

- Given a question, find its answer from existing documents



Document

Question:

when were women allowed to vote in the usa?

Passage Retrieval

Answer Candidate:

1. Ratified on August 18, 1920, the 19th Amendment to the U.S. Constitution granted American women the right to vote—a right known as woman suffrage.
2. It was not until 1848 that the movement for women's rights launched on a national level .
3. Following the convention, the demand for the vote became a centerpiece of the women's rights movement.
4. …

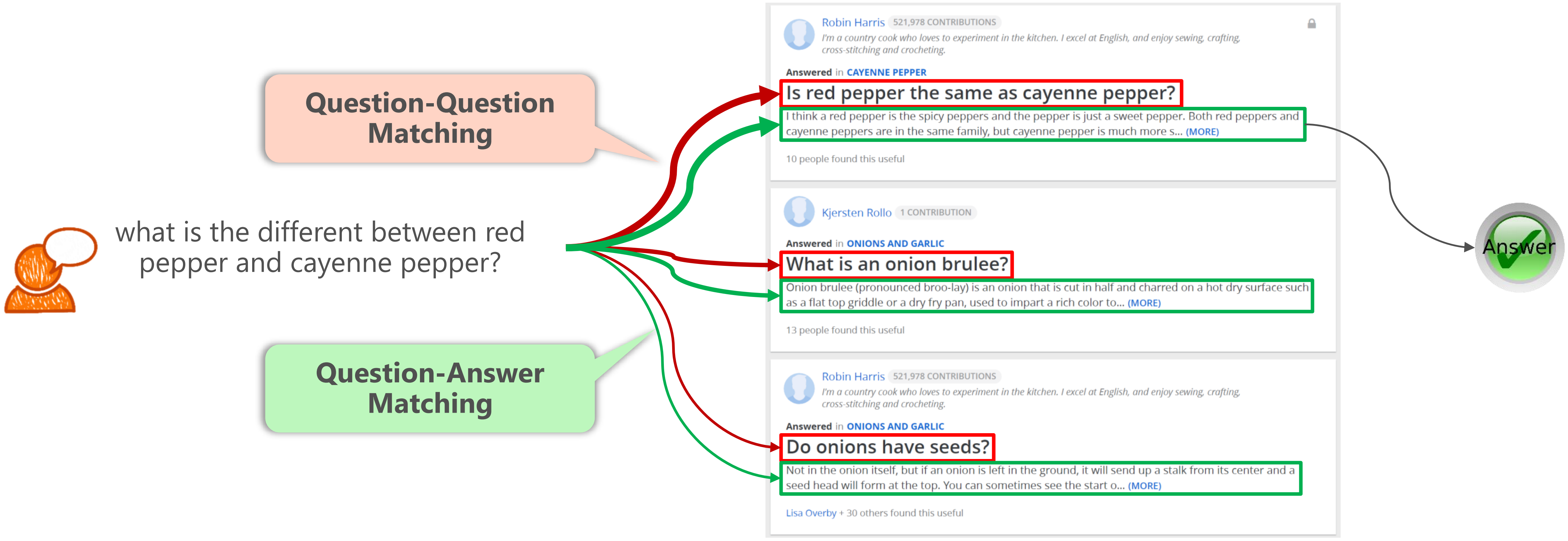
Question-Passage Matching

Answer:

Ratified on August 18, 1920, the 19th Amendment to the U.S. Constitution granted American women the right to vote—a right known as woman suffrage.

CommunityQA

- Given a question, find its answer from existing <question, answer> pairs



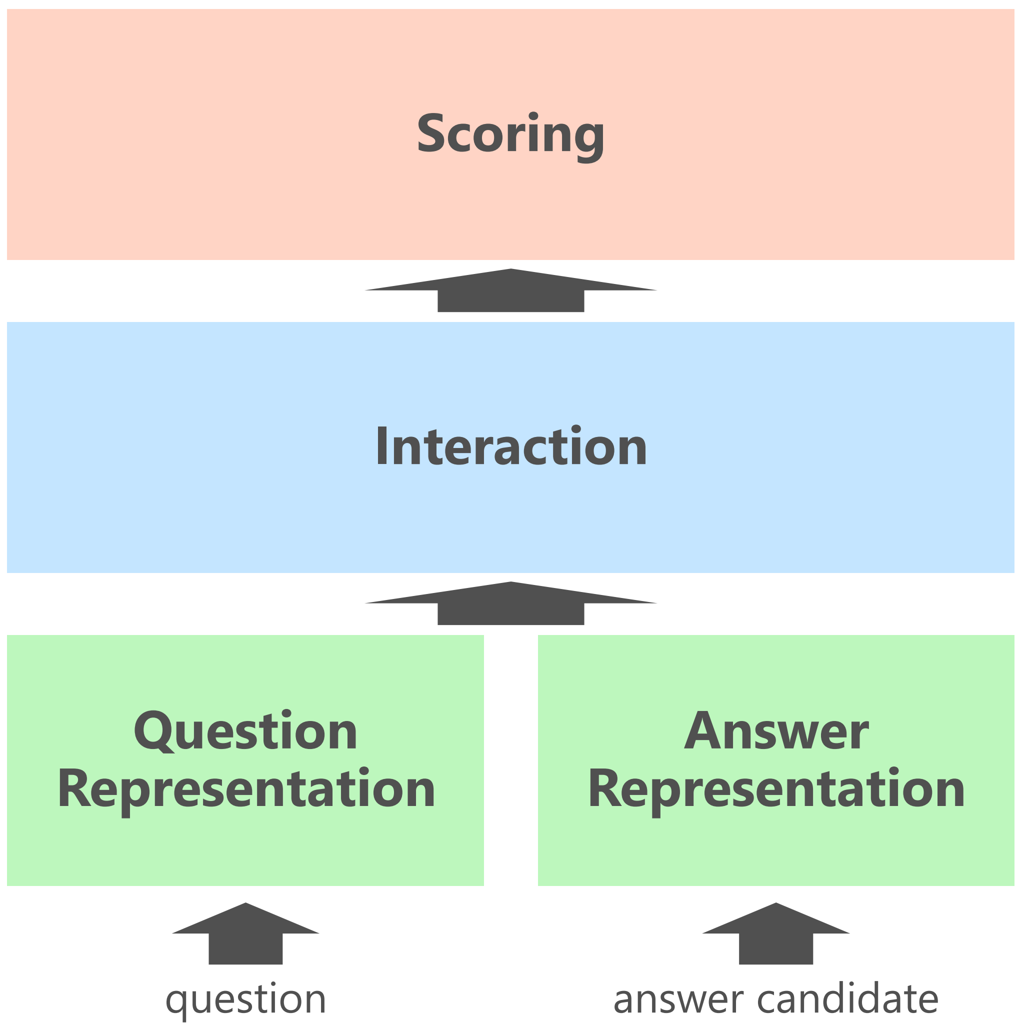
FAQ Pairs

Matching is the Fundamental Problem

- PassageQA
  - Question-Answer matching
- CommunityQA
  - Question-Question matching
  - Question-Answer matching

Neural Network-based Matching

1. **Representation** of question and answer
   - Word embedding
   - Convolution neural network (CNN)
   - Recurrent neural network (RNN)
   - …
2. **Interaction** between question and answer
   - Transition matrix
   - Attention
   - Tensor
   - …
3. **Scoring** relevance
   - Dot-product
   - Cosine
   - MLP
   - …

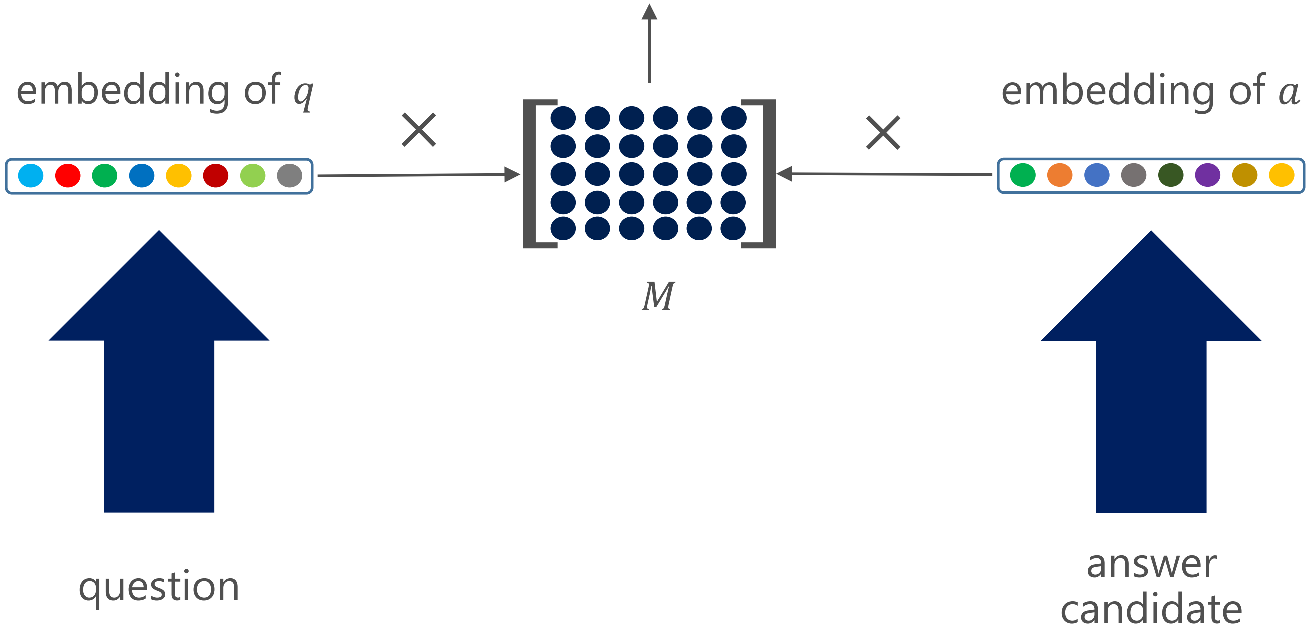


Word Embedding

(Yu et al., 2014)

$$p(y=1|q,a) = \mathrm{sigmoid}(v_q^{\mathrm{T}} M v_a + b)$$

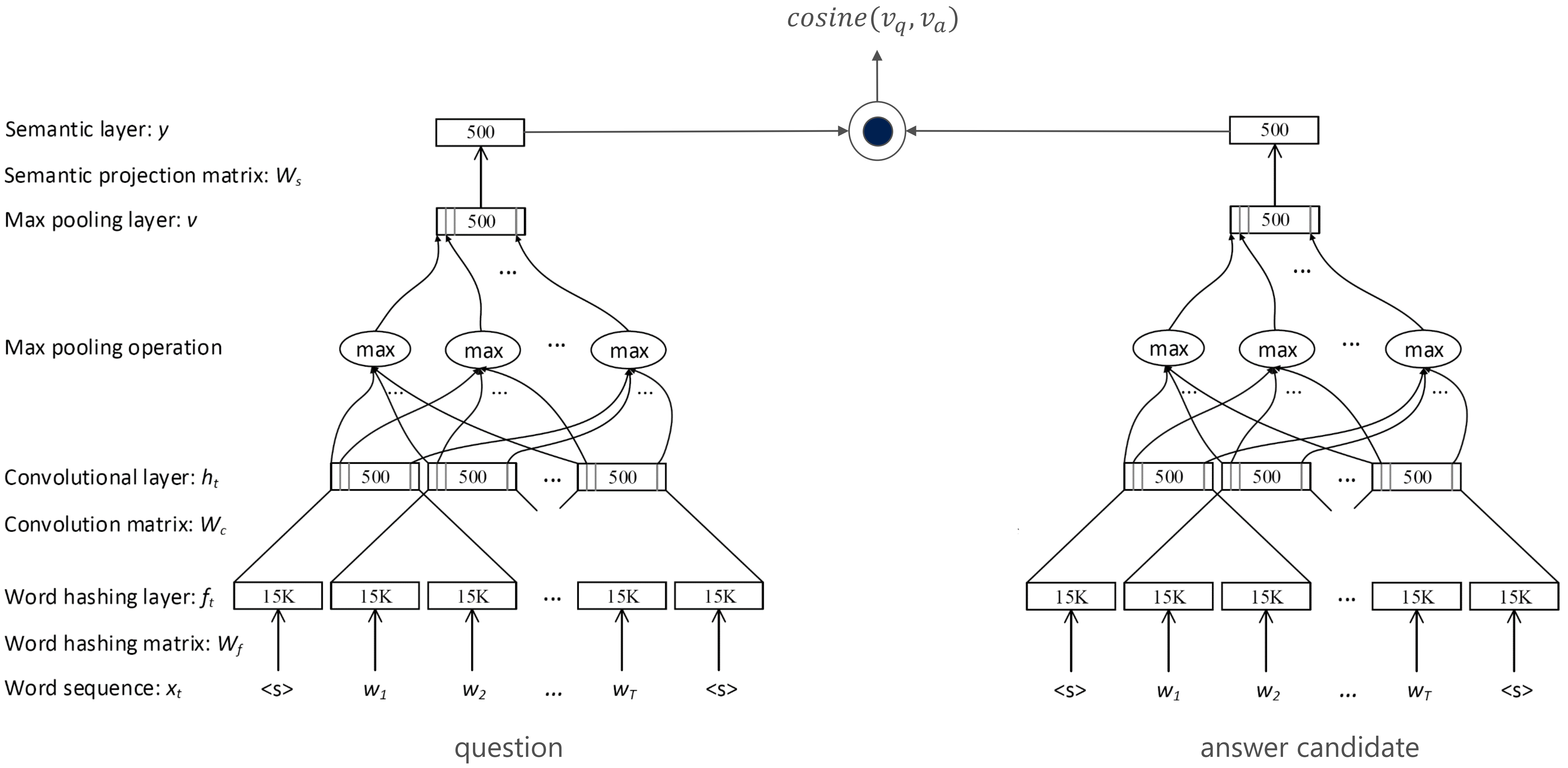
$$v_Q = \frac{1}{|Q|}\sum_{i=1}^{|Q|} v_{Q_i}$$



$$v_A = \frac{1}{|A|}\sum_{j=1}^{|A|} v_{A_j}$$

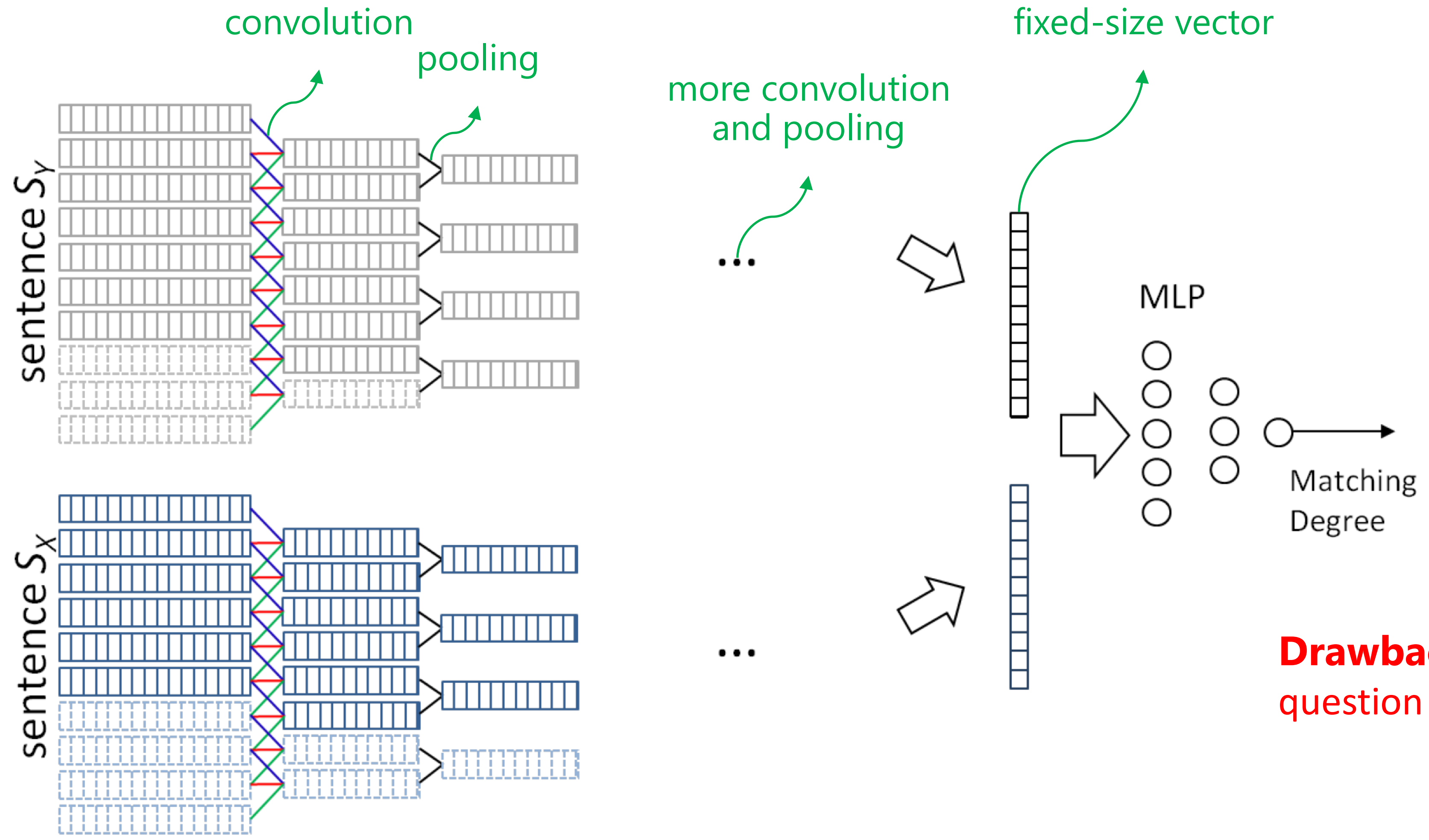
CNN

(Shen et al., 2014a; Shen et al., 2014b; Kim, 2014)



CNN + MLP

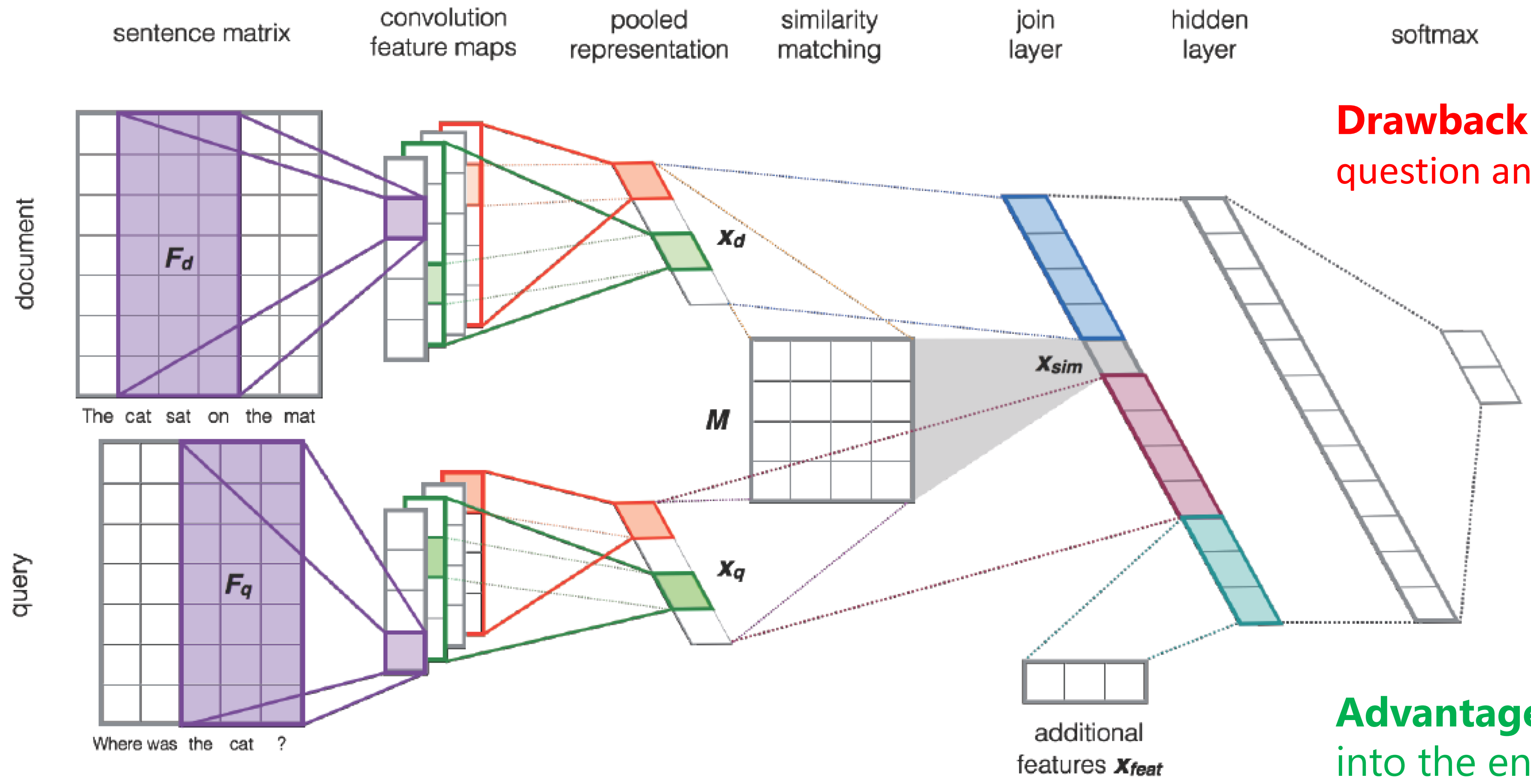
(Hu et al., 2015)



Drawback: defer the interaction between question and answer in the final MLP

CNN + MLP + External Features

(Severyn and Moschitti, 2015)

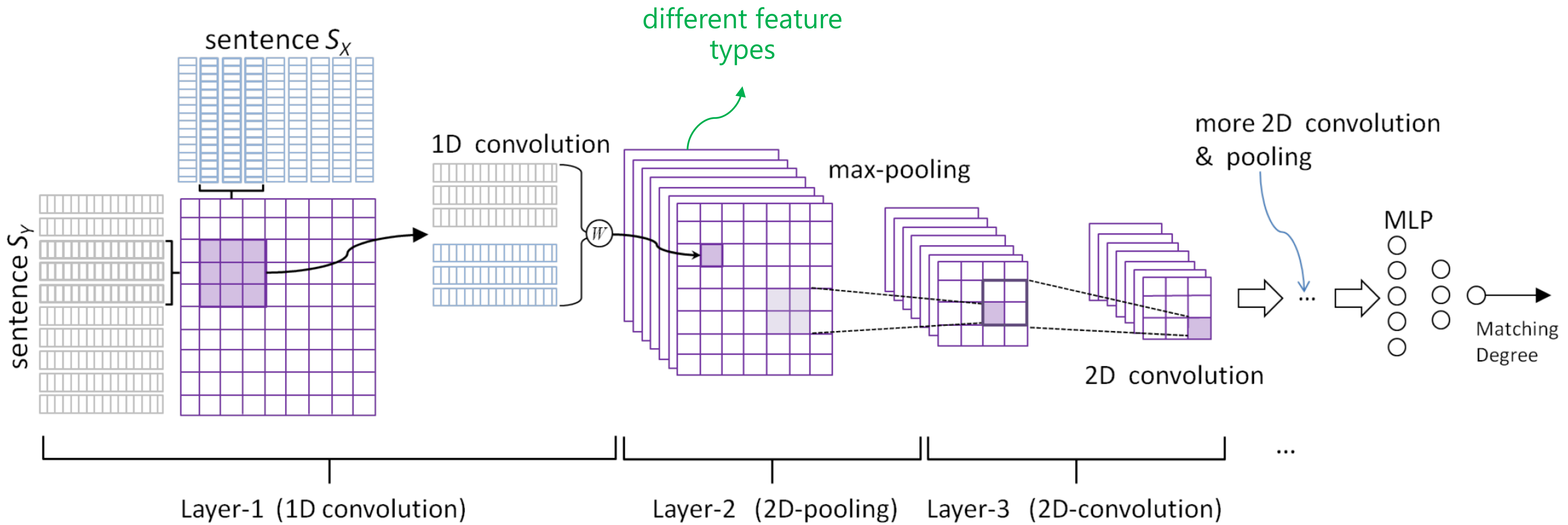


Drawback: defer the interaction between question and answer in the final MLP

Advantage: integrate external features into the end-to-end model

CNN + MLP + Early Interaction

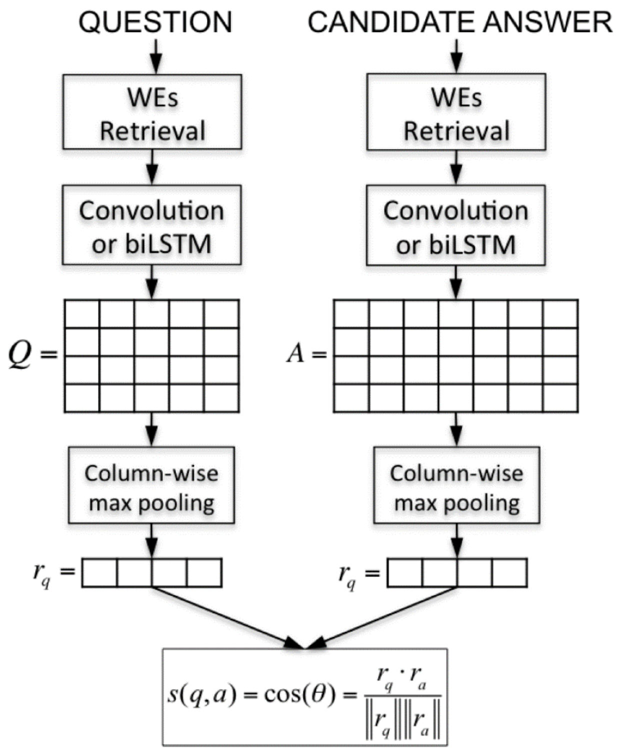
(Hu et al., 2014)



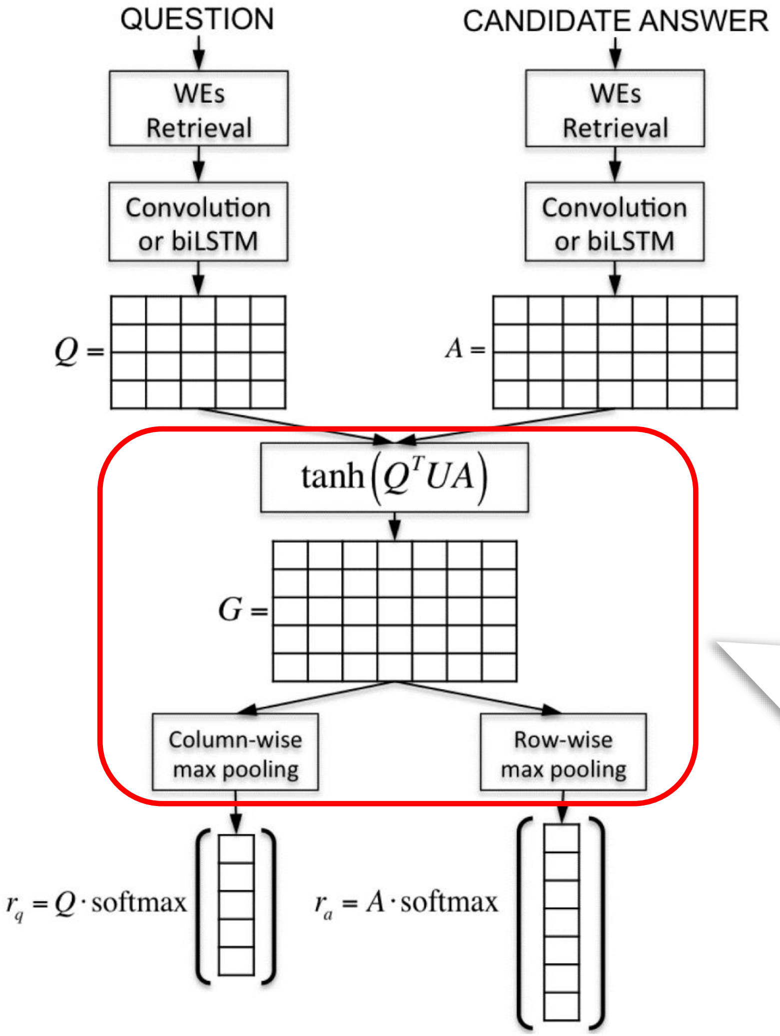
Advantage: let question and answer interact in early stage

CNN + Attentive Pooling

(Santos et al., 2016; Yin et al., 2016; Yan et al., 2016)



CNN



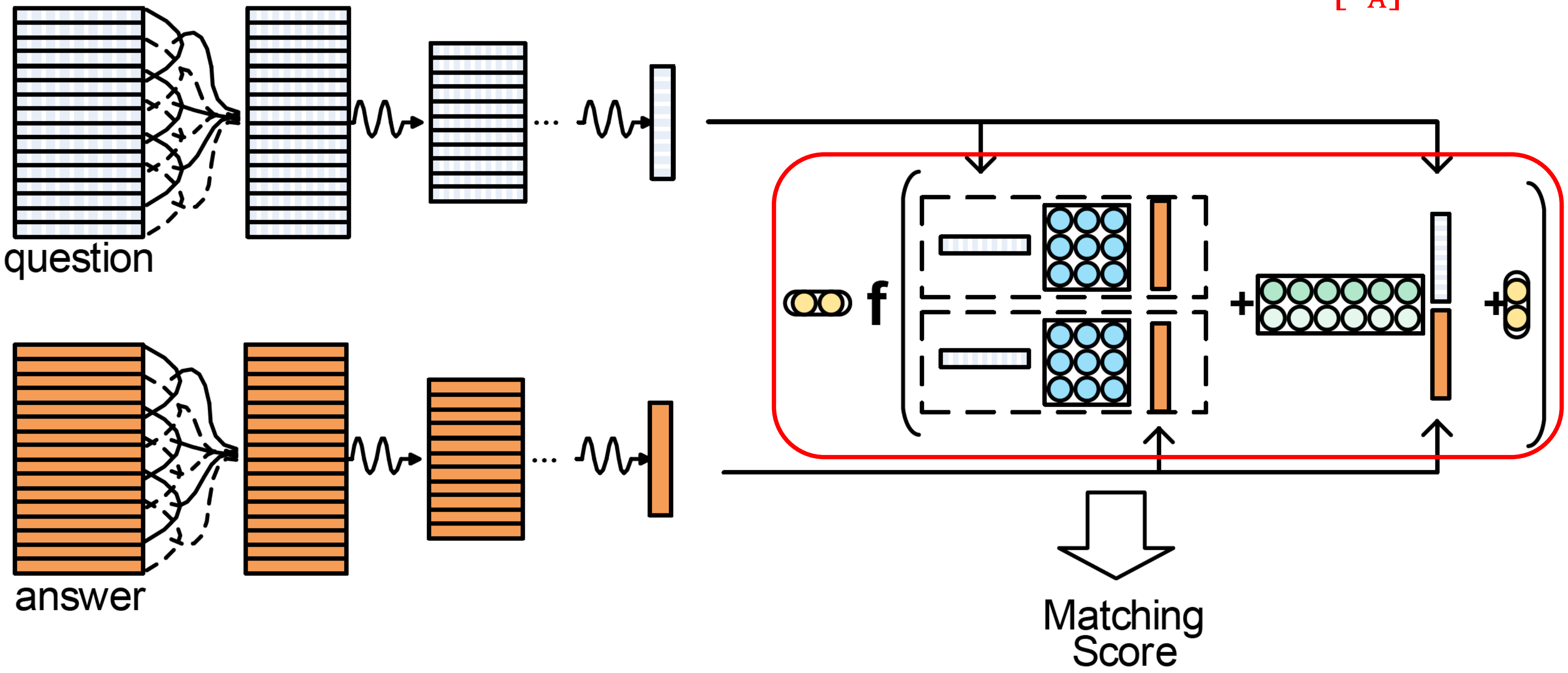
CNN with Attentive Pooling

- **Attention** allows question and answer to interact with each other
- **Pooling** takes the most influential ones for ranking

CNN + Tensor

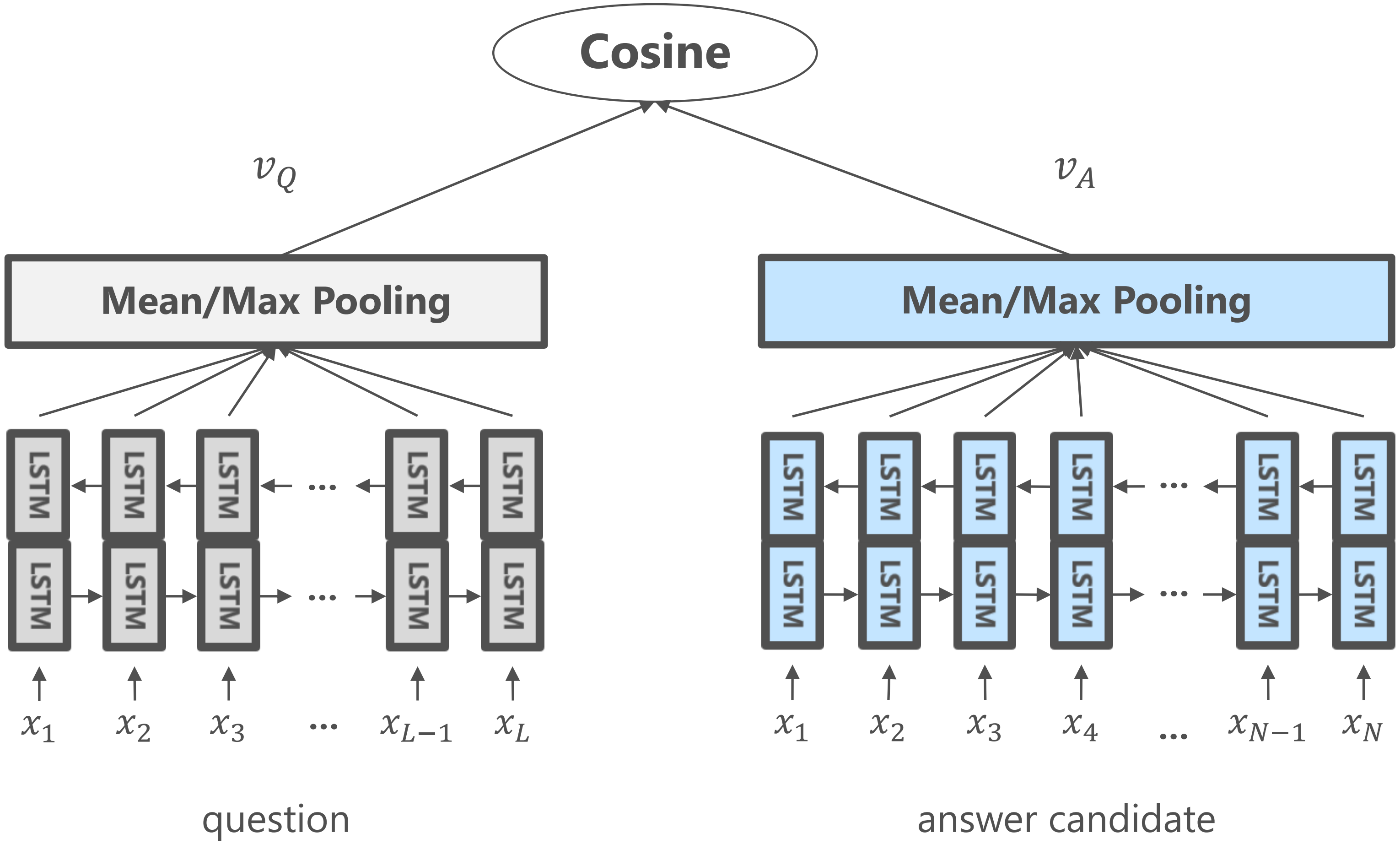
(Qiu and Huang, 2015; Socher et al., 2013)

$$s(Q,A) = u^T f\left(v_Q^T \mathbf{M}^{[1:r]} v_A + V \begin{bmatrix} v_Q \\ v_A \end{bmatrix} + b\right)$$



RNN

(Wang and Nyberg, 2015; Tan et al., 2015; Hsu et al., 2016)



$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$

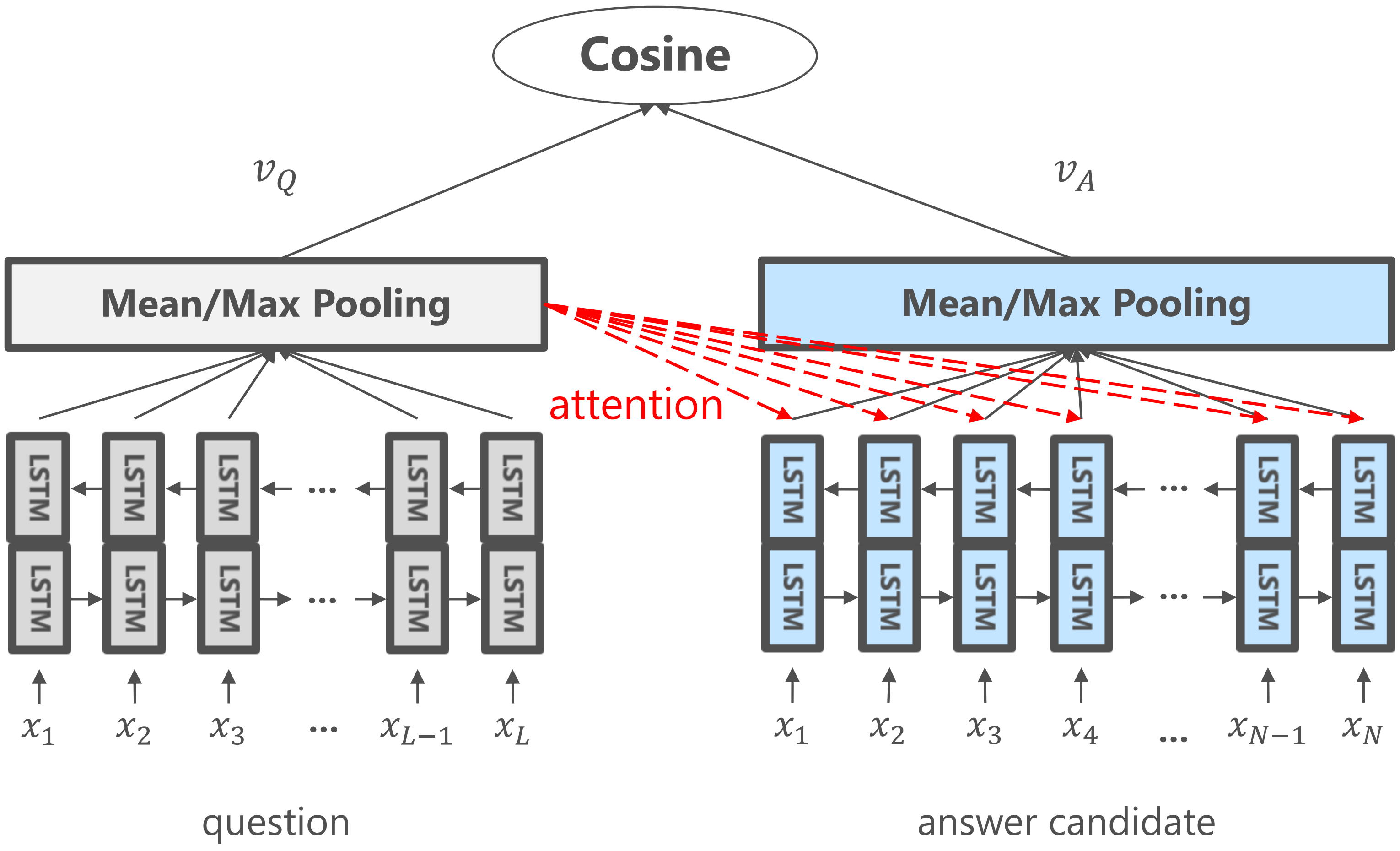
$$C_t = i_t * \tilde{C}_t + f_t * C_{t-1}$$

$$h_t = o_t * \tanh(C_t)$$

$$L = \max\{0, 1 - \mathrm{cosine}(q, a_+) + \mathrm{cosine}(q, a_-)\}$$

RNN with Attention

(Miao et al., 2016; Tan et al., 2015; Hsu et al., 2016)



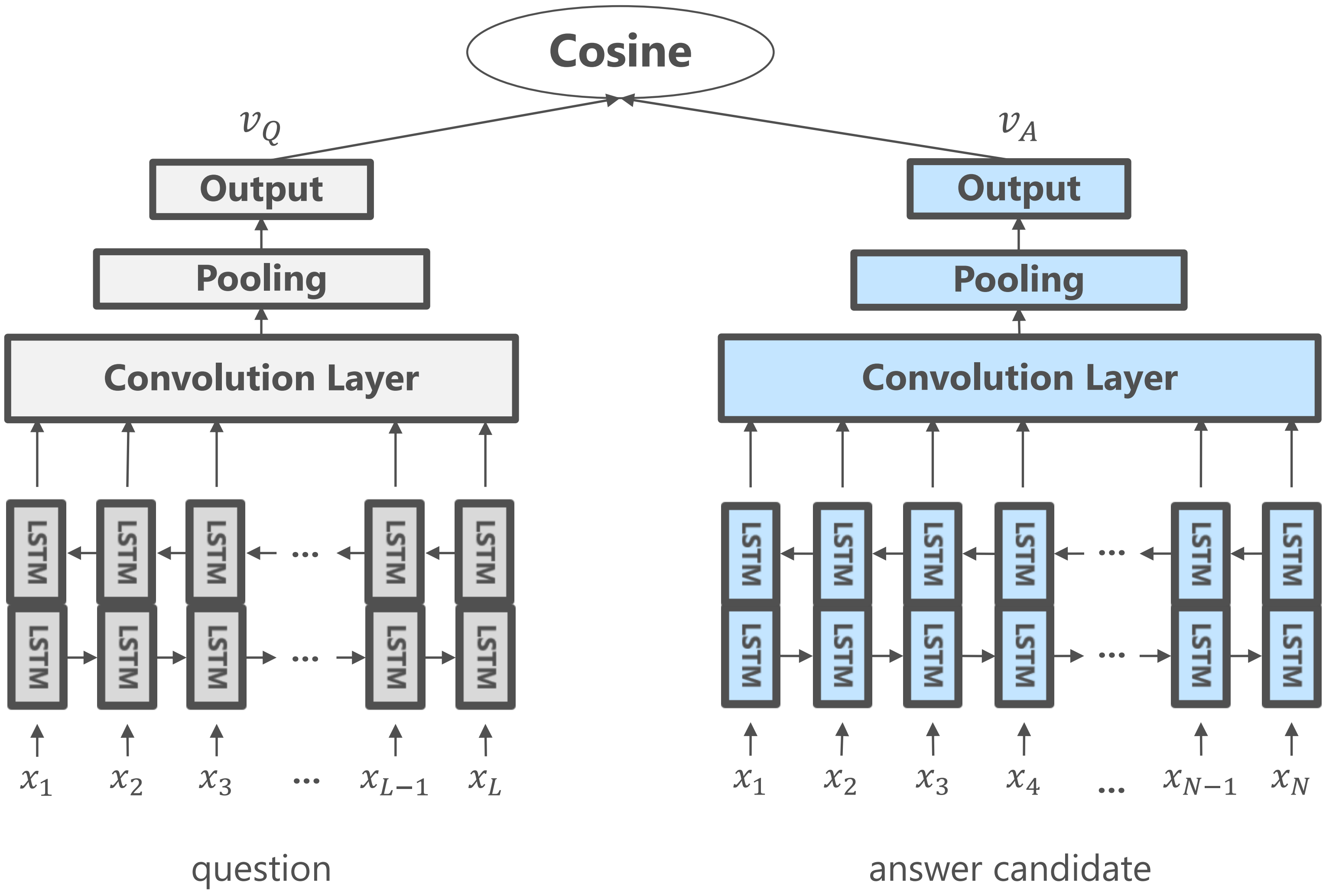
$$\tilde{h}_t = \alpha_t \cdot h_t$$

$$\alpha_t = \frac{\exp f(h_t, v_Q)}{\sum_{j=1}^{N} \exp f(h_j, v_Q)}$$

$$f(h_t, v_Q) = v_a^T \tanh(W_{aa} h_t + W_{aq} v_Q)$$

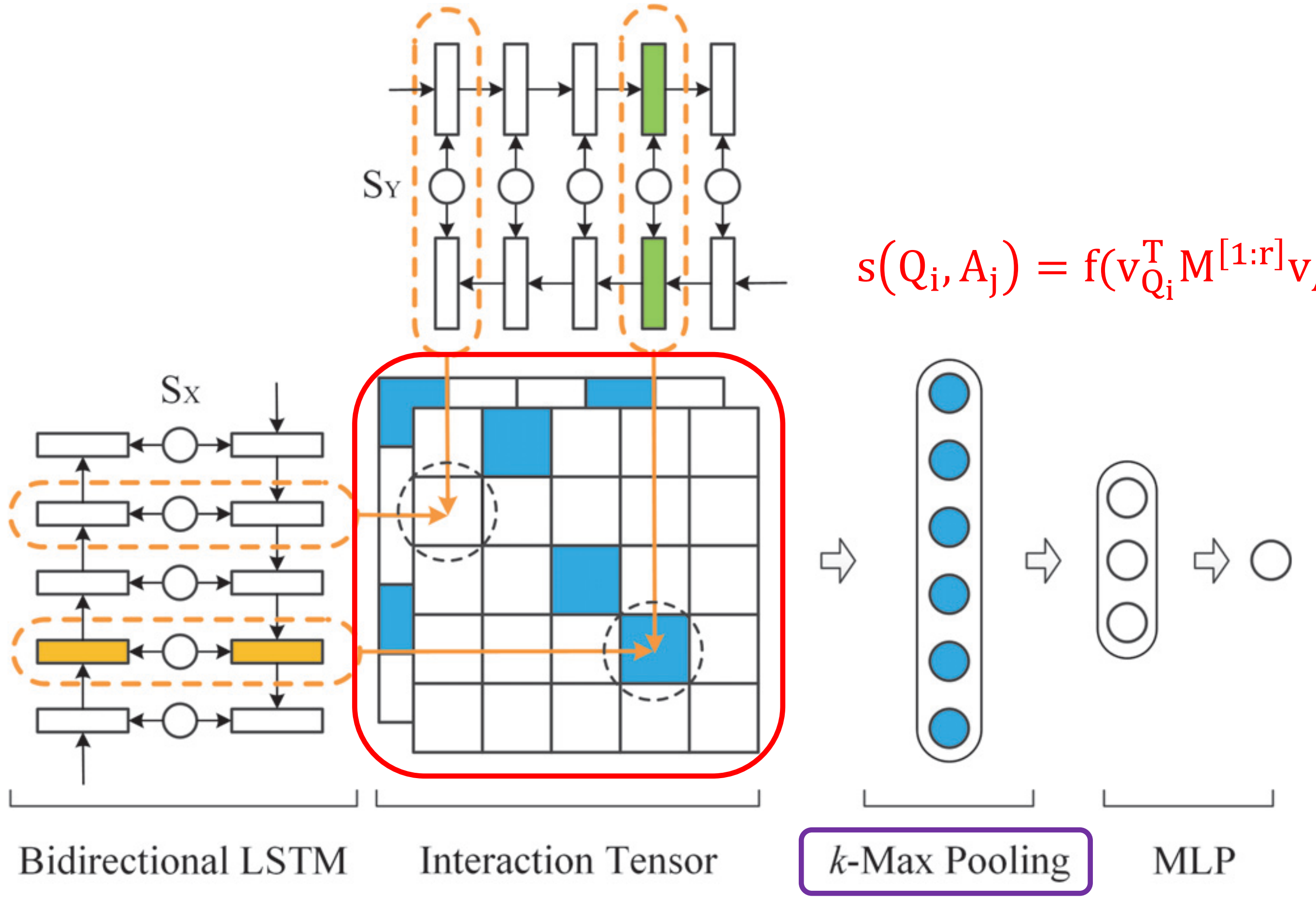
RNN + CNN

(Tan et al., 2015)



RNN with Tensor

(Wan et al., 2016)



$$s(Q_i, A_j) = f(v_{Q_i}^T M^{[1:r]} v_{A_j} + V \begin{bmatrix} v_{Q_i} \\ v_{A_j} \end{bmatrix} + b)$$

For the interaction tensor, the top k values of each slice of the tensor are returned to form a vector. Then, these vectors are further concatenated to a single vector for MLP.

Evaluation Results

- Dataset
  - WikiQA
- Metric
  - MAP, MRR, P@1

	WikiQA		
	Train	Dev	Test
# Question	2,118	296	633
# Sentence	20,360	2,733	6,165
# Answer Sentence	1,040	140	293
Ave. S / Q	9.61	9.23	9.74
# Question (Have Answer)	873	126	243
	41.22%	42.57%	38.39%
Ave. Q Length	7.16	7.23	7.26
Ave. S Length	25.29	24.59	24.95

Type	Method		MAP	MRR
CNN	bi-gram CNN	Yang et al., 2015	0.6520	0.6652
	CNNr	Severyn and Moschitti., 2016	0.6951	0.7107
	Attentive pooling CNN	Santos et al., 2016	0.6886	0.6957
	Attention-based CNN	Yin et al., 2016	0.6914	0.7127
	DocChat	Yan et al., 2016	0.7008	0.7222
	CNN + Features	Tymoshenko et al., 2016	**0.7417**	**0.7588**
RNN	LSTM+Attention	Miao et al., 2016	0.6855	0.7041
	NASM	Miao et al., 2016	0.6886	0.7069
	IARNN-Occam (context)	Wang et al., 2016 (a)	0.7341	0.7418
CNN+RNN	conv-RNN	Wang et al., 2017	**0.7427**	**0.7504**
	RNN+CNN with Mult Interaction	Wang and Jiang., 2016	**0.7433**	**0.7545**
Other	Key-Value Memory Network	Miller et al., 2016	0.7069	0.7265
	Pairwise Rank	Rao et al., 2016	0.7010	0.7180
	L.D.C	Wang et al., 2016 (b)	0.7058	0.7226
	CubeCNN (Pairwise Rank)	He and Lin., 2016	0.7090	0.7234

QG and its Interaction with QA

Question Generation (QG)

- Input
  - A piece of content D
  - Answer $\mathrm{A} \in \mathrm{D}$
- Output
  - Question Q, which can be answered by A based on D
- Motivation
  - A well-defined text generation task
  - QA research has accumulated a large number of QA pairs
  - Generated QA pairs for community QA engine
  - Generated QA pairs for QA model training
  - Generated QA pairs for QA data annotation
  - …

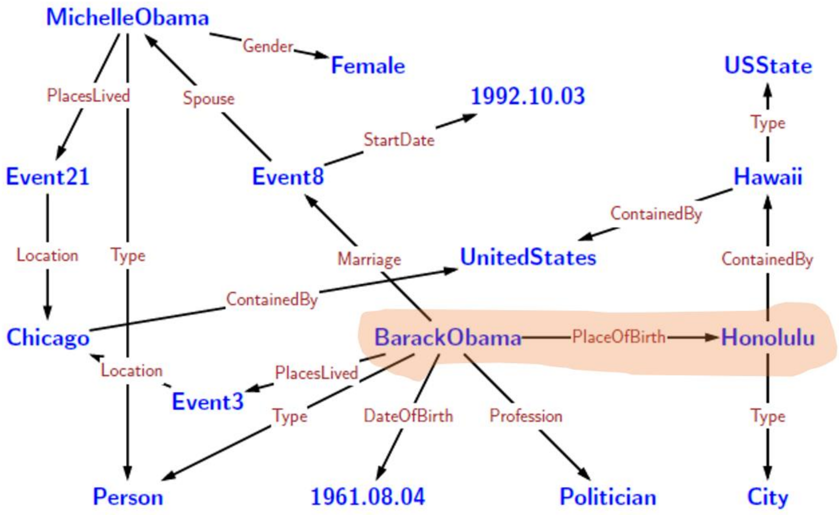
Two Types of QG Tasks

Flappy Bird is a 2013 mobile game, developed by Vietnam-based developer Dong Nguyen and published by GEARS Studios, a small, independent game developer also based in Vietnam.

Passage-based QG (PBQG)

Question: When was Flappy Bird released?

Answer: **2013**



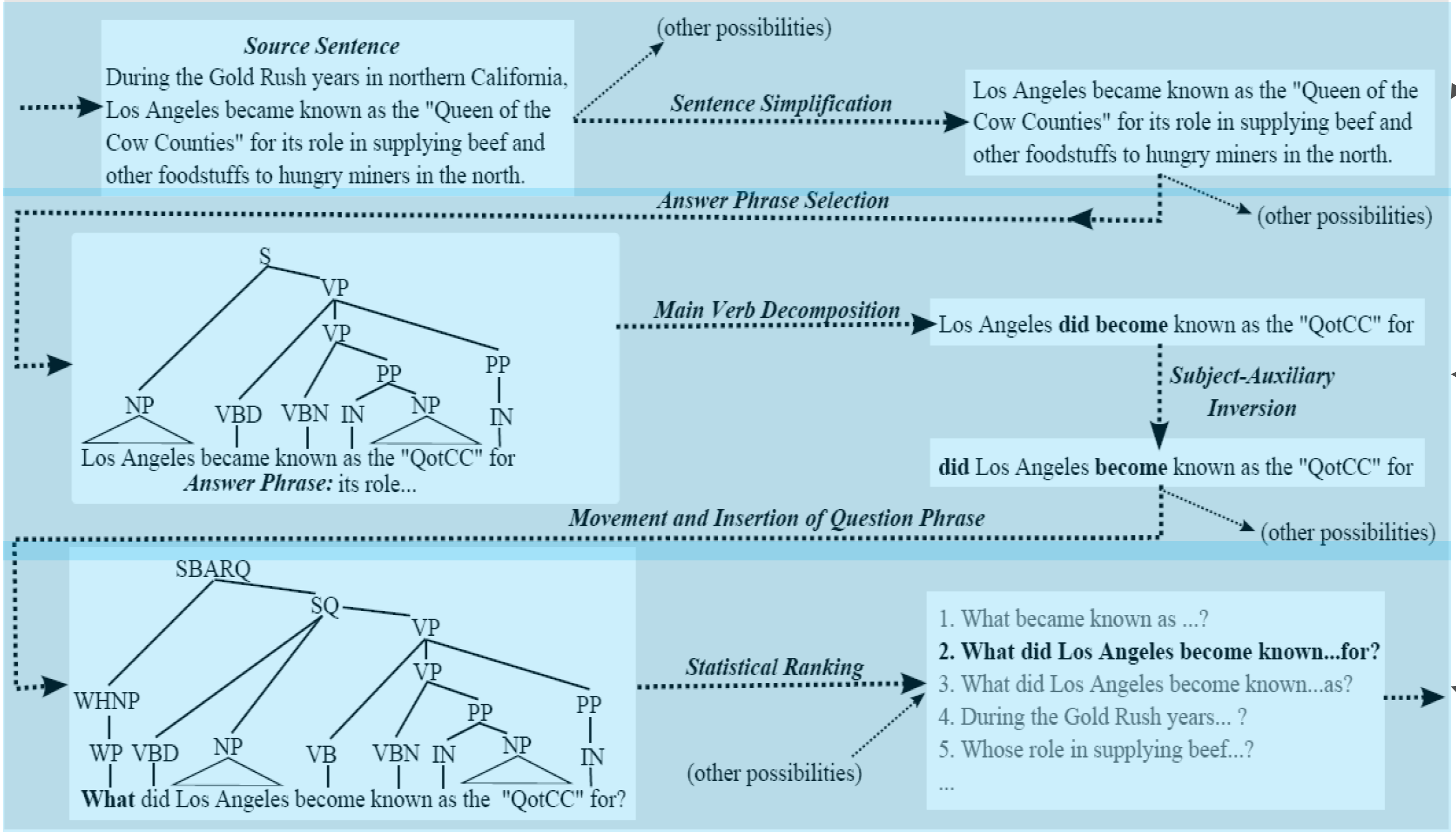
Knowledge-based QG (KBQG)

Question: Where was Barack Obama born?

Answer: **Honolulu**

Early Work of PBQG using Overgenerate-and-Rank

(Heilman and Smith, 2009)



Sentence Simplification

simplify input sentence by removing phrases with predefined types

Question Transformation

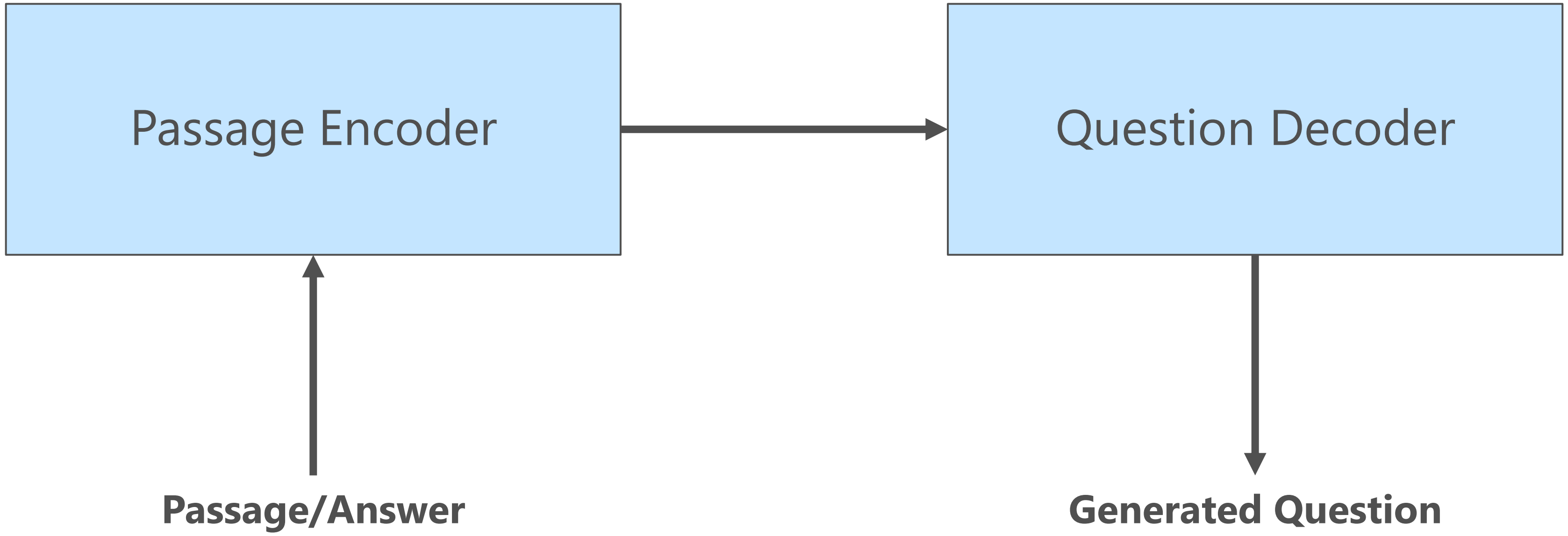
1. select a noun phrase as answer and remove it
2. decompose the main verb and invert the subject and auxiliary verb
3. insert one of the possible question phrases (who, when, where, what, which, why, and how)

Question Ranking

rank generated questions using a set of features combined by logistic regression

PBQG using Neural Networks

- Sequence-to-Sequence Model



PBQG Dataset

- Dataset: **SQuAD** (Rajpurkar et al., 2016)
  - The passage is considered as content D
  - The answer span is considered as answer A
  - 86,636 <passage, answer, question> triples are used as train set
  - 1,001 <passage, answer, question> triples are used as dev set
  - 8,965 <passage, answer, question> triples are used as test set

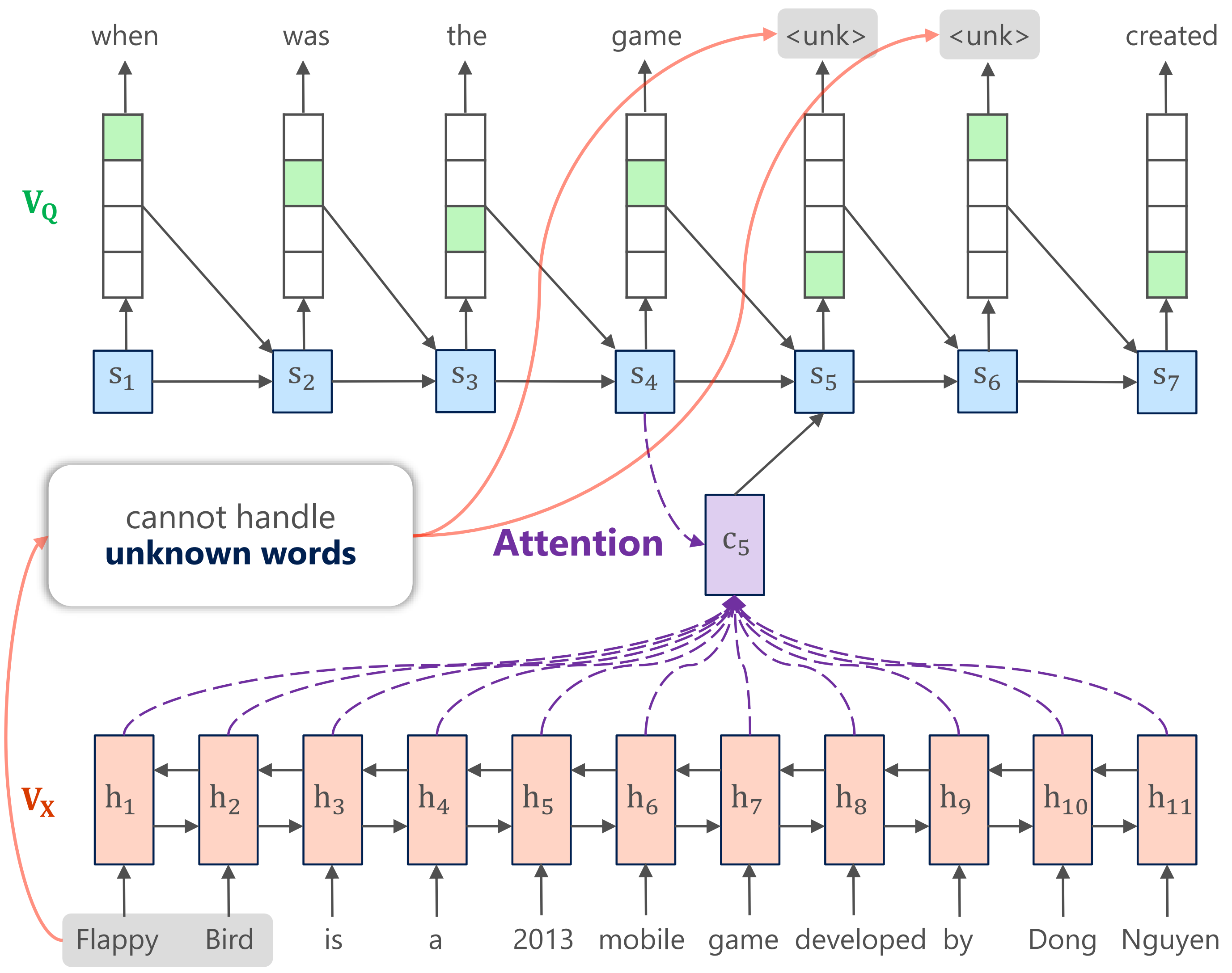
- Data examples (format: passage ➔ labeled question)

The original theatre royal in Newcastle was opened on **21 January 1788** and was located on Mosley street .
➔ When did the theater in Newcastle originally open ?

Peyton Manning became the first quarterback ever to lead **two** different teams to multiple super bowls .
➔ How many teams has Manning played for that reached the super bowl , while he was on their team ?

On 7 January 1943 , at the age of **86** , Tesla died alone in room 3327 of the New Yorker hotel .
➔ How old was Tesla when he died ?

Seq-to-Seq based PBQG
(Duan et al., 2017)



y_i

$$p(y_i|s_i, y_{i-1}, c_i) = \frac{\exp(y_i^T W_o t_i)}{\sum_{y_k \in V_Q} \exp(y_k^T W_o t_i)}$$

$$t_i = \text{maxout}(U_o s_i + V_o y_{i-1} + C_o c_i)$$

s_i

$$r_i = \sigma(W_r y_{i-1} + U_r s_{i-1} + C_r c_i)$$

$$z_i = \sigma(W_z y_{i-1} + U_z s_{i-1} + C_z c_i)$$

$$\tilde{s}_i = \tanh(W y_{i-1} + U[r_i \circ s_{i-1}] + C c_i)$$

$$s_i = z_i \tilde{s}_i + (1 - z_i) s_{i-1}$$

c_i

$$e_{ij} = v_a^T \tanh(W_a s_{i-1} + U_a h_j)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{|X|} \exp(e_{ik})}$$

$$c_i = \sum_{j=1}^{|X|} \alpha_{ij} h_j$$

h_j

$$r_j = \sigma(W_r x_j + U_r h_{j-1})$$

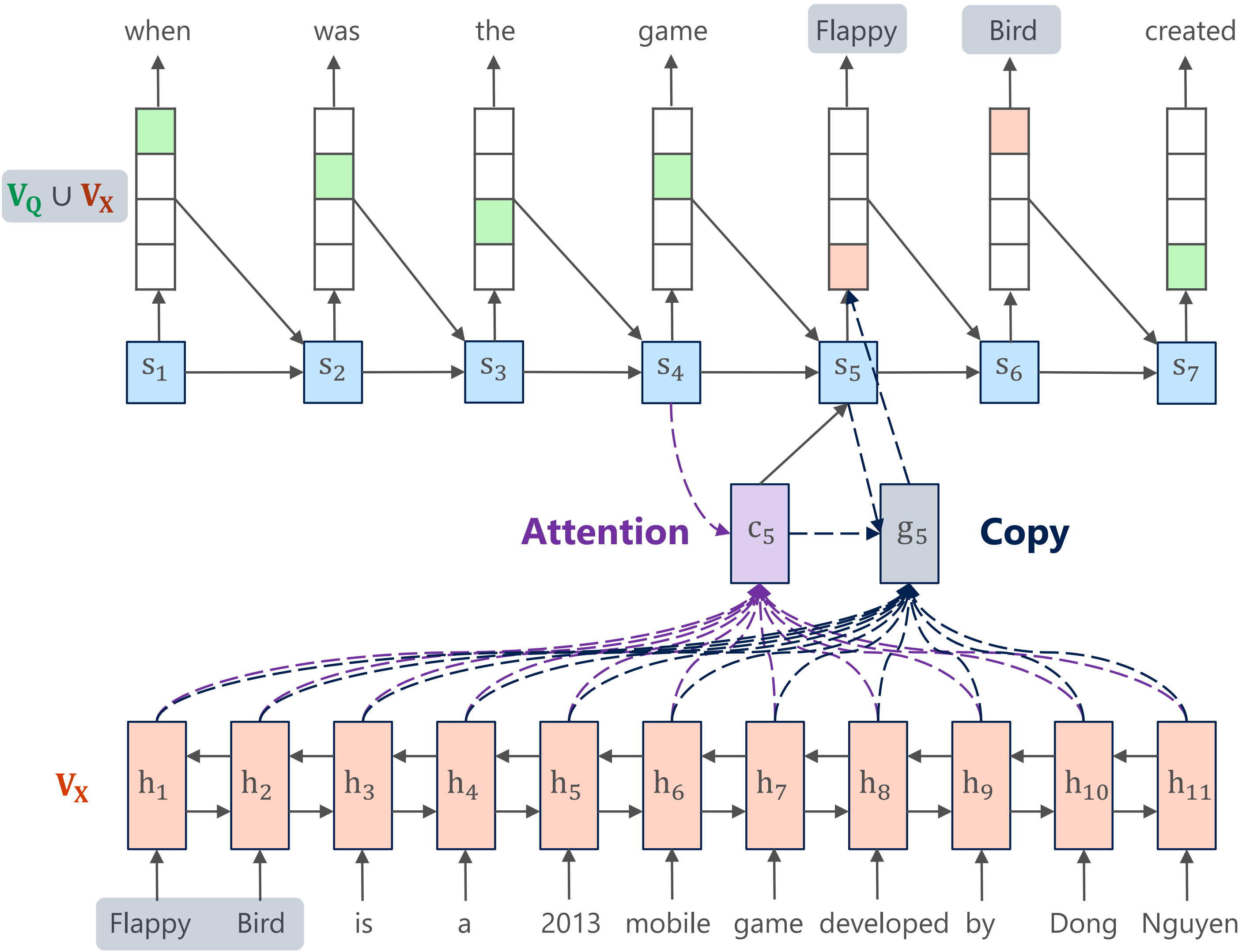
$$z_j = \sigma(W_z x_j + U_z h_{j-1})$$

$$\tilde{h}_j = \tanh\big(W x_j + U[r_j \circ h_{j-1}]\big)$$

$$h_j = z_j \tilde{h}_j + (1 - z_j) h_{j-1}$$

Seq-to-Seq based PBQG with Copy

(Duan et al., 2017)



y_i

$$p_{combine}(y_i|s_i, y_{i-1}, c_i, X) = g_i p(y_i|s_i, y_{i-1}, c_i) + (1 - g_i) p_{copy}(y_i|X)$$

$$p_{copy}(y_i = x_j|X) = \frac{\alpha_{ij}}{\sum_{t \in \{1,\ldots,|X|\}} \alpha_{it}}$$

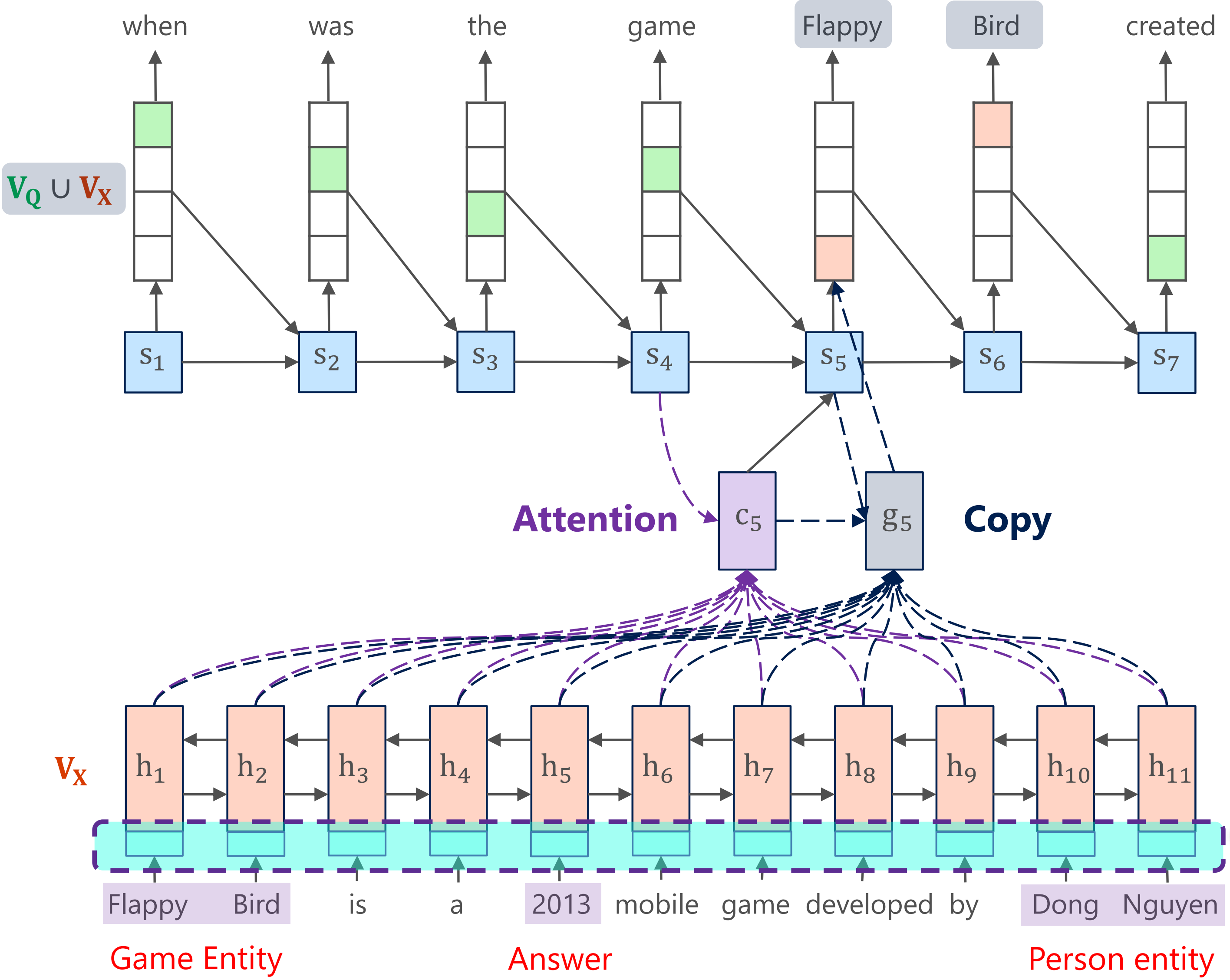
g_i

$$g_i = \mathrm{MLP}(s_i, c_i)$$

Copy mechanism allows decoder using input passage vocabulary in question generation.

Seq-to-Seq based PBQG with Copy and Prior Knowledge

(Duan et al., 2017)



y_i

$$p_{combine}(y_i|s_i, y_{i-1}, c_i, X) = g_i p(y_i|s_i, y_{i-1}, c_i) + (1 - g_i) p_{copy}(y_i|X)$$

$$p_{copy}(y_i = x_j|X) = \frac{\alpha_{ij}}{\sum_{t \in \{1,\ldots,|X|\}} \alpha_{it}}$$

g_i

$$g_i = \mathrm{MLP}(s_i, c_i)$$

Add more knowledge in passage encoding

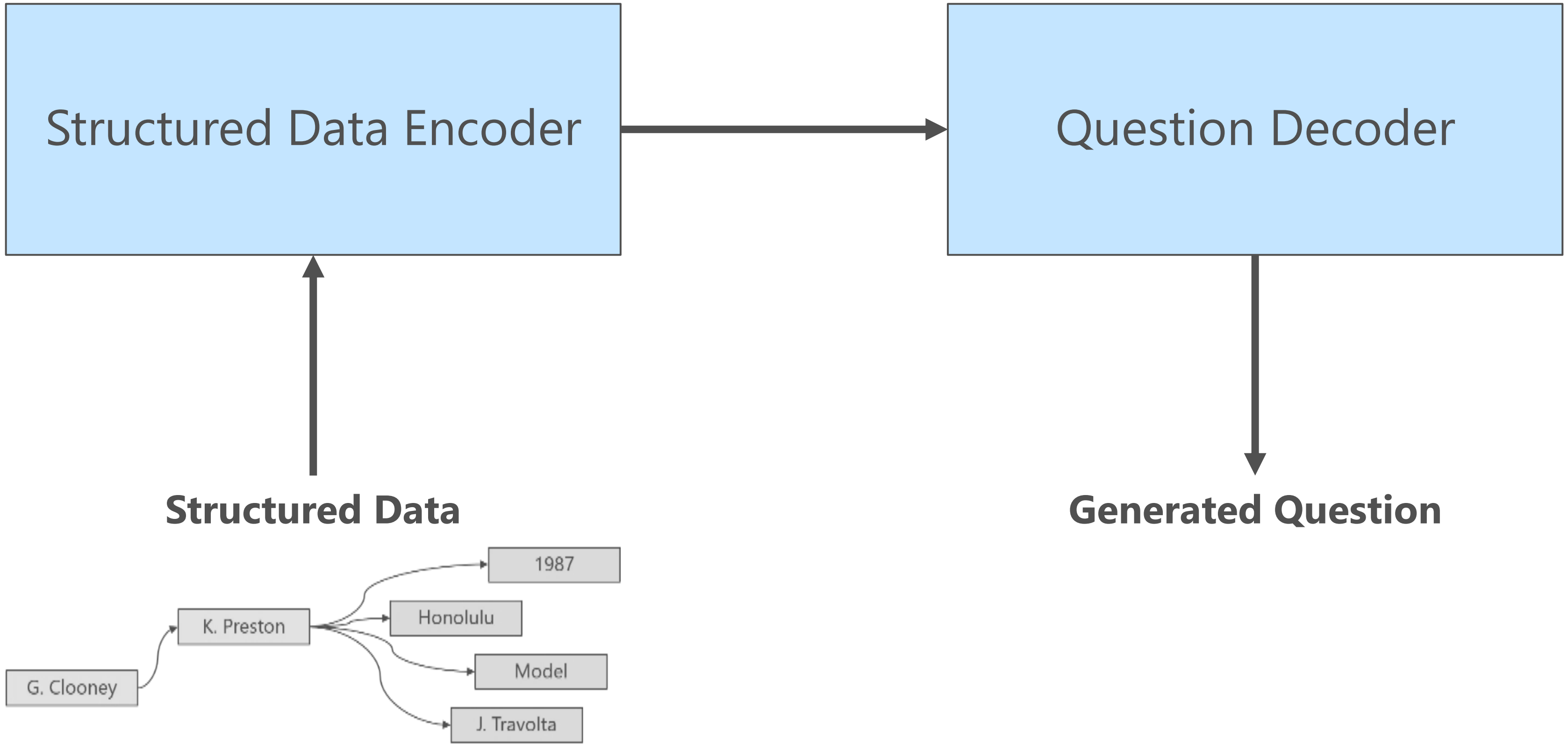
- Answer position
- Entities occurred in passage
- Syntactic types (NP, VP) of passage spans
- …

An Input-Output Example

- Input passage and answer
  - After the death of Tugh Temür in **1332** and subsequent death of Rinchinbal (emperor Ningzong) the same year , the 13-year-old Toghun Temür (emperor Huizong) , the last of the nine successors of Kublai Khan , was summoned back from Guangxi and succeeded to the throne .
- Labeled question
  - when did Tugh Temür die ?
- Generated questions (5-best)
  - **beam 0:** in what year did Tugh Temür die ?
  - **beam 1:** in what year did the death of Rinchinbal die ?
  - **beam 2:** in what year did the death of Tugh die ?
  - **beam 3:** in what year did the death of Rinchinbal Temür occur ?
  - **beam 4:** in what year did the death of Tugh Temür occur ?
- BLEU-4 score on SQuAD dataset: **12.99**

KBQG using Neural Networks

- Structured Data-to-Sequence Model



KBQG Dataset

- Dataset: **SimpleQuestions** (Bordes et al., 2015)
  - The KB triple is considered as content D
  - The object of each KB triple is considered as answer A
  - 75,910 <KB triple, question> pairs are used as train set
  - 10,845 <KB triple, question> pairs are used as dev set
  - 20,687 <KB triple, question> pairs are used as test set

- Data examples (format: KB triple ➔ labeled question)

<Andy Lippincott, Character_Created_By, Garry Trudeau>
➔ What American cartoonist is the creator of Andy Lippincott?

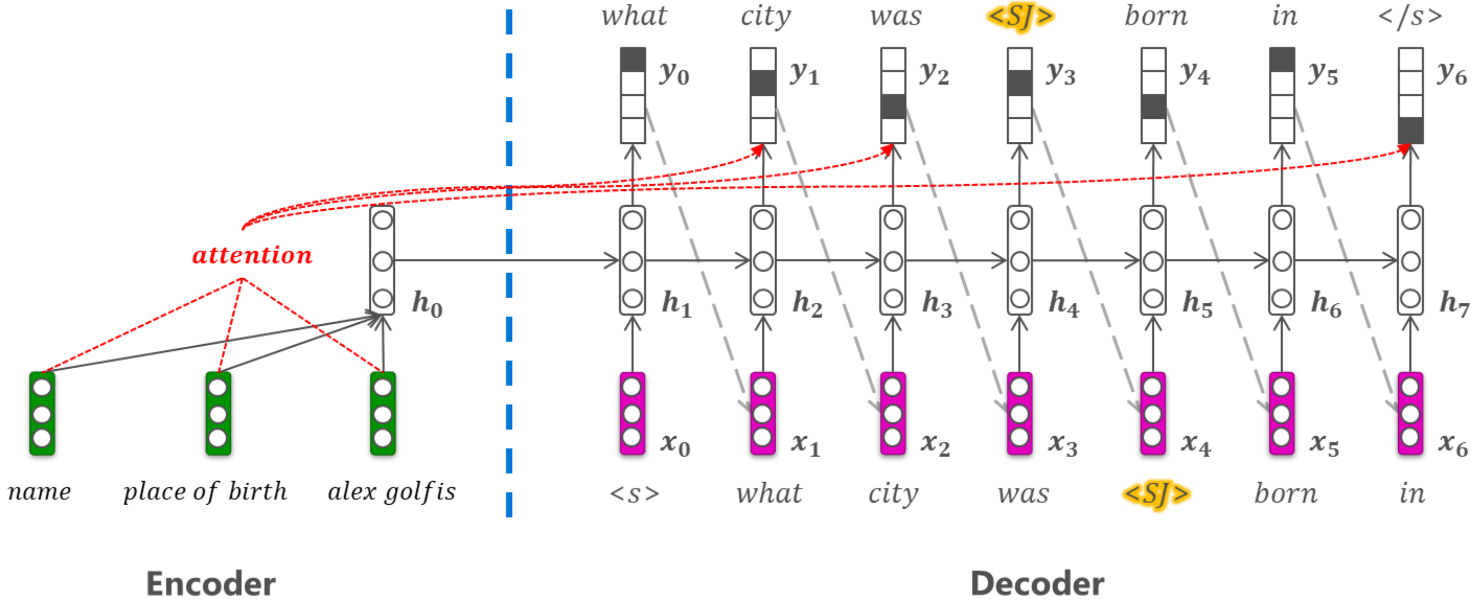
<Fires Creek, Contained_By, Nantahala National Forest>
➔ Which forest is Fires Creek in?

<Jimmy Neutron, Fictional_Character_Occupation, Inventor>
➔ What does Jimmy Neutron do?

Structured Data-to-Seq based KBQG

(Bao et al., 2017; Serban et al., 2016)

- <SJ> is a placeholder, which will be replaced by the subject entity
- Entity and Predicate embeddings are trained based on TransE (Bordes et al., 2013)



<Alex Golfis, Place_of_Birth, Athens>

KBQG

Ground Truth	what city was Alex Golfis born in
Beam 1	where was Alex Golfis born
Beam 2	what is Alex Golfis 's place of birth ?
Beam 3	where was Alex Golfis born ?
Beam 4	which city is Alex Golfis from
Beam 5	what was Alex Golfis 's place of birth ?

An Input-Output Example

- Input KB triple
  - <Dana Stevens, Film.Writer.Film, Blink>
- Labeled question
  - what is a film that Dana Stevens wrote ?
- Generated questions (5-best)
  - **beam 0:** what film did Dana Stevens write
  - **beam 1:** what film did Dana Stevens write ?
  - **beam 2:** what is a film that Dana Stevens wrote ?
  - **beam 3:** what is a film that Dana Stevens wrote
  - **beam 4:** what is the name of a film that was written by Dana Stevens
- BLEU-4 score on SimpleQuestions dataset: **39.12**

Some Issues Observed

- Training data is not balanced
- Generated questions in N-best list are similar to each other
- The copied contents from passages are wrong
- The copied contents from passages are incomplete
- Input passage lacks of background knowledge for a better question generation

Can QA and QG help each other?

- Interaction via **Feature**
  - Nan Duan, Duyu Tang, Peng Chen, Ming Zhou. Question Generation for Question Answering. EMNLP, 2017.
- Interaction via **Joint Model**
  - Tong Wang, Xingdi Yuan, Adam Trischler. A Joint Model for Question Answering and Question Generation. arXiv, 2017.
- Interaction via **Dual Learning**
  - Duyu Tang, Nan Duan, Tao Qin, Ming Zhou. Question Answering and Question Generation as Dual Tasks. arXiv, 2017.
- Interaction via **GDAN**
  - Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, William W. Cohen. Semi-Supervised QA with Generative Domain-Adaptive Nets. ACL, 2017.

QA Tasks and Datasets

English QA Datasets

- KBQA
  - **WebQuestions** (Stanford)
    - https://nlp.stanford.edu/software/sempre/
  - **SimpleQuestions** (Facebook)
    - https://research.fb.com/downloads/babi/
- TableQA
  - **WikiTableQuestions** (Stanford)
    - https://nlp.stanford.edu/blog/wikitablequestions-a-complex-real-world-question-understanding-dataset/
- PassageQA
  - **WikiQA** (Microsoft Research)
    - https://www.microsoft.com/en-us/research/publication/wikiqa-a-challenge-dataset-for-open-domain-question-answering/
- CommunityQA
  - **Question Pairs** (Quora)
    - https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
  - **Task 3: Community QA** (SemEval)
    - http://alt.qcri.org/semeval2017/index.php?id=tasks
- Machine Reading Comprehension
  - **SQuAD** (Stanford)
    - https://rajpurkar.github.io/SQuAD-explorer/
  - **MS MARCO** (Microsoft)
    - http://www.msmarco.org/
  - **CNN/Daily Mail** (DeepMind)
    - http://cs.nyu.edu/~kcho/DMQA/

Chinese QA Datasets

- KBQA
  - **NLPCC2016-KBQA** (NLPCC & Microsoft Research Asia)
    - http://tcci.ccf.org.cn/conference/2016/pages/page05_evadata.html
  - **NLPCC2017-KBQA** (NLPCC & Microsoft Research Asia)
    - http://tcci.ccf.org.cn/conference/2017/taskdata.php
- PassageQA
  - **NLPCC2016-DBQA** (NLPCC & Microsoft Research Asia)
    - http://tcci.ccf.org.cn/conference/2016/pages/page05_evadata.html
  - **NLPCC2017-DBQA** (NLPCC & Microsoft Research Asia)
    - http://tcci.ccf.org.cn/conference/2017/taskdata.php

Chinese Open Domain QA Tasks in NLPCC

- **Knowledge-Based Question Answering (KBQA)** task

- An example
  - <question id=28> 新版还珠格格的导演是谁
  - <answer id=28> 李平，丁仰国

	# of <Question, Triple, Answer> Triples
Train set	14,609
Test set	9,870

	# of <Subject, Predicate, Object> Triples
Chinese KB	47,943,429

```
新还珠格格 ||| entity.primaryName ||| 新还珠格格
新还珠格格 ||| 中文名 ||| 新还珠格格
新还珠格格 ||| 外文名 ||| New my fair Princess
新还珠格格 ||| 出品时间 ||| 2011年和2014年
新还珠格格 ||| 出品公司 ||| 上海创翊文化传播有限公司
新还珠格格 ||| 制片地区 ||| 中国大陆，中国台湾
新还珠格格 ||| 拍摄地点 ||| 横店影视城
新还珠格格 ||| 发行公司 ||| 上海创翊文化传播有限公司
新还珠格格 ||| 首播时间 ||| 2011年7月16日
新还珠格格 ||| 导演 ||| 李平，丁仰国
新还珠格格 ||| 编剧 ||| 琼瑶，黄素媛
新还珠格格 ||| 主演 ||| 李晟，海陆，张睿，李佳航，潘杰明，赵丽颖，邱心志，邓萃雯，刘雪华
新还珠格格 ||| 集数 ||| 总共98集→第一部1至37集→第二部37至74集→第三部74至98集
新还珠格格 ||| 每集长度 ||| 前三部：45分钟 第四部：48分钟
新还珠格格 ||| 类型 ||| 古装，爱情，励志，喜剧
新还珠格格 ||| 上映时间 ||| 前三部：2011年07月16日至2011年9月8日第四部：2016年暑期档
新还珠格格 ||| 在线播放平台 ||| 芒果TV,PPTV,暴风影音，优酷，搜狐。
新还珠格格 ||| 总策划 ||| 杨文红，苏晓
新还珠格格 ||| 出品人 ||| 欧阳常林
新还珠格格 ||| 总监制 ||| 魏文彬
新还珠格格 ||| entity.description ||| 《新还珠格格》翻拍自琼瑶经典之作《还珠格格》，由李晟、
```

- Labeling guideline
  1. Show a KB triple to a human annotator;
  2. Let the human annotator to ask a question about this KB triple, whose answer should be the object of the triple.

Chinese Open Domain QA Tasks in NLPCC

- **D**ocument-**B**ased **Q**uestion **A**nswering (**DBQA**) task

- An example
  - 首尔地铁一共有多少个车站?
    - 首尔地铁四通八达，目前共有8条线，全长约290多公里，位居世界第4位。 0
    - 年运送乘客数为 22亿名，位居全球第3位。 0
    - 第9号线于2001年开工，预计到2007年完工。 0
    - 首尔地铁共有260多个车站,每站都有很多出口。 **1**
    - 几乎所有的地铁站周边都是社区的中心点， 出了检票口不远即可抵达办公的地点、购物的场所、休闲的去处、就医的地点等。 0
    - 运行时间从早晨5点开始到凌晨1点。 0
    - 为使每一条地铁线路都能够让乘客一眼就辨认出来，采用了最简易的办法——颜色区别法。 0
    - 例如，地铁1号线为红色，2号线为绿色，3号线为橘黄色等。 0

	# of <Question, Document, Answer> Triples
Train set	8,772
Test set	5,779

- Labeling guideline
  1. Show a document to a human annotator;
  2. Let the human annotator to select a sentence from the document;
  3. Let the human annotator to ask a question, whose answer should the selected sentence.

Final Evaluation Results (NLPCC 2016)

KBQA Submissions	Rank (by F1 Score)
北京大学	1
国防科学技术大学	2
华中师范大学	3
哈尔滨工业大学 (HIT-SCIR)	4
东北大学 (自然语言处理实验室)	5
Harbin ShenZhi Technology Co., Ltd.	6
哈尔滨工业大学 (机器智能与翻译实验室)	7
西南交通大学 (信息科学与技术学院)	8
中科院自动化所 (CBrain Team)	9
郑州大学 (自然语言处理实验室)	10
华东理工大学 (NLP and Big Data Lab)	11
北京航空航天大学	12
山西大学	13
北京理工大学 (计算机学院)	14
浙江大学	15
同济大学 & Tokushima University	16
大连理工大学	17
Late Submissions	
哈尔滨工业大学 (ITNLP Group)	-
复旦大学 (计算机学院)	-
北京理工大学 (自动化学院)	-
重庆理工大学 (计算机学院)	-

DBQA Submissions	Rank (by MRR)
复旦大学	1
中科院自动化所 (CBrain Team)	2
哈尔滨工业大学 (机器智能与翻译实验室) [Secondary]	3
哈尔滨工业大学 (机器智能与翻译实验室) [Primary]	4
天津大学	5
黑龙江工程学院 [MART]	6
哈尔滨工业大学 (ITNLP Group)	7
黑龙江工程学院 [CA]	8
哈尔滨工业大学 (HIT-SCIR)	9
黑龙江工程学院 [LM]	10
北京航空航天大学	11
山西大学	12
大连理工大学	13
同济大学	14
东北大学	15
武汉科技大学 (NLP@WUST)	16
Harbin ShenZhi Technology Co., Ltd.	17
浙江大学	18

Final Evaluation Results (NLPCC 2017)

KBQA Submissions	Rank (by F1 Score)
北京大学.name_system	1
东北大学	2
北京大学.ICL	3
浙江大学.TeamTCM	4
华中师范大学.CCNU-NLP-Blaze	5
北京大学.IASO	6
华东师范大学	7
复旦大学.XiaoCuiQA	8
DeepIntell.primary	9
DeepIntell.run1	10
DeepIntell.run2	11
东南大学.MotorQASystem	12
北京大学.strawberry	13
重庆理工大学.CQUT_AC996	14

DBQA Submissions	Rank (by MRR)
复旦大学	1
北京邮电大学	2
同济大学	3
DeepIntell-1	4
天津大学	5
DeepIntell-2	6
DeepIntell-3	7
DeepIntell-4	8
华中师范大学	9
北京大学	10
浙江大学	11
网龙网络有限公司	12
北京航空航天大学	13
国防科技大学	14
华东师范大学	15
北京联合大学	16
北京大学	17
大连理工大学-1	18
大连理工大学-2	19
大连理工大学-3	20
重庆理工大学	21

More Chinese/English QA tasks and datasets in the future! ☺

Summary, Latest Trends and Future Directions

Summary

- **InfoBot**
  - Crucial to conversational bot and search engine
  - Focus on single-turn QA/QG currently, and will extend to multi-turn QA/QG next
- **QA based on Structured Data**
  - Several labeled datasets released, but biased on factoid and one-hop answers
  - Answer as weak supervision + Deep learning methods dominate
  - The size of labeled data is limited
- **QA based on Unstructured Data**
  - Several labeled datasets released
  - Deep learning methods dominate
  - Query understanding and reason capabilities are weak
  - The size of labeled data is limited
- **QG and its Interaction with QA**
  - Training data is not balanced
  - Generated questions in N-best list are similar to each other
  - The copied contents from passages are wrong or incomplete
  - Input passage lacks of background knowledge for a better question generation
  - The improvement of QG and QA co-training is limited

Latest Trends and Future Directions

- **Large Scale QA Dataset**
  - Human-labeled
  - Machine-generated
- **Method**
  - Deep Learning
  - Reinforcement Learning
  - Unsupervised Learning (e.g., Dual Learning and GAN)
  - Transfer Learning
  - …
- **Task**
  - Semantic Parsing
  - QA/QG
  - Reading Comprehension
  - Multimodal QA
  - Interactive QA
  - Reasoning QA
  - Open IE QA
  - …

Machine Reading Comprehension (MRC)

SQuAD: 100,000+ Questions for Machine Comprehension of Text

Pranav Rajpurkar and **Jian Zhang** and **Konstantin Lopyrev** and **Percy Liang**
{pranavsr,zjian,klopyrev,pliang}@cs.stanford.edu
Computer Science Department
Stanford University

Abstract

We present the Stanford Question Answering Dataset (SQuAD), a new reading comprehension dataset consisting of 100,000+ questions posed by crowdworkers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage. We analyze the dataset to understand the types of reasoning required to answer the questions, leaning heavily on dependency and constituency trees. We build a strong logistic regression model, which achieves an F1 score of 51.0%, a significant improvement over a simple baseline (20%). However, human performance (86.8%) is much higher, indicating that the dataset presents a good challenge problem for future research. The dataset is freely available at https://stanford-qa.com.

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

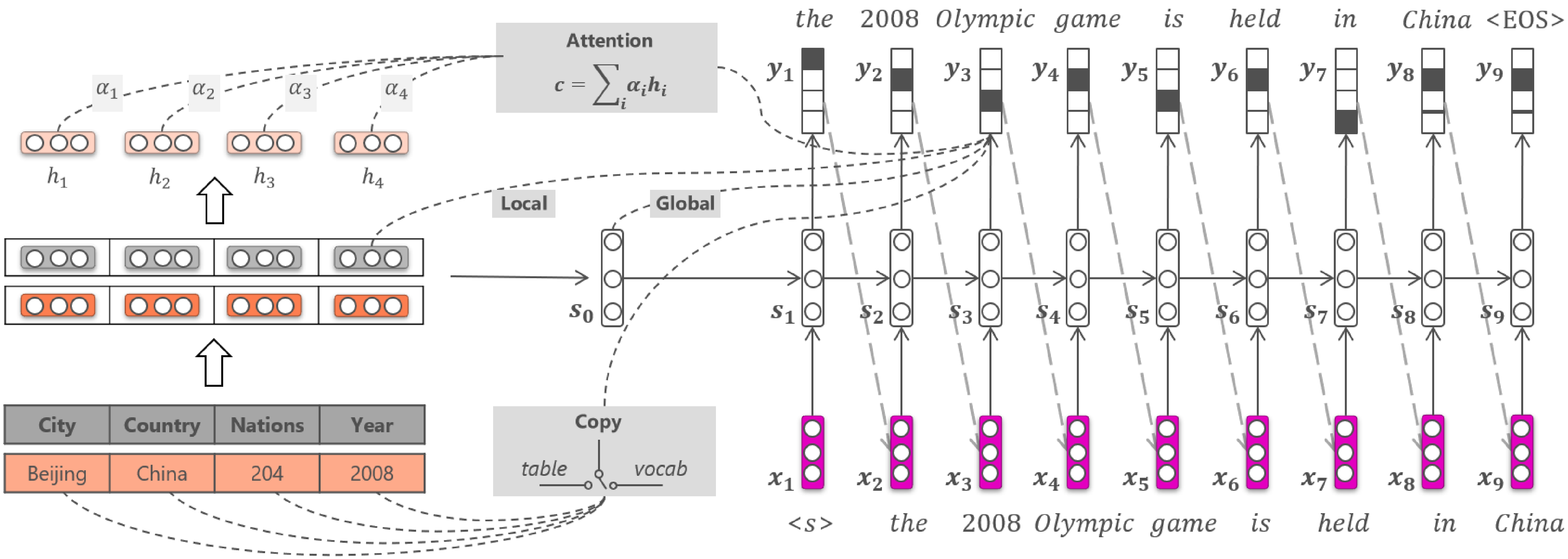
Figure 1: Question-answer pairs for a sample passage in the

Leaderboard

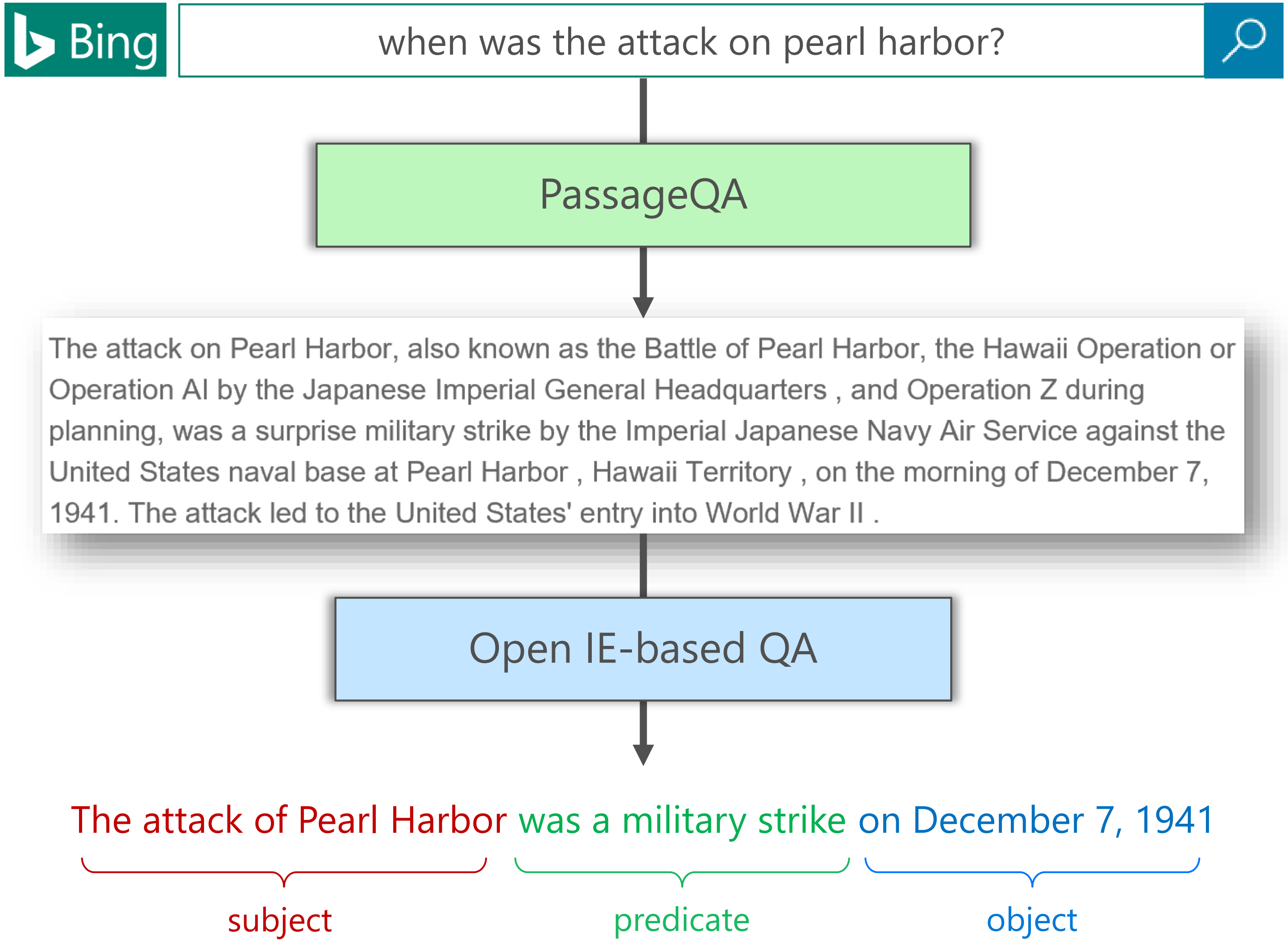
Since the release of our dataset, the community has made rapid progress! Here are the ExactMatch (EM) and F1 scores of the best models evaluated on the test and development sets of v1.1. Will your model outperform humans on the QA task?

Rank	Model	EM	F1
1 Sep 20, 2017	AIR-FusionNet (ensemble) *Microsoft Business AI Solutions Team*	78.842	85.936
2 Aug 16, 2017	DCN+ (ensemble) *Salesforce Research*	78.706	85.619
3 Jul 25, 2017	Interactive AoA Reader (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	77.845	85.297
3 Sep 01, 2017	r-net (ensemble) *Microsoft Research Asia* http://aka.ms/rnet	78.244	85.206
4 Aug 21, 2017	Reinforced Mnemonic Reader (ensemble) *NUDT and Fudan University* https://arxiv.org/abs/1705.02798	77.678	84.888
5 Sep 08, 2017	AIR-FusionNet (single model) *Microsoft Business AI Solutions team*	75.968	83.900
6 Jul 17, 2017	r-net (single model) *Microsoft Research Asia* http://aka.ms/rnet	75.705	83.496

Generative QA (accepted by AAAI 2018)

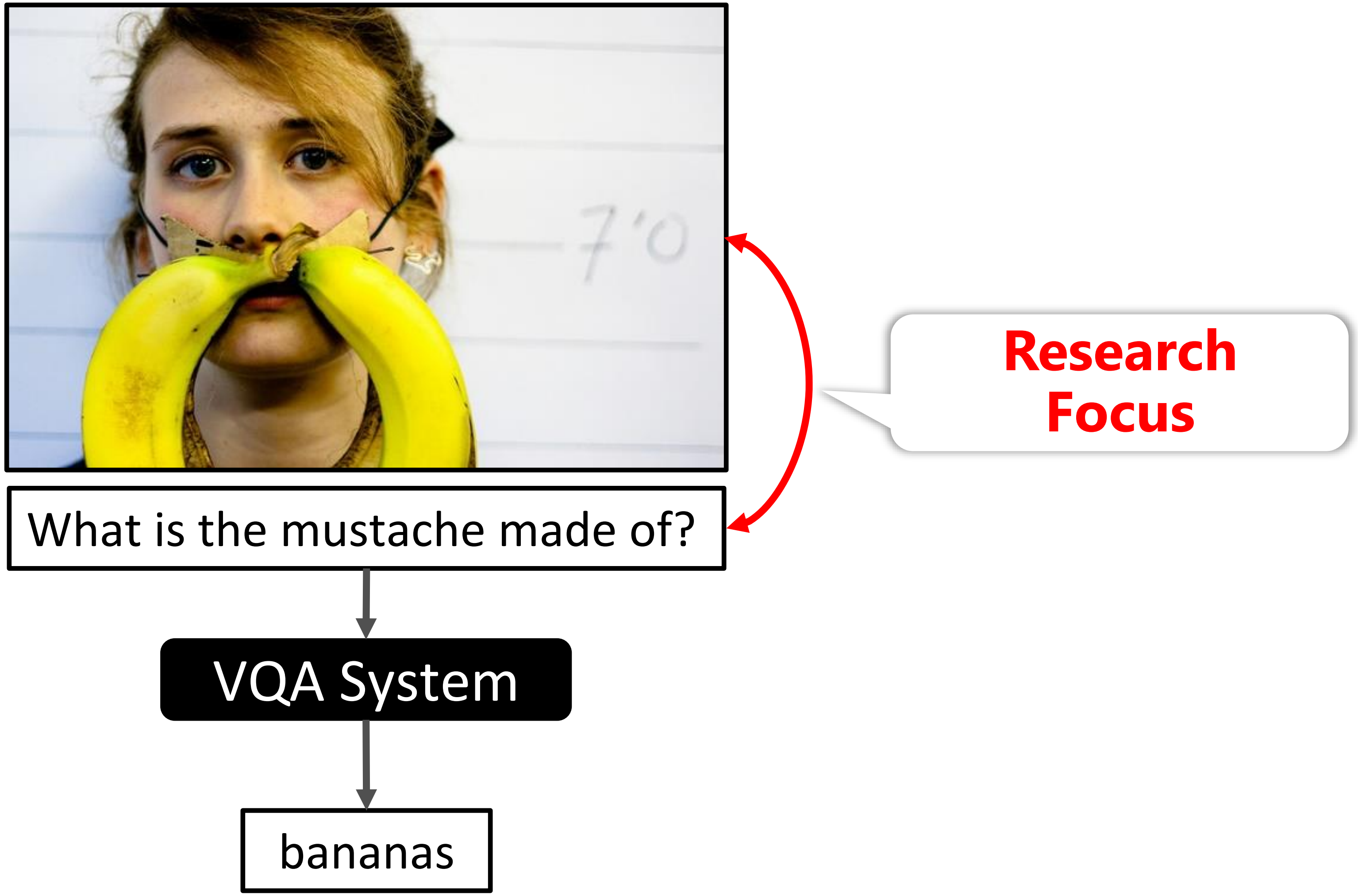


Open IE QA (accepted by AAAI 2018)



Visual QA

- Input
  - Image
  - Question
- Image encoding
  - Convolutional neural network (CNN)
- Question encoding
  - Recurrent neural network (RNN)
- **Fusion of image and question**
  - Concatenation
  - Attention
  - Tensor
  - …
- Output
  - 1-of-K most common answers



Interaction between QA and QG via Dual Learning/GAN/…

Question Answering

$$P(Q,A) = \underbrace{\overbrace{P(Q)}^{\text{Language model}}\;P(A|Q)}_{\text{QA model}} = \underbrace{\overbrace{P(A)}^{\text{Language model}}\;P(Q|A)}_{\text{QG model}}$$

$$\boldsymbol{l_{dual} = (\log P(A) + \log P(Q|A) - \log P(Q) - \log P(A|Q))^2}$$

Question Generation

Update QA model: $\theta_{QA} = sgd(\theta_{QA}, \nabla_{\theta_{QA}} l_{QA} + \boldsymbol{\lambda_{QA} \nabla_{\theta_{QA}} l_{dual}})$

Update QG model: $\theta_{QG} = sgd(\theta_{QG}, \nabla_{\theta_{QG}} l_{QG} + \boldsymbol{\lambda_{QG} \nabla_{\theta_{QG}} l_{dual}})$

Semantic Parsing-based TBQA with Neural Networks

Question: how many CFL teams are from York College?

$$\alpha_t^{inp} = W^{inp} h_t^{enc}$$

$$\beta^{inp} = \mathrm{softmax}(\alpha^{inp})$$

$$\kappa^{agg} = \sum_t \beta_t^{inp} h_t^{enc}$$

$$\alpha^{agg} = W^{agg} \tanh(V^{agg}\kappa^{agg} + b^{agg}) + c^{agg}$$

Operator Selection

COUNT **MAX** **MIN** **...** **NULL**

Column Selection

$$\alpha_j^{sel} = W^{sel} \tanh\left(V^{sel}\kappa^{sel} + V^c e_j^c\right)$$

Pick #	CFL Team	Player	Position	College
27	Hamilton Tiger-Cats	Connor Healy	DB	Wilfrid Laurier
28	Calgary Stampeders	Anthony Forgone	OL	York
29	Ottawa Renegades	L.P. Ladouceur	DT	California
30	Toronto Argonauts	Frank Hoffman	DL	York
...	...	...	...	...

Pointer Network

$$\alpha_{s,t}^{ptr} = W^{ptr} \tanh(U^{ptr} g_s + V^{ptr} h_t)$$

```
SELECT COUNT
CFL Team FROM CFLDraft
WHERE College = "York"
```

Interactive QA

q_1 = "how tall is Kobe Bryant"

"6' 6""

q_2 = "when and where was he born"

$he = "Kobe Bryant"

"Aug. 23, 1978, Philadelphia, PA"

q_3 = "who was the president then"

+"of united states in 1978"

"Jimmy Carter"

q_1 = "how to split string in python"

"the method split() returns …"

q_2 = "how to read file"

+"in python"

q_1 = "what is deep learning"

"a set of techniques for training neural …"

q_2 = "what is Michael Jordan working on"

+"the UC Berkeley professor"

- Anaphora/co-reference resolution
- Intent completion
- Context beyond entities
- Context from previous answers
- Explicit + Implicit context
- Intent disambiguation

Core Technologies

- Conversational Question Understanding
  - Intent clarification
  - Multi-turn/Context
  - NLU
- Conversational answer generation
  - QA/MRC
  - Machine Writing (NL answer conversion)

Domain Adaptive Model Training for QA

$$Q_S = \{Q_S^1, \ldots, Q_S^N\}$$

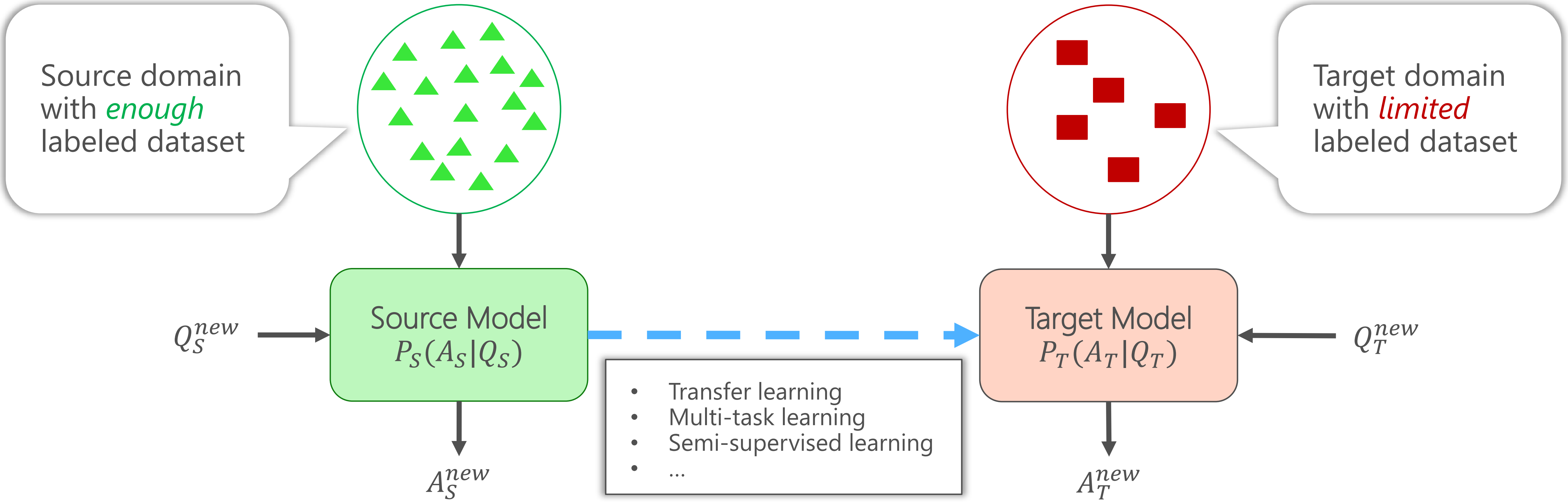
$$A_S = \{A_S^1, \ldots, A_S^N\}$$

$$D_S = \{(Q_S^1, A_S^1), \ldots, (Q_S^N, A_S^N)\}$$

$$Q_T = \{Q_T^1, \ldots, Q_T^L\}$$

$$A_T = \{A_T^1, \ldots, A_T^L\}$$

$$D_T = \{(Q_T^1, A_T^1), \ldots, (Q_T^L, A_T^L)\}$$



Microsoft

Thank You!

nanduan@microsoft.com

Reference

- Warren McCulloch and Walter Pitts. **A Logical Calculus of the Ideas Immanent in Nervous Activity**. 1943.
- Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. **Efficient Estimation of Word Representations in Vector Space**. arXiv, 2013.
- Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Honza Cernocky, Sanjeev Khudanpur. **Recurrent Neural Network based Language Model**. Interspeech, 2010.
- Sepp Hochreiter and Jurgen Schmidhuber. **Long Short-Term Memory**. NIPS, 1997.
- Luke S. Zettlemoyer and Michael Collins. **Online Learning of Relaxed CCG Grammars for Parsing to Logical Form**. EMNLP, 2007.

Reference

- Antoine Bordes, Jason Weston, Sumit Chopra. **Question Answering with Subgraph Embeddings**. EMNLP, 2014.
- Min-Chul Yang, Nan Duan, Ming Zhou, Hae Chang Rim. **Joint Relational Embeddings for Knowledge-based Question Answering**. EMNLP, 2014.
- Li Dong, Furu Wei, Ming Zhou, Ke Xu. **Question Answering over Freebase with Multi-Column Convolutional Neural Networks**. ACL, 2015.
- Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, Jason Weston. **Key-Value Memory Networks for Directly Reading Documents**. arXiv, 2016.
- Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, Rob Fergus. **End-To-End Memory Networks**. NIPS, 2015.

Reference

- Wen-tau Yih, Ming-Wei Chang, Xiaodong He, Jianfeng Gao. **Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base**. ACL, 2015.
- Jonathan Berant and Percy Liang. **Semantic Parsing via Paraphrasing**. ACL, 2014.
- Iulian Vlad Serban, Alberto Garc´ıa-Dur´an, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, Yoshua Bengio. **Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus**. ACL, 2016.
- Robin Jia and Percy Liang. **Data Recombination for Neural Semantic Parsing**. ACL, 2015.
- Yuk Wah Wong and Raymond J. Mooney. **Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus**. ACL, 2007.

Reference

- Robin Jia and Percy Liang. **Data Recombination for Neural Semantic Parsing**. ACL, 2016.
- Junwei Bao, Nan Duan, Ming Zhou, Tiejun Zhao. **Knowledge-Based Question Answering as Machine Translation**. ACL, 2014.
- Luke S. Zettlemoyer and Michael Collins. **Online Learning of Relaxed CCG Grammars for Parsing to Logical Form**. EMNLP, 2007.
- Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, Mark Steedman. **Lexical Generalization in CCG Grammar Induction for Semantic Parsing**. EMNLP, 2011.
- Siva Reddy, Mirella Lapata, Mark Steedman. **Large-scale Semantic Parsing without Question-Answer Pairs**. TACL, 2014.

Reference

- Siva Reddy, Oscar Tackstrom, Slav Petrov, Mark Steedman, Mirella Lapata. **Universal Semantic Parsing**. EMNLP, 2017.
- Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang. **Semantic Parsing on Freebase from Question-Answer Pairs**. EMNLP, 2013.
- Li Dong and Mirella Lapata. **Language to Logical Form with Neural Attention**. ACL, 2016.
- Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, Ni Lao. **Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision**. ACL, 2017.
- Xuchen Yao and Benjamin Van Durme. **Information Extraction over Structured Data: Question Answering with Freebase**. ACL, 2014.

Reference

- Victor Zhong, Caiming Xiong, Richard Socher. **Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning**. arXiv, 2017.
- Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, Xifeng Yan. **Table Cell Search for Question Answering**. WWW, 2016.
- Arvind Neelakantan, Quoc V. Le, Ilya Sutskever. **Neural Programmer: Inducing Latent Programs with Gradient Descent**. ICLR, 2016.
- Arvind Neelakantan, Quoc V. Le, Martin Abadi, Andrew McCallum, Dario Amodei. **Learning a Natural Language Interface with Neural Programmer**. ICLR, 2017.
- Zhao Yan, Duyu Tang, Nan Duan, Junwei Bao, Yuanhua Lv, Ming Zhou, Zhoujun Li. **Content-Based Table Retrieval for Web Queries**. arXiv, 2017.

Reference

- Abdessamad Echihabi and Daniel Marcu. **A Noisy-Channel Approach to Question Answering**. ACL, 2003.
- Michael Heilman and Noah A. Smith. **Good Question! Statistical Ranking for Question Generation**. NAACL, 2009.
- Xinya Du, Junru Shao, Claire Cardie. **Learning to Ask: Neural Question Generation for Reading Comprehension**. ACL, 2017.
- Nan Duan, Duyu Tang, Peng Chen, Ming Zhou. **Question Generation for Question Answering**. EMNLP, 2017.
- Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Sandeep Subramanian, Saizheng Zhang, Adam Trischler. **Machine Comprehension by Text-to-Text Neural Question Generation**. arXiv, 2017.

Reference

- Iulian Vlad Serban, Alberto Garc´ıa-Dur´an, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, Yoshua Bengio. **Generating Factoid Questions with Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus**. ACL, 2016.

- Junwei Bao, Duyu Tang, Ming Zhou. **Table-to-Text: Describing Table Region with Natural Language**. arXiv, 2017.

- Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, William W. Cohen. **Semi-Supervised QA with Generative Domain-Adaptive Nets**. ACL, 2017.

- Duyu Tang, Nan Duan, Tao Qin, Ming Zhou. **Question Answering and Question Generation as Dual Tasks**. arXiv, 2017.

- Tong Wang, Xingdi Yuan, Adam Trischler. **A Joint Model for Question Answering and Question Generation**. arXiv, 2017.

Reference

- Lei Yu, Karl Moritz Hermann, Phil Blunsom, Stephen Pulman. **Deep Learning for Answer Sentence Selection**. NIPS Deep Learning Workshop, 2014.
- Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, Gregoire Mesnil. **Learning Semantic Representations using Convolutional Neural Networks for Web Search**. WWW, 2014a.
- Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, Gregoire Mesnil. **A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval**. CIKM, 2014b.
- Aliaksei Severyn, Alessandro Moschitti. **Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks**. SIGIR, 2015.
- Cicero dos Santos, Ming Tan, Bing Xiang, Bowen Zhou. **Attentive Pooling Networks**. arXiv, 2016.

Reference

- Wenpeng Yin, Hinrich Schutze, Bing Xiang, Bowen Zhou. **ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs**. TACL, 2016.
- Di Wang and Eric Nyberg. **A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering**. ACL, 2015.
- Ming Tan, Cicero dos Santos, Bing Xiang, Bowen Zhou. **LSTM-Based Deep Learning Models for Non-factoid Answer Selection**. arXiv, 2015.
- Wei-Ning Hsu, Yu Zhang, James Glass. **Recurrent Neural Network Encoder with Attention for Community Question Answering**. arXiv, 2016.
- Yishu Miao, Lei Yu, Phil Blunsom. **Neural Variational Inference for Text Processing**. ICLR, 2016.

Reference

- Xipeng Qiu and Xuanjing Huang. **Convolutional Neural Tensor Network Architecture for Community-based Question Answering**. IJCAI, 2015.
- Richard Socher, Danqi Chen, Christopher D. Manning, Andrew Y. Ng. **Reasoning with Neural Tensor Networks for Knowledge Base Completion**. NIPS, 2013.
- Zhao Yan, Nan Duan, Jun-Wei Bao, Peng Chen, Ming Zhou, Zhoujun Li, Jianshe Zhou. **DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents**. ACL, 2016.
- Baotian Hu, Zhengdong Lu, Hang Li, Qingcai Chen. **Convolutional Neural Network Architectures for Matching Natural Language Sentences**. NIPS, 2014.
- Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, Xueqi Cheng. **A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations**. AAAI, 2016.

Reference

- Yoon Kim. **Convolutional Neural Networks for Sentence Classification**. EMNLP, 2014.
- Mihai Surdeanu, Massimiliano Ciaramita, Hugo Zaragoza. **Learning to Rank Answers on Large Online QA Collections**. ACL, 2008.
- Mengqiu Wang, Noah A. Smith, Teruko Mitamura. **What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA**. EMNLP, 2007.
- Michael Heilman and Noah A. Smith. **Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions**. NAACL, 2010.
- Wen-tau Yih, Ming-Wei Chang, Christopher Meek, Andrzej Pastusiak. **Question Answering Using Enhanced Lexical Semantic Models**. ACL, 2013.

Reference

- S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, M. Gatford**. Okapi at TREC-3**. TREC, 1996.
- Dekang Lin and Patrick Pantel. **DIRT – Discovery of Inference Rules from Text**. KDD, 2001.
- Colin Bannard and Chris Callison-Burch. **Paraphrasing with Bilingual Parallel Corpora**. ACL, 2005.
- Richard Socher, Eric Huang, Jeffrey Pennington, Andrew Ng, Christopher Manning. **Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection**. NIPS, 2011.
- Cicero dos Santos, Luciano Barbosa, Dasha Bogdanova, Bianca Zadrozny. **Learning Hybrid Representations to Retrieve Semantically Equivalent Questions**. ACL, 2015.

Reference

- Remi Lebret, David Grangier, Michael Auli. **Neural Text Generation from Structured Data with Application to the Biography Domain**. EMNLP, 2016.
- Lili Mou, Zhengdong Lu, Hang Li, Zhi Jin. **Coupling Distributed and Symbolic Execution for Natural Language Queries**. ICML, 2017.
- Sujay Kumar Jauhar, Peter D. Turney, Eduard Hovy. **Tables as Semi-structured Knowledge for Question Answering**. ACL, 2016.
- Li Dong, Furu Wei, Ming Zhou, Ke Xu. **Question Answering over Freebase with Multi-Column Convolutional Neural Networks**. ACL, 2015.
- Stephen Wan, Mark Dras, Robert Dale, C´ecile Paris. **Using Dependency-Based Features to Take the "Para-farce" out of Paraphrase**. ALTW, 2006.

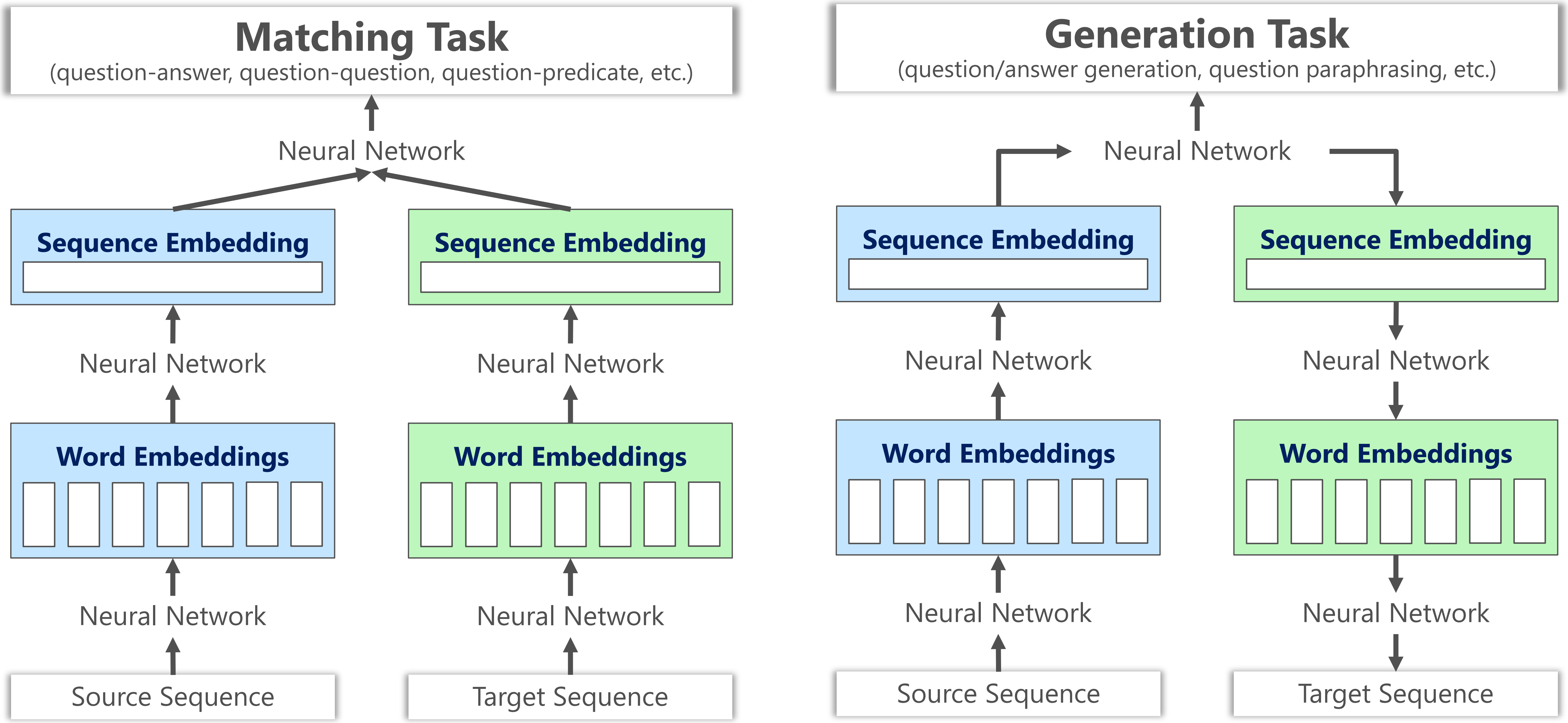
Reference

- Wei-Ning Hsu, Yu Zhang, James Glass. **Recurrent Neural Network Encoder with Attention for Community Question Answering**. arXiv, 2016.

- Salvatore Romeo, Giovanni Da San Martino, Alberto Barr´on-Cede˜no, Alessandro Moschitti, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Mitra Mohtarami, James Glass. **Neural Attention for Learning to Rank Questions in Community Question Answering**. COLING, 2016.

Backups

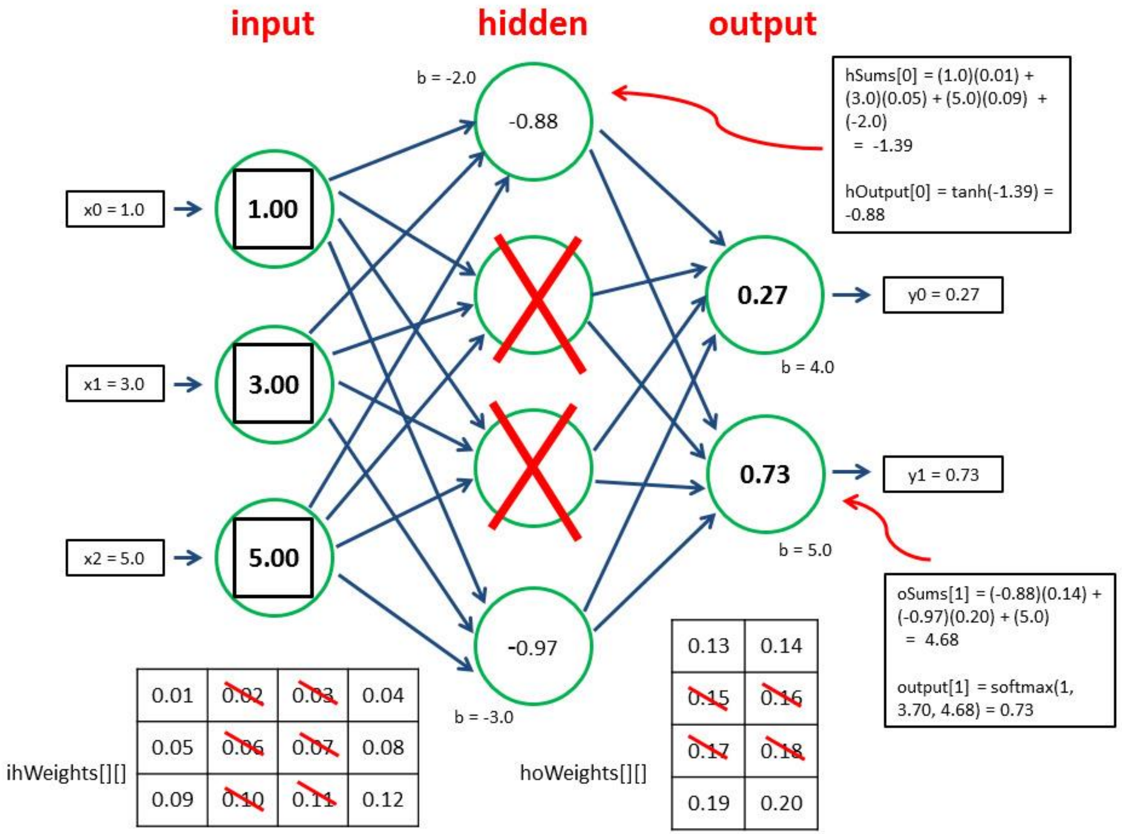
Neural Network for QA/QG Tasks

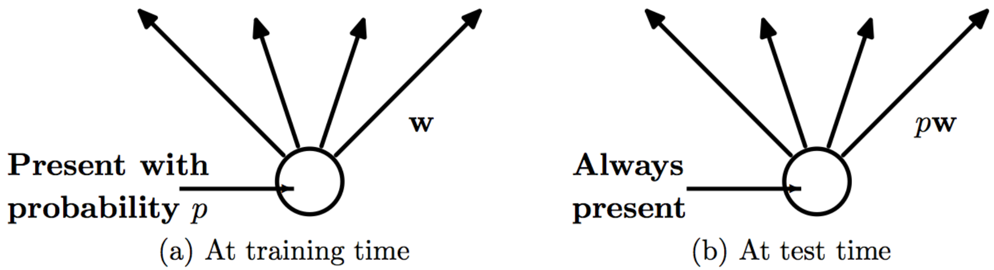


Deep Learning Basics

Dropout

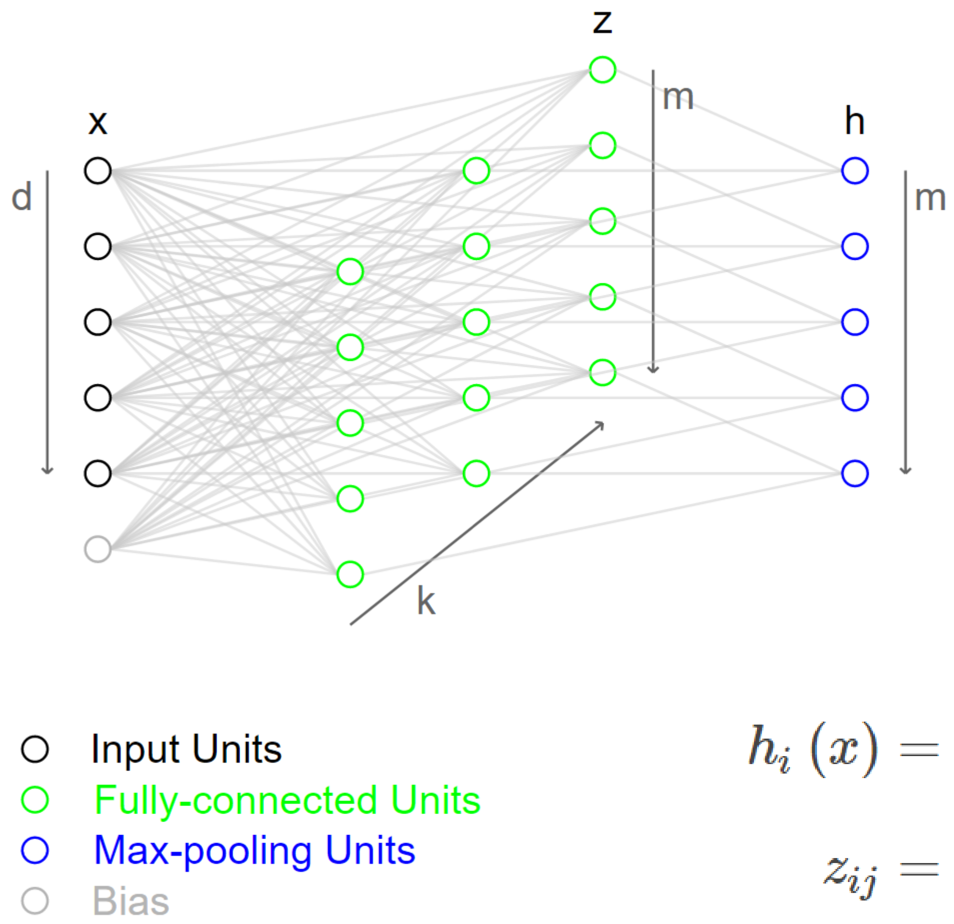
- Dropout is a regularization technique
- Dropout is implemented by dropping neurons with a certain probability $1 - p$ during training
- Dropout is implemented by scaling all weights with a certain probability p during testing





Maxout

- Maxout is an activation function
- Maxout is a piecewise linear approximation to an arbitrary convex function



$$h_i(x) = \max_{j \in [1,k]} (z_{ij})$$

$$z_{ij} = x^T W_{\ldots ij} + b_{ij}$$

h	*Maxout* function
x	Input ($\in \mathbb{R}^d$)
W	4D tensor of learned weights ($\in \mathbb{R}^{d \times m \times k}$)
d	Number of input units (length of x)
m	Number of units in each linear feature extractor (complexity)
k	Number of linear feature extractors
b	Matrix of learned biases ($\in \mathbb{R}^{m \times k}$)
i	Runs over the number of *Maxout* units ($\in [1, m]$)
j	Runs over the number of feature extractors ($\in [1, k]$)

CommunityQA

n-gram Matching

(Wan et al., 2006)

- Word-overlap
- (Tree) Edit distance
- Longest Common Sequence
- BLEU score
- …

Pros
- Easy to define
- Efficient to compute

Cons
- Ignore stop-words
- Ignore synonyms

Okapi BM25

(Robertson et al., 1996)

$$f_{\overrightarrow{BM25}}(Q_{in}, Q_c) = \sum_{i=1}^{|Q_{in}|} IDF(Q_{in}^i) freq(Q_{in}^i, Q_c)$$

$$f_{\overleftarrow{BM25}}(Q_c, Q_{in}) = \sum_{j=1}^{|Q_c|} IDF(Q_c^j) freq(Q_c^j, Q_{in})$$

$$f_{BM25}(Q_{in}, Q_c) = \frac{f_{\overrightarrow{BM25}}(Q_{in}, Q_c) + f_{\overleftarrow{BM25}}(Q_c, Q_{in})}{2}$$

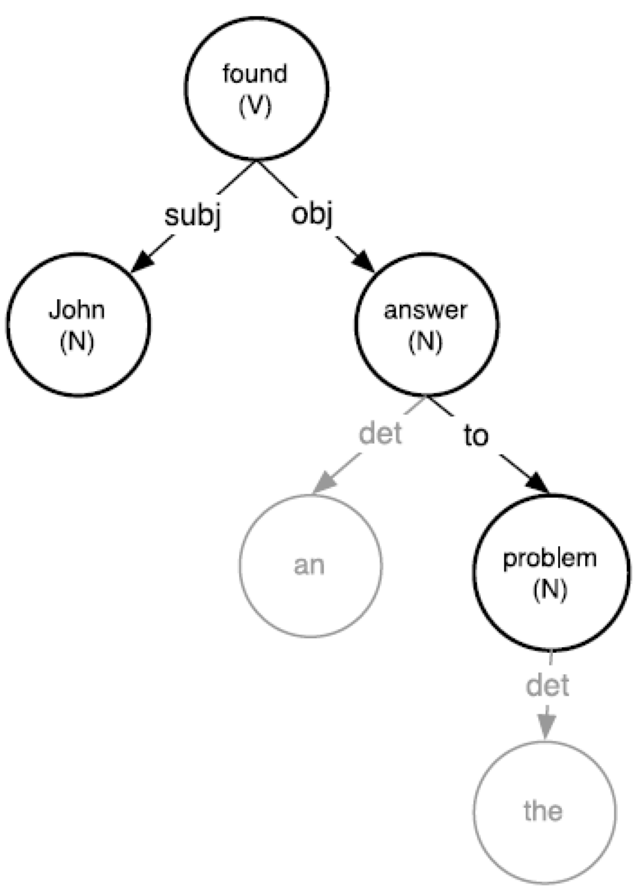
$$IDF(w) = \log \frac{N - n(w) + 0.5}{n(w) + 0.5}$$

Penalize high-frequent words

$$freq(w, D) = \frac{\#(w, D)(k_1 + 1)}{\#(w, D) + k_1 (1 - b + b \cdot \frac{|D|}{avglen(D)})}$$

Discovery of Inference Rules from Text (DIRT)

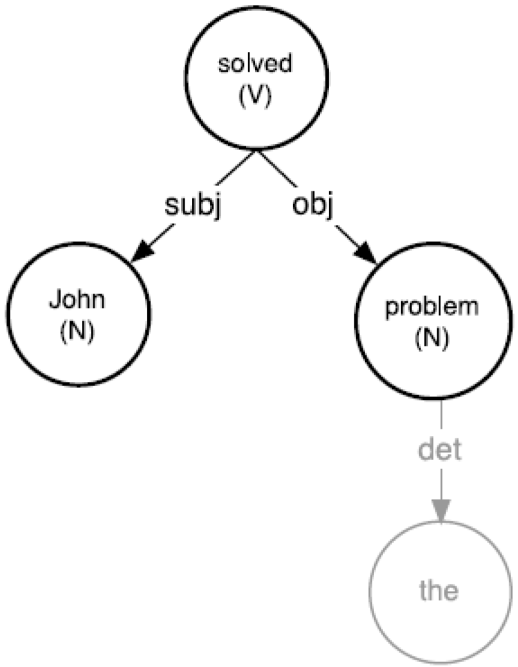
(Lin and Pantel, 2001)



X Y

N:subj:V <- find-> V:obj:N -> answer -> N:to:N

"**X** found answer to **Y**"



X Y

N:subj:V <- solve-> V:obj:N

"**X** solved **Y**"

$$mi(p, X, w) = \log\left(\frac{|p, X, w| \times |*, X, *|}{|p, X, *| \times |*, X, w|}\right)$$

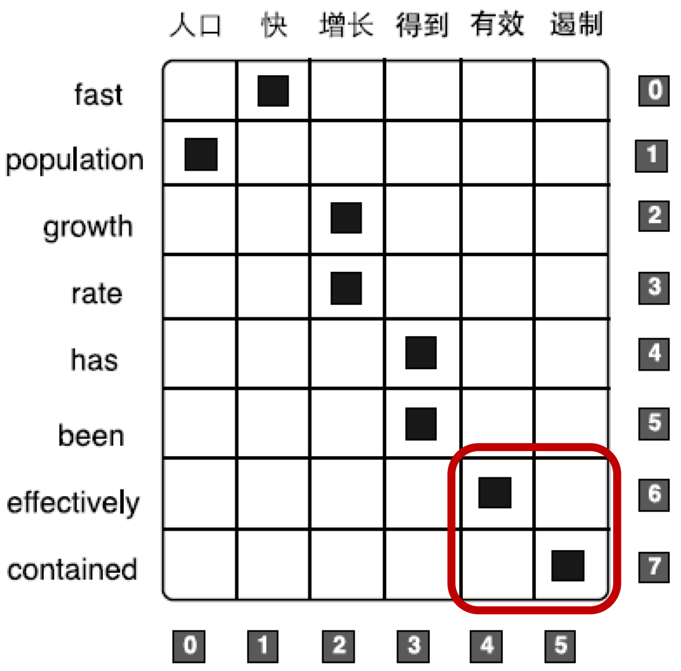
$$slots_i = (p_i, s)$$

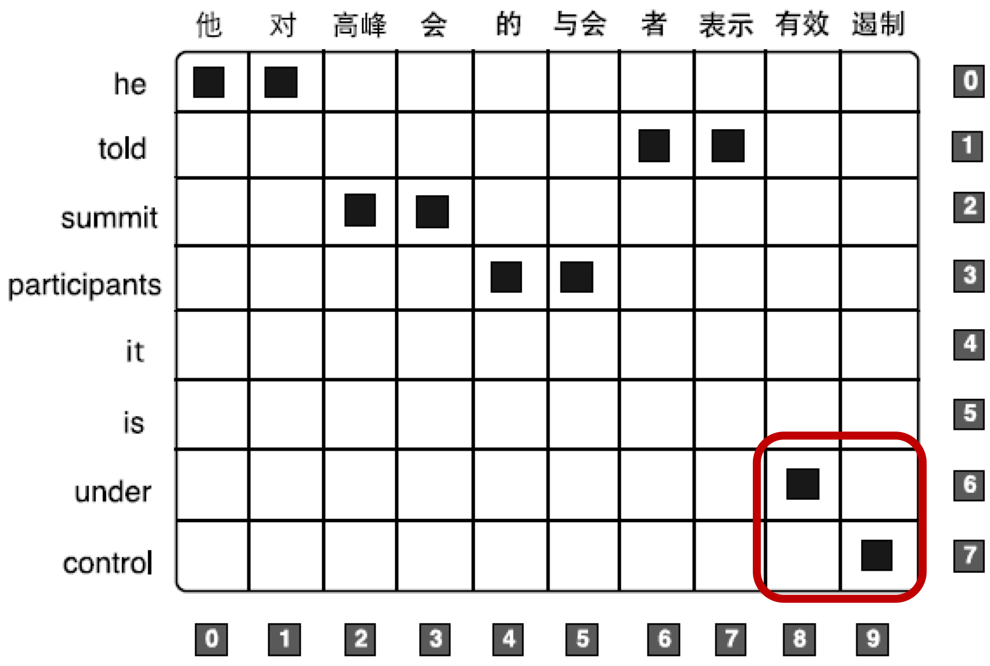
$$sim(slots_1, slots_2) = \frac{\sum_{w \in T(p_1,s) \cap T(p_2,s)} (mi(p_1, s, w) + mi(p_2, s, w))}{\sum_{w \in T(p_1,s)} mi(p_1, s, w) + \sum_{w \in T(p_2,s)} mi(p_2, s, w)}$$

$$sim(p_1, p_2) = \sqrt{sim(slotX_1, slotX_2) \times sim(slotY_1, slotY_2)}$$

Paraphrasing with Bilingual Parallel Corpora

(Bannard and Callison-Burch, 2005)





Extracted phrases

(0,0) x (1,1) → <人口, population>
(1,1) x (0,0) → <快, fast>
(2,2) x (2,3) → <增长, growth rate>
...
...
(4,5) x (6,7) → <有效 遏制, effectively contained>
...
...

Extracted phrases

(0,1) x (0,0) → <他 对, he>
(6,7) x (1,1) → <者 表示, told>
(2,3) x (2,2) → <高峰 会, summit>
...
...
(8,9) x (6,7) → <有效 遏制, under control>
...
...

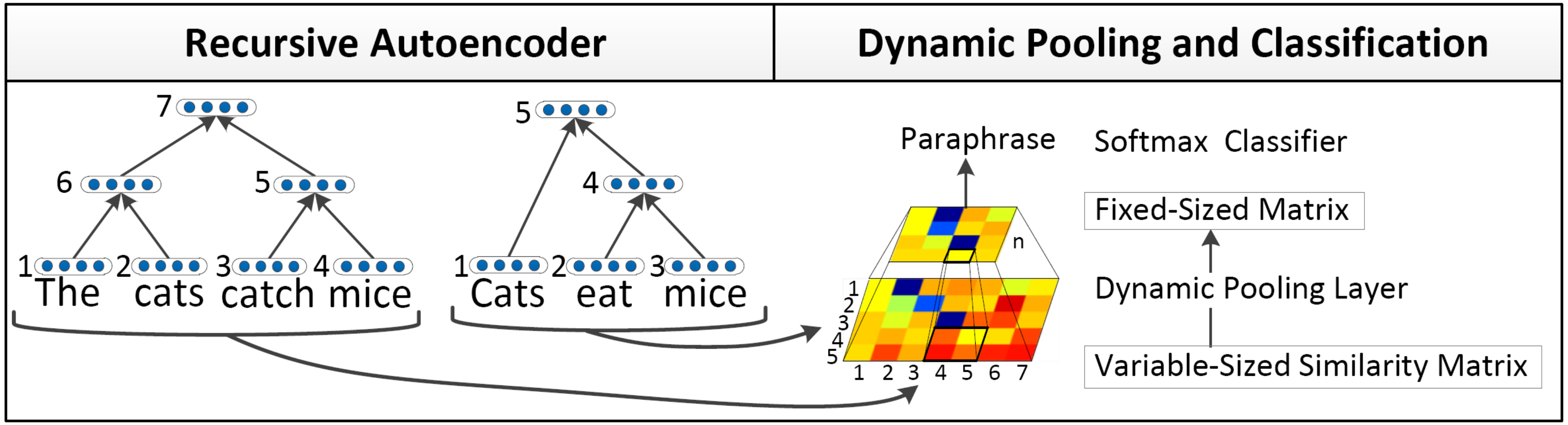
Phrasal Paraphrase Table

...
founder ||| creator ||| 0.01214179
founder ||| founded ||| 0.00694428
founder ||| start ||| 0.00280628
founder ||| set up ||| 0.00088949
founder ||| established ||| 0.00065527
founder ||| pioneer ||| 0.00047020
...
a athlete ||| a player ||| 0.06162226
a athlete ||| a sportsman ||| 0.03968052
a athlete ||| a sports ||| 0.00810965
a athlete ||| a runner ||| 0.00622368
...
net impact ||| net effect ||| 0.11601010
net impact ||| net influence ||| 0.00349070
net impact ||| net result ||| 0.00064136
...

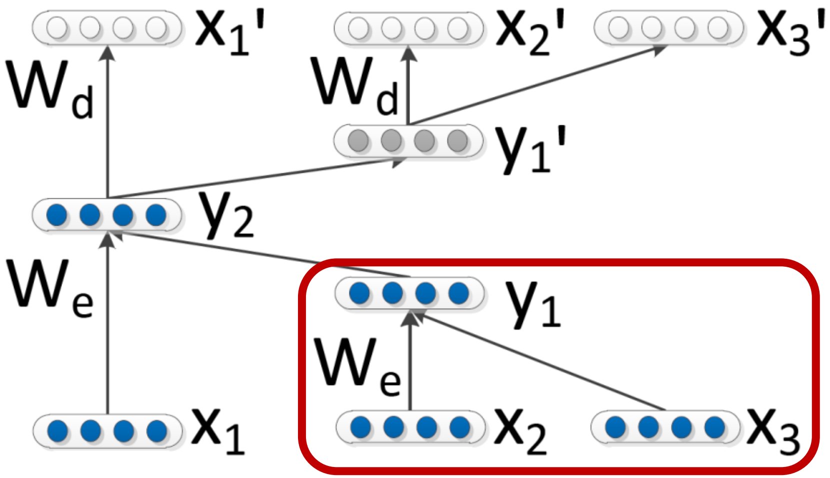
$$p(s_j|s_i) = \sum_t p(t|s_i) \cdot p(s_j|t)$$

Unfolding Recursive Autoencoder + Dynamic Pooling

(Socher et al., 2011)



Unfolding Recursive Autoencoder

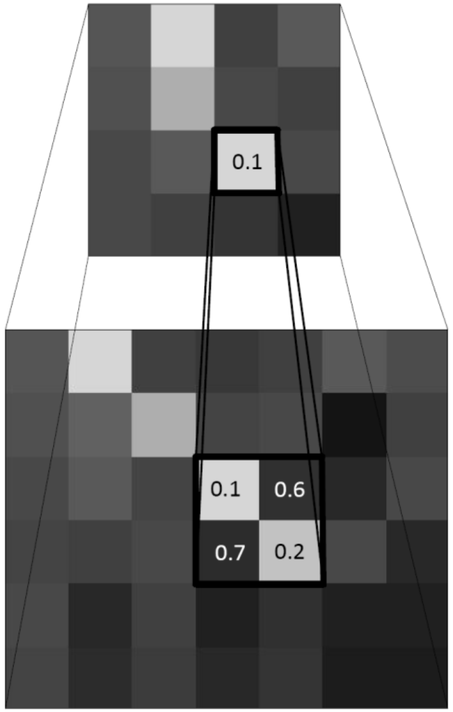


The unfolding autoencoder tries to encode each hidden layer such that it best reconstructs its entire subtree to the leaf nodes.

$$y_1 = f(W_e[x_2; x_3] + b_e)$$

$$[x_2'; x_3'] = f(W_d y_1 + b_d)$$

$$E_{rec}(y_1) = \|[x_2; x_3] - [x_2'; x_3']\|^2$$

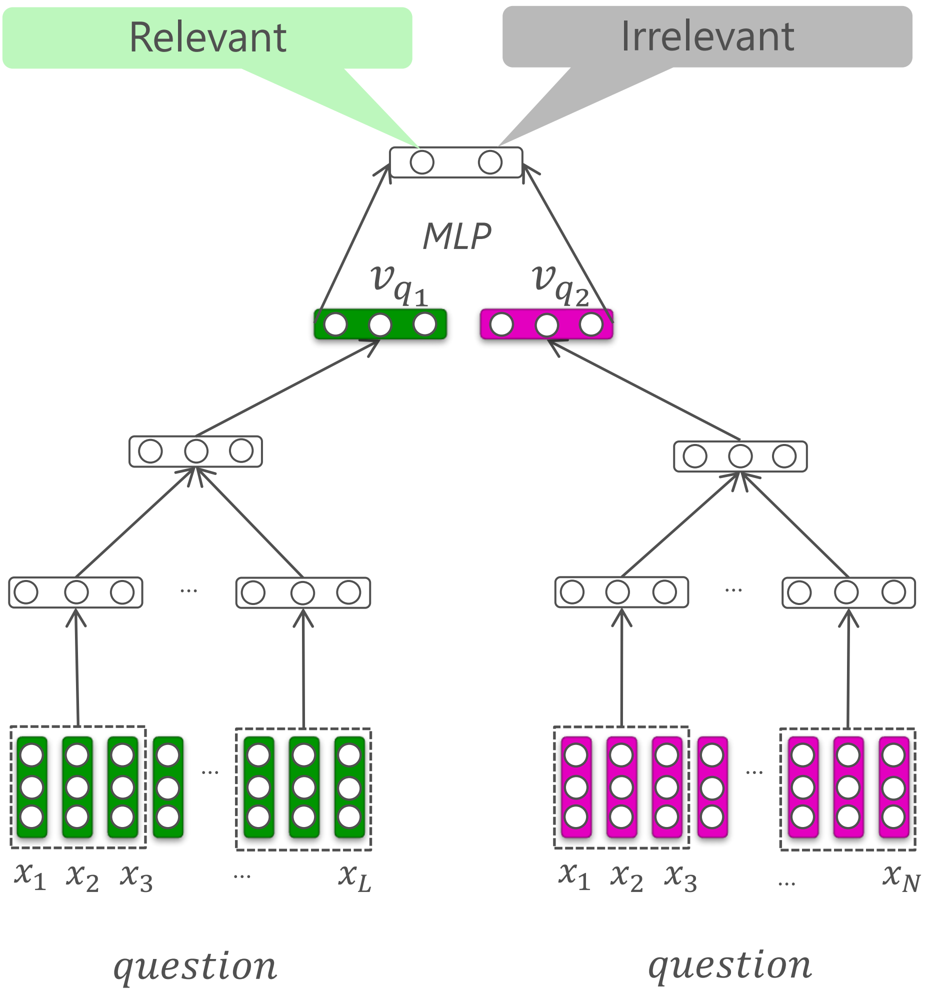


Dynamic Pooling

1. Generate similarity matrix S;
2. Split S into $n_p \times n_p$ grids;
3. Run pooling on each grid;
4. Output a fixed-size matrix.

Convolution Neural Network

(Santos et al., 2015)



$$v_{q_1} = \tanh(W_O \cdot \hat{c} + b_O)$$

$$\hat{c}^j = \max\{c_1^j, \dots, c_L^j\}$$

$$c_i = f(W_C \cdot x_{i:i+d-1} + b_C)$$

$$v_{q_2} = \tanh(W_O \cdot \hat{c} + b_O)$$

$$\hat{c}^j = \max\{c_1^j, \dots, c_N^j\}$$

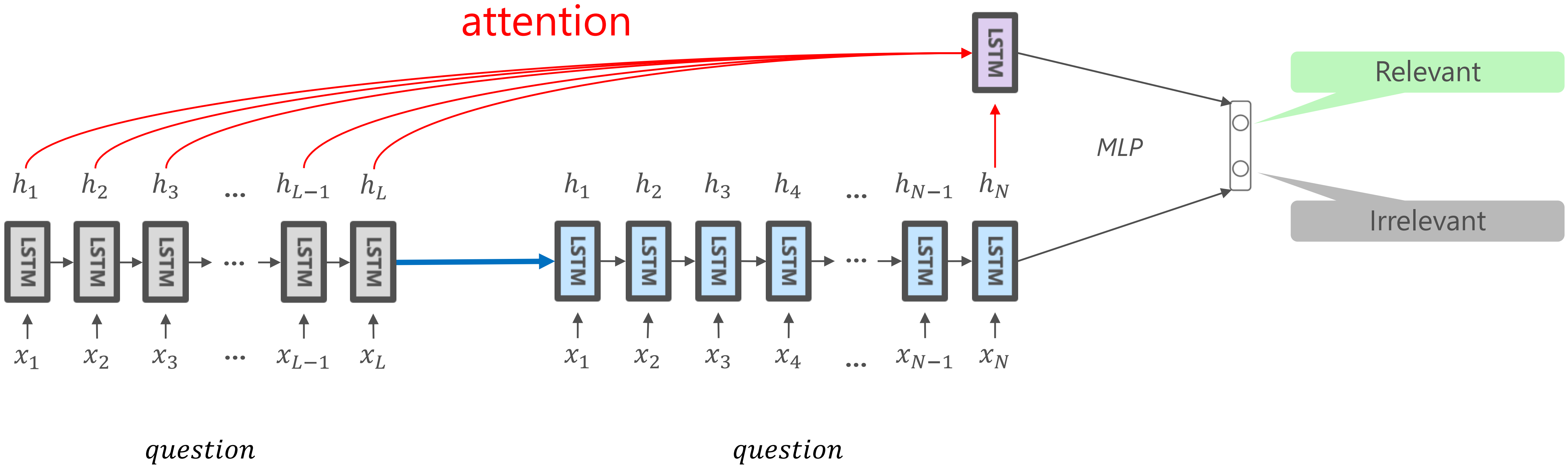
$$c_i = f(W_C \cdot x_{i:i+d-1} + b_C)$$

Recurrent Neural Network

(Hsu et al., 2016; Romeo et al., 2016)

$$h' = \sum_{i=1}^{L} \alpha_i h_i$$

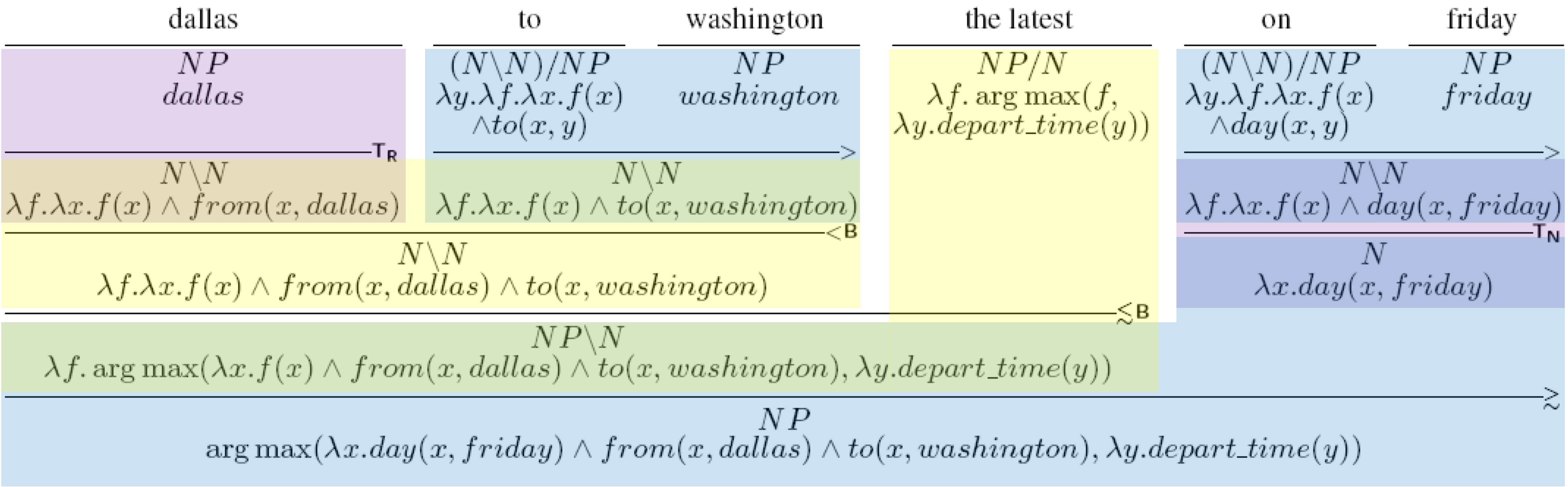
$$\alpha_i = \frac{\exp a(h_i, h_N)}{\sum_{j=1}^{L} \exp a(h_j, h_N)}$$



QA with Structured Data

Semantic Parsing with Combinatorial Categorial Grammar

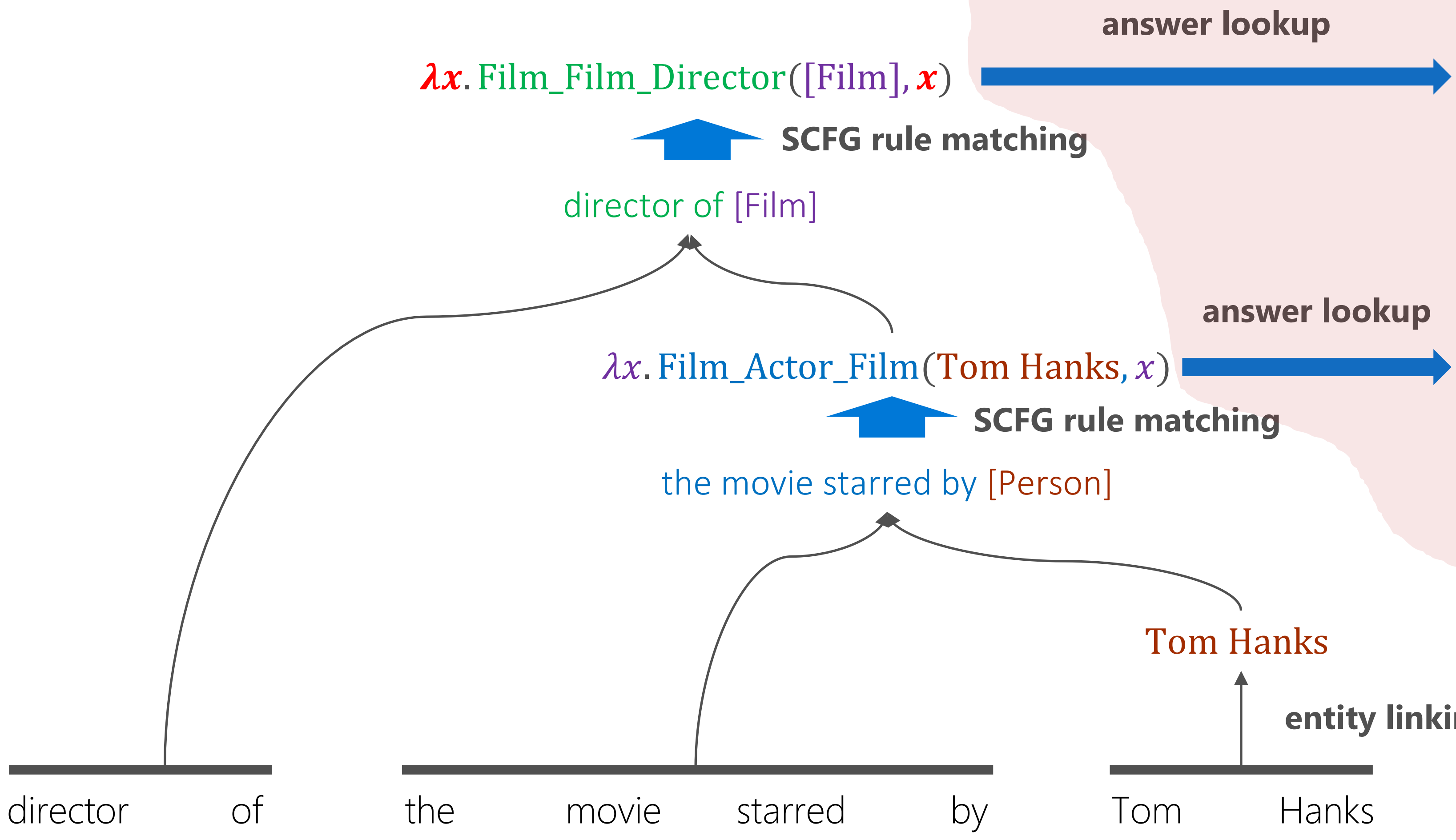
(Zettlemoyer and Collins, 2007; Kwiatkowski et al., 2011)



Functional Application **Type-Raising** **Functional Composition**

Semantic Parsing with SCFG

(Bao et al., 2014; Wong and Mooney, 2007)



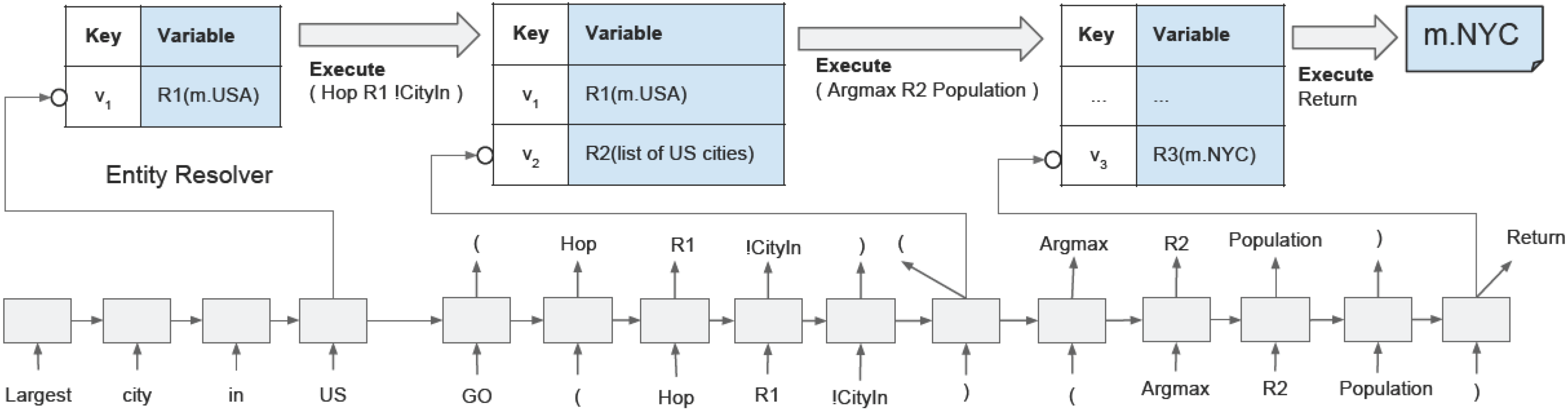
Subject Entity	Predicate	Object Entity
Saving Private Ryan	Film_Film_Director	Steven Spielberg
Forrest Gump	Film_Film_Director	Robert Zemeckis
…	…	…

Subject Entity	Predicate	Object Entity
Tom Hanks	Film_Actor_Film	Saving Private Ryan
Tom Hanks	Film_Actor_Film	Forrest Gump
…	…	…

Using internal answers for pruning can greatly reduce the size of the search space.

Semantic Parsing with Neural Symbolic Machine

(Liang et al., 2017)



Manager

- Provide weak supervision through a reward indicating how well a task is accomplished.

Programmer

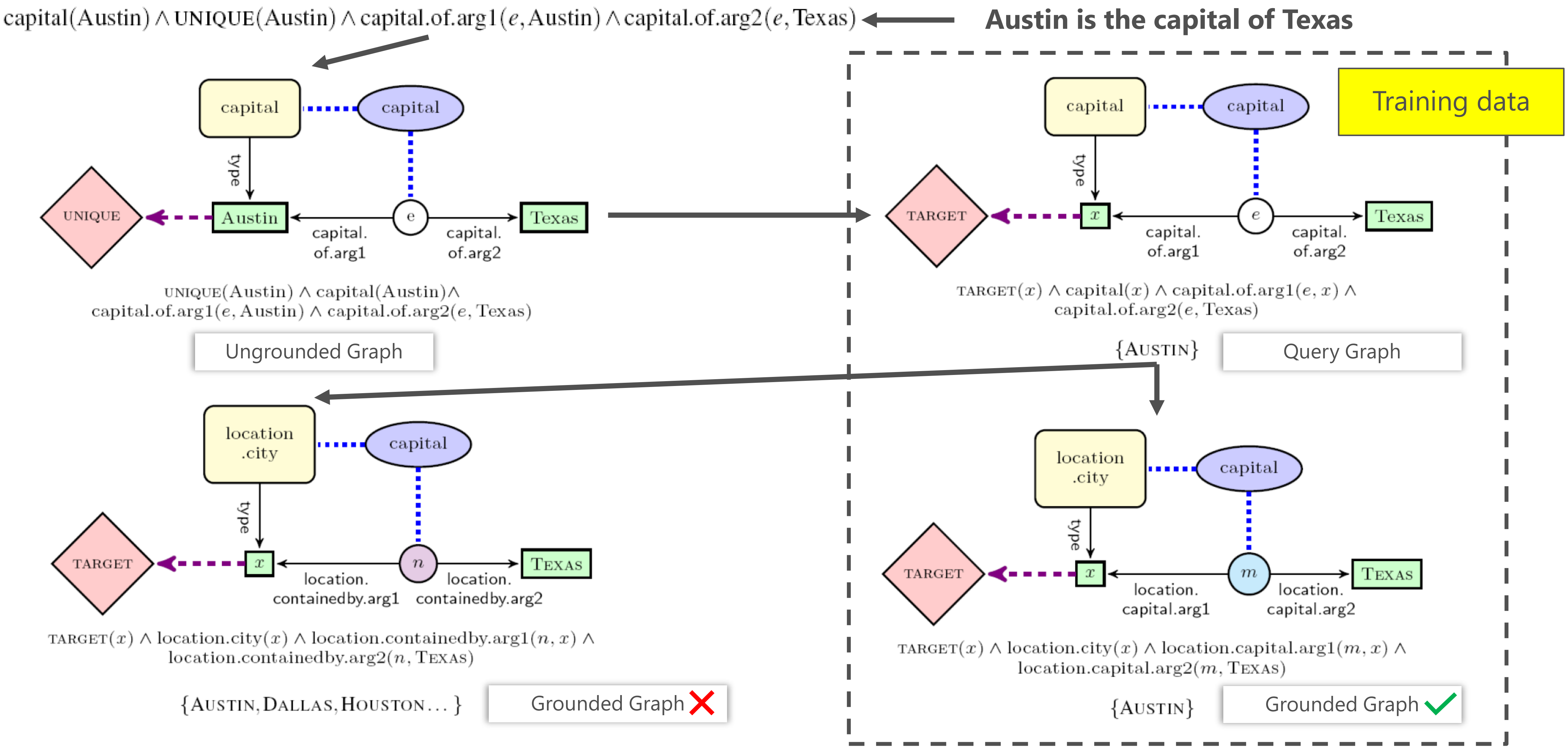
- Take natural language as input and generate a program that is a sequence of tokens.
- $C = (c_1, \ldots, c_N)$
- c_i
    1. GO/Return
    2. $(F\ A_1 \ldots A_K)$

Computer

- Execute programs in a high-level programming language.

Semantic Parsing with Graph Matching

(Reddy et al., 2014; Reddy et al., 2017)

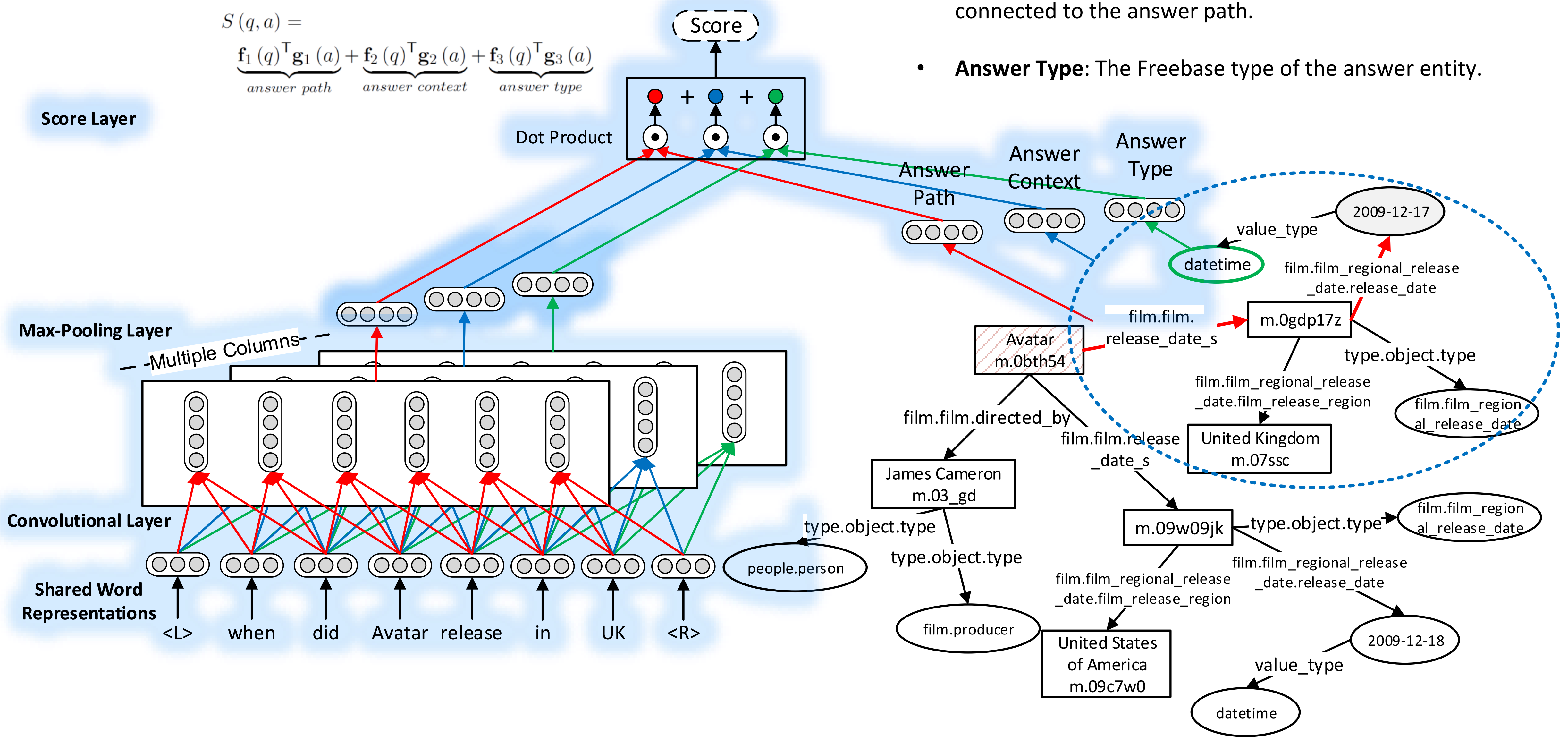


Multi-Column CNN for KBQA

(Dong et al., 2015)

- **Answer Path**: The set of predicates between answer entity and question entity.
- **Answer Context**: The 1-hop entities and predicate connected to the answer path.
- **Answer Type**: The Freebase type of the answer entity.

$$S(q,a) = \underbrace{\mathbf{f}_1(q)^{\mathsf{T}}\mathbf{g}_1(a)}_{answer\ path} + \underbrace{\mathbf{f}_2(q)^{\mathsf{T}}\mathbf{g}_2(a)}_{answer\ context} + \underbrace{\mathbf{f}_3(q)^{\mathsf{T}}\mathbf{g}_3(a)}_{answer\ type}$$



Answer Ranking with Memory Network

(Miller et al., 2016)

Each candidate memory is assigned a relevance probability by comparing its key with the question

$$p_{h_i} = \mathrm{Softmax}(A\Phi_X(x) \cdot A\Phi_K(k_{h_i}))$$

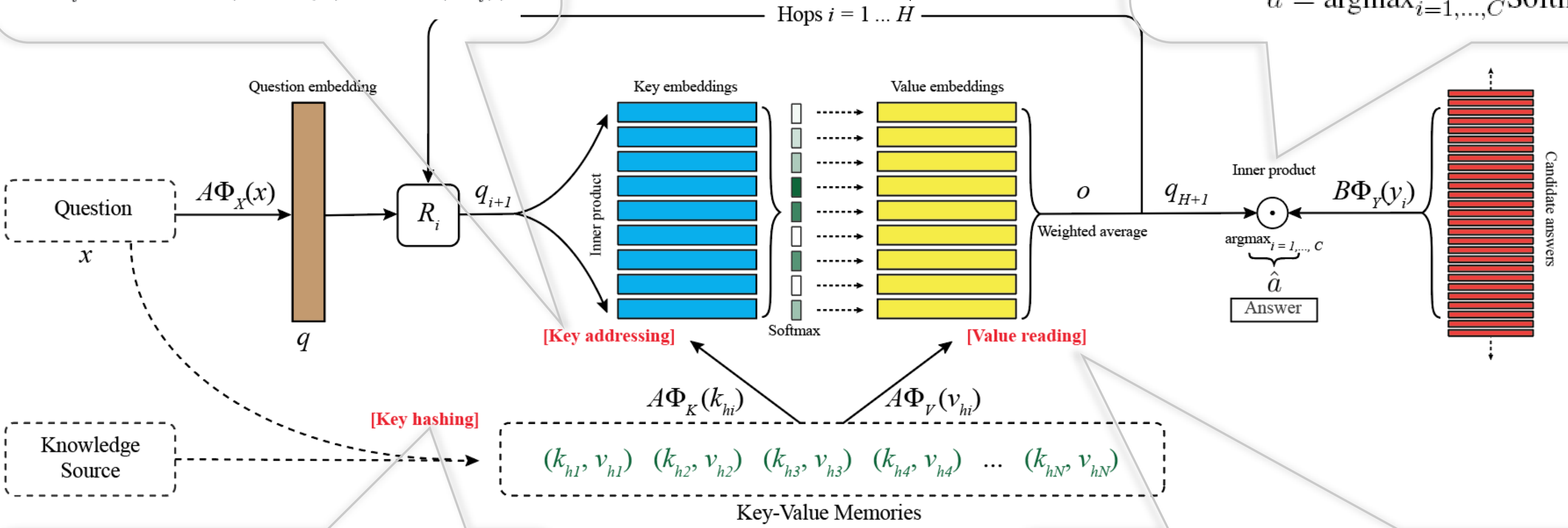
H is a fixed size

After receiving the result o, the query is updated with:

$$q_2 = R_1(q + o)$$

After a fixed number of H hops, the final prediction is:

$$\hat{a} = \mathrm{argmax}_{i=1,\ldots,C}\mathrm{Softmax}(q_{H+1}^{\top} B\Phi_Y(y_i))$$



Finds a subset of K-V pairs, where each key shares at least one word with the question

K-V pairs

- <sbj+pred, obj>
- <sentence, sentence>
- <window, center word>
- …

The values of the memories are read by taking their weighted sum using the addressing probabilities, and o is returned.

$$o = \sum_i p_{h_i} A\Phi_V(v_{h_i})$$

Table

- Semi-structured data with flexible schema

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Pairs	France	24
…	…	…	…
2004	Athens	Greece	201
2008	Beijing	China	204

Host Cities of Summer Olympic Games

- **Table Cell**
  Objects/Values in the world
- **Table Header**
  The type of table cells in the same column
- **Table Row**
  A record of an information piece
- **Table Caption**
  Summary of the entire table

Characteristics of Knowledge Base and Table

	Knowledge Base	**Table**
Schema	Well-defined	Flexible
Type of Knowledge	Common Knowledge (Person, Location, Organization, Film, ...)	Both Common and Domain Knowledge (Product, Finance, Sport, Profession, ...)
Size	Large-scale	Even larger
Data Quality	High	Moderate
Operability	Hard	Easy

Semantic Parsing-based TBQA with Neural Networks

(Zhong et al., 2017)

Question: how many CFL teams are from York College?

$$\alpha_t^{inp} = W^{inp} h_t^{enc}$$

$$\beta^{inp} = \mathrm{softmax}(\alpha^{inp})$$

$$\kappa^{agg} = \sum_t \beta_t^{inp} h_t^{enc}$$

$$\alpha^{agg} = W^{agg} \tanh(V^{agg} \kappa^{agg} + b^{agg}) + c^{agg}$$

Operator Selection

COUNT **MAX** **MIN** **...** **NULL**

Column Selection

$$\alpha_j^{sel} = W^{sel} \tanh\left(V^{sel} \kappa^{sel} + V^c e_j^c\right)$$

Pick #	CFL Team	Player	Position	College
27	Hamilton Tiger-Cats	Connor Healy	DB	Wilfrid Laurier
28	Calgary Stampeders	Anthony Forgone	OL	York
29	Ottawa Renegades	L.P. Ladouceur	DT	California
30	Toronto Argonauts	Frank Hoffman	DL	York
...	...	...	...	...

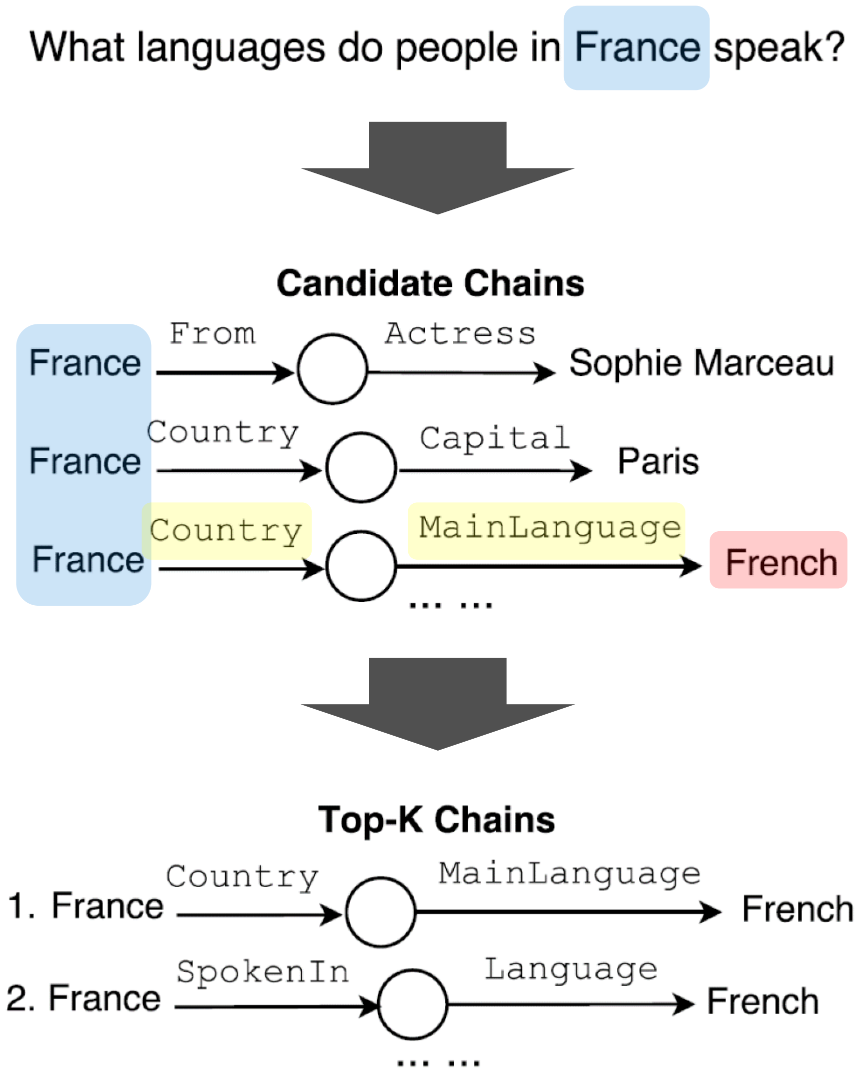
Pointer Network

$$\alpha_{s,t}^{ptr} = W^{ptr} \tanh(U^{ptr} g_s + V^{ptr} h_t)$$

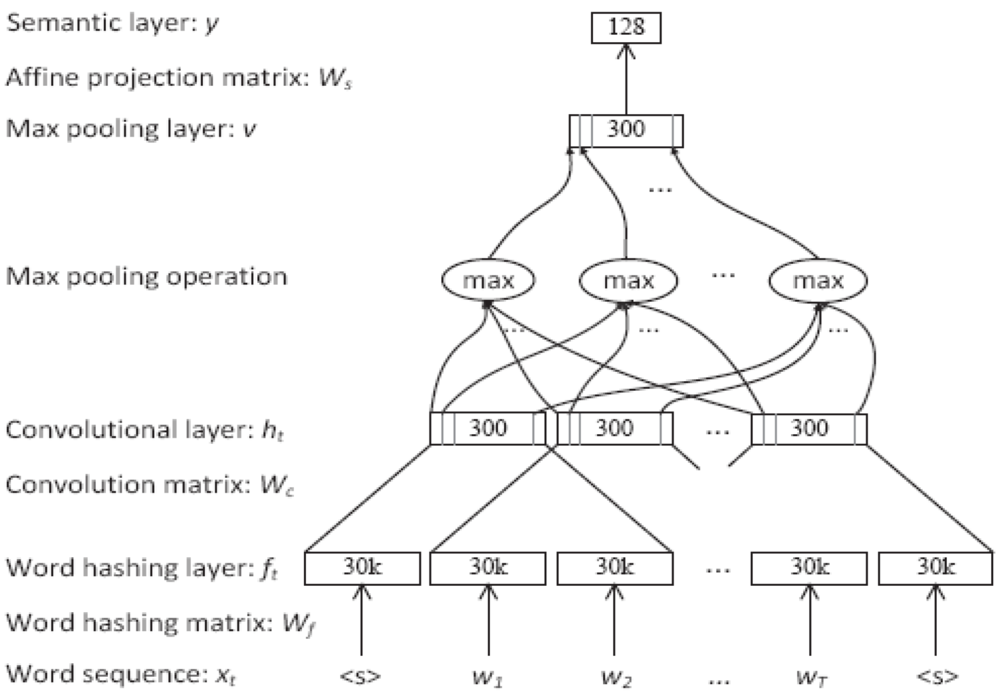
```
SELECT COUNT
CFL Team FROM CFLDraft
WHERE College = "York"
```

Answer Ranking-based TBQA with Features

(Sun et al., 2016; Jauhar et al., 2016)



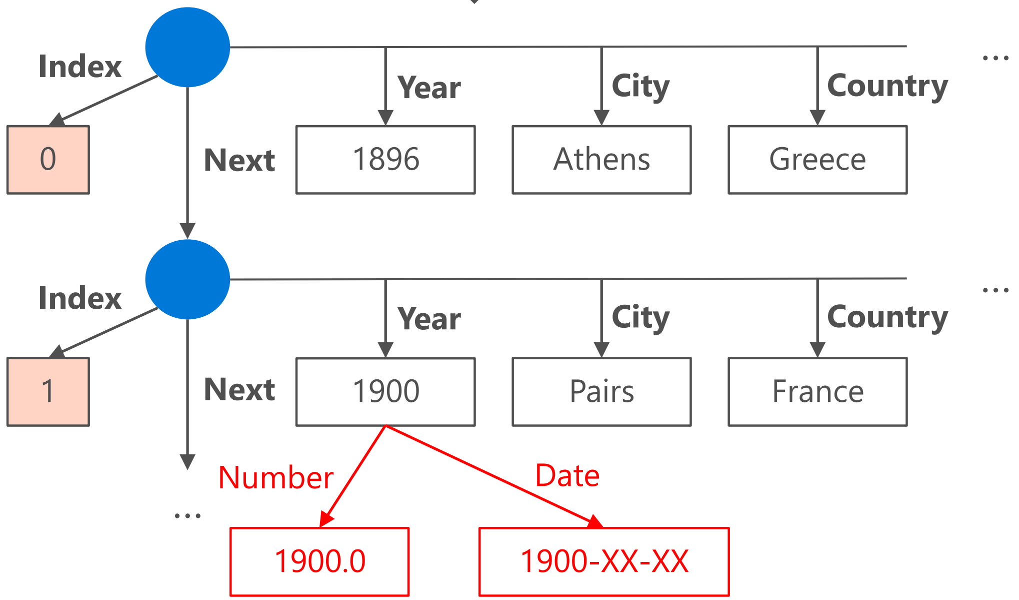
Country	Capital	Currency	Main Language
Algeria	Algiers	Dinar	Arabic
Egypt	Cairo	Pound	Arabic
France	Paris	Euro	French
...	...	...	...



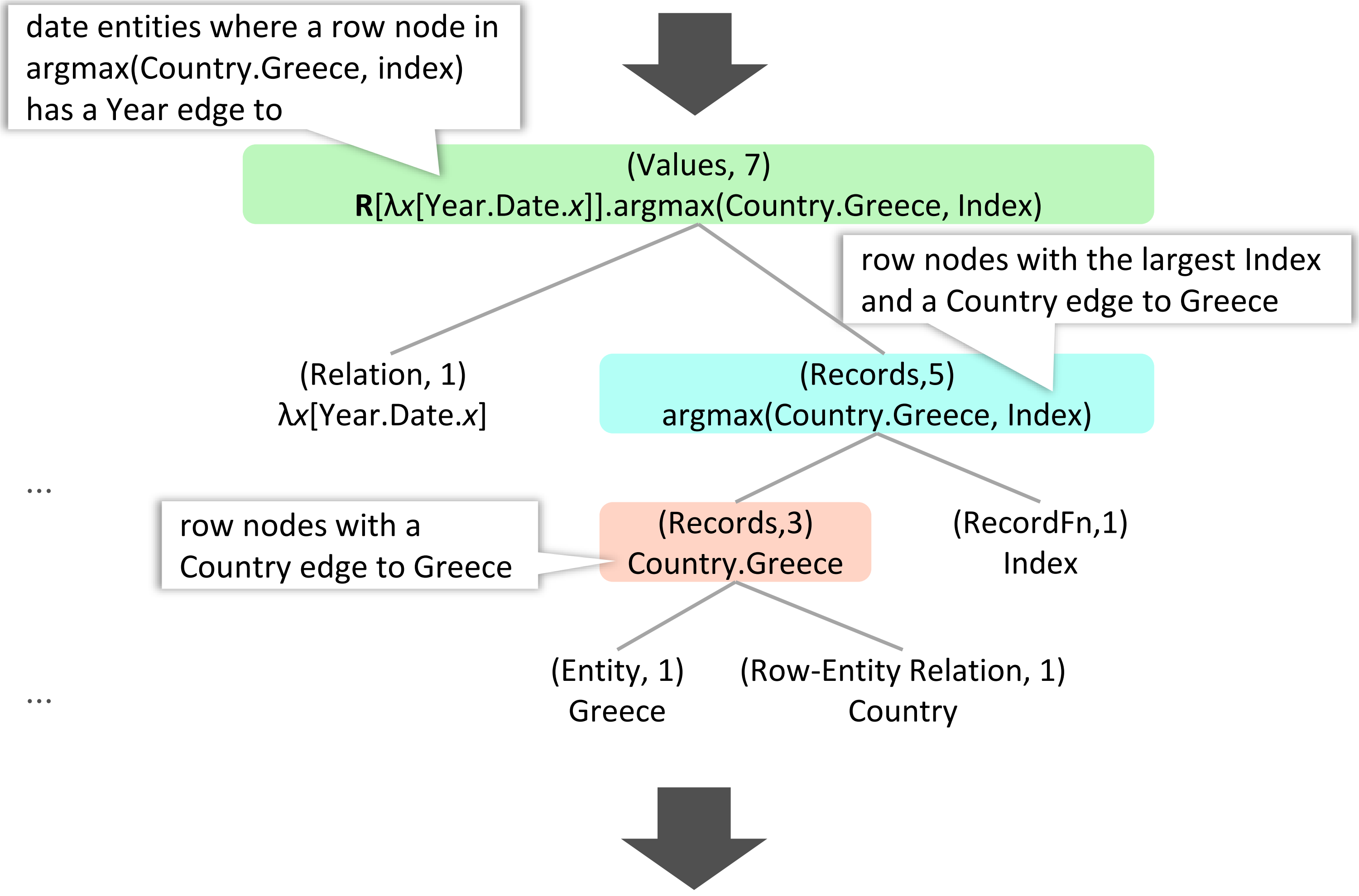
Semantic Parsing-based TBQA with Deduction Rules

(Pasupat and Liang, 2015)

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Pairs	France	24
…	…	…	…
2004	Athens	Greece	201
2008	Beijing	China	204



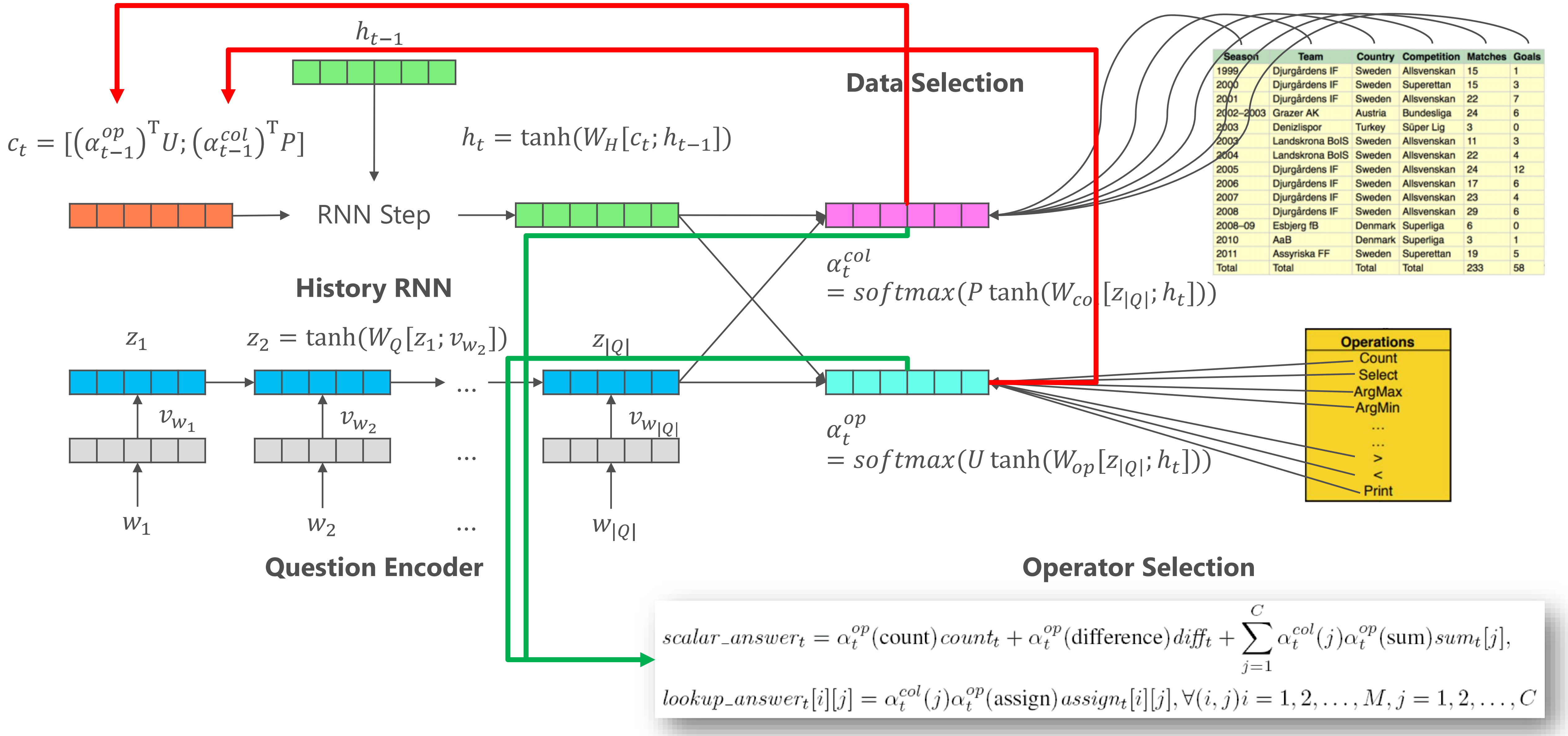
Question: when did Greece hold its last Summer Olympics?



Answer: 2004

TBQA with End-to-End Neural Networks

(Neelakantan et al., 2016; Neelakantan et al., 2017; Mou et al., 2017)



Season	Team	Country	Competition	Matches	Goals
1999	Djurgårdens IF	Sweden	Allsvenskan	15	1
2000	Djurgårdens IF	Sweden	Superettan	15	3
2001	Djurgårdens IF	Sweden	Allsvenskan	22	7
2002–2003	Grazer AK	Austria	Bundesliga	24	6
2003	Denizlispor	Turkey	Süper Lig	3	0
2003	Landskrona BoIS	Sweden	Allsvenskan	11	3
2004	Landskrona BoIS	Sweden	Allsvenskan	22	4
2005	Djurgårdens IF	Sweden	Allsvenskan	24	12
2006	Djurgårdens IF	Sweden	Allsvenskan	17	6
2007	Djurgårdens IF	Sweden	Allsvenskan	23	4
2008	Djurgårdens IF	Sweden	Allsvenskan	29	6
2008–09	Esbjerg fB	Denmark	Superliga	6	0
2010	AaB	Denmark	Superliga	3	1
2011	Assyriska FF	Sweden	Superettan	19	5
Total	Total	Total	Total	233	58

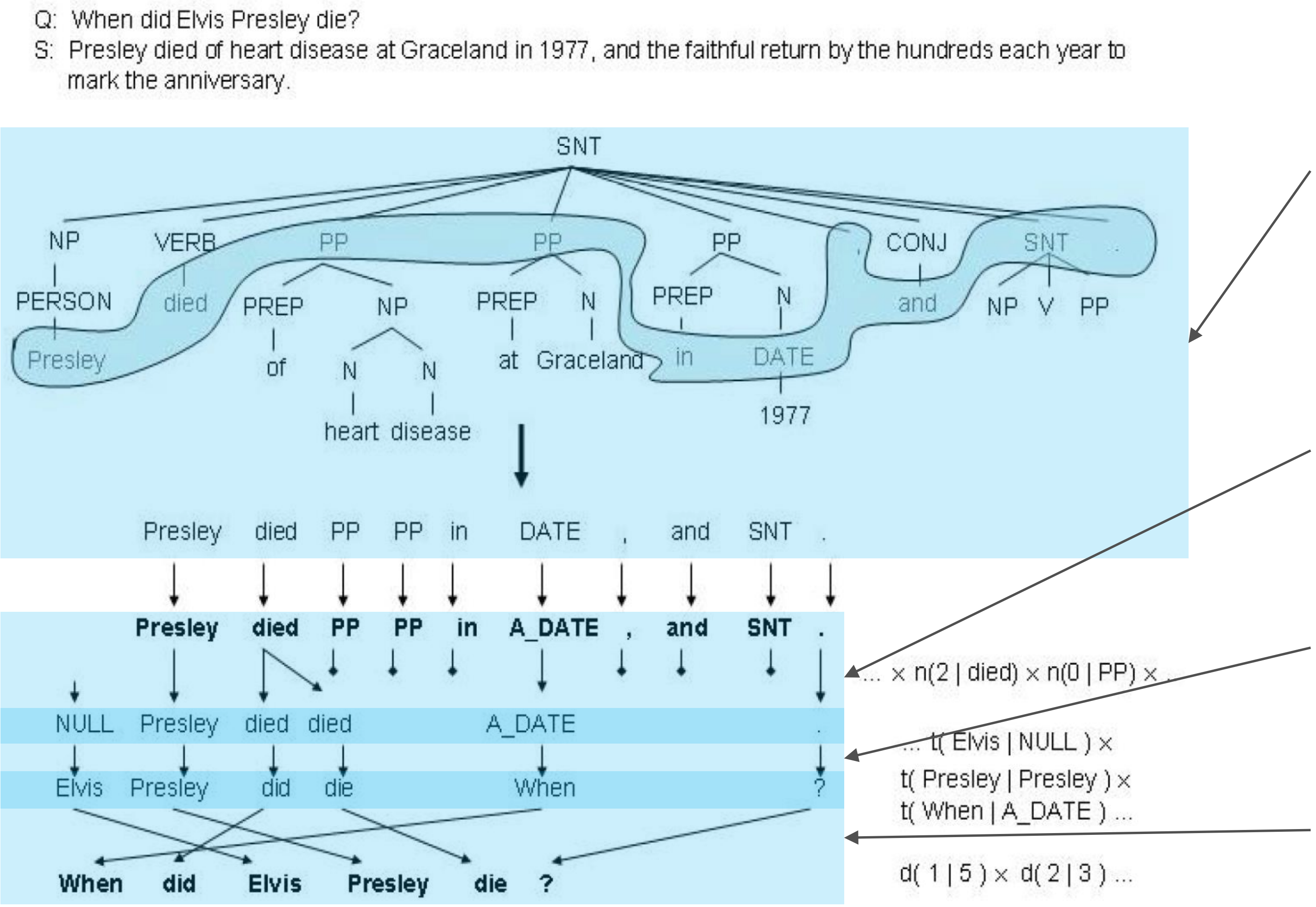
Recap of Structured Data-based QA

- Two types of structured data
  - Knowledge base
  - Table
- Two types of QA methods
  - Semantic parsing-based
  - Answer ranking-based
- Two types of training data
  - <Q, LF> pairs
  - <Q, A> pairs
- Some issues
  - Still lack of training data
  - Current datasets and tasks mainly focus on single-relation questions
  - Neural network-based methods dominate but hard to debug or interpret
  - …

Interaction between QA and QG

Early Work of PBQG using Noisy-Channel Model

(Echihabi and Marcu, 2003)



Answer Sentence "Cut"

1. words overlapped with Q are preserved;
2. A is reduced to its semantic or syntactic class;
3. non-leaves, which don't have any question word or answer offspring, are reduced to their semantic or syntactic class;
4. all remaining leaves are preserved.

Fertility Table $n(\phi|A_i)$

guarantee the number of words in the cut and the number of words in the question match

Replace Table $t(Q_j|A_i)$

replace answer words with question words

Distortion Table $d(j|i)$

permute the question words, in order to obtain a well-formed, grammatical question

trained by IBM models

Interaction between QA and QG via Features

(Duan et al., 2017)

- QA improves QG

$$\hat{Q} = \arg\max_{Q} P_{QG}(Q|A) \quad \Rightarrow \quad \hat{Q} = \arg\max_{Q}\{P_{QG}(Q|A) + \boldsymbol{\alpha} \cdot \boxed{\mathbf{P_{QA}(A|Q)}}\}$$

- QG improves QA

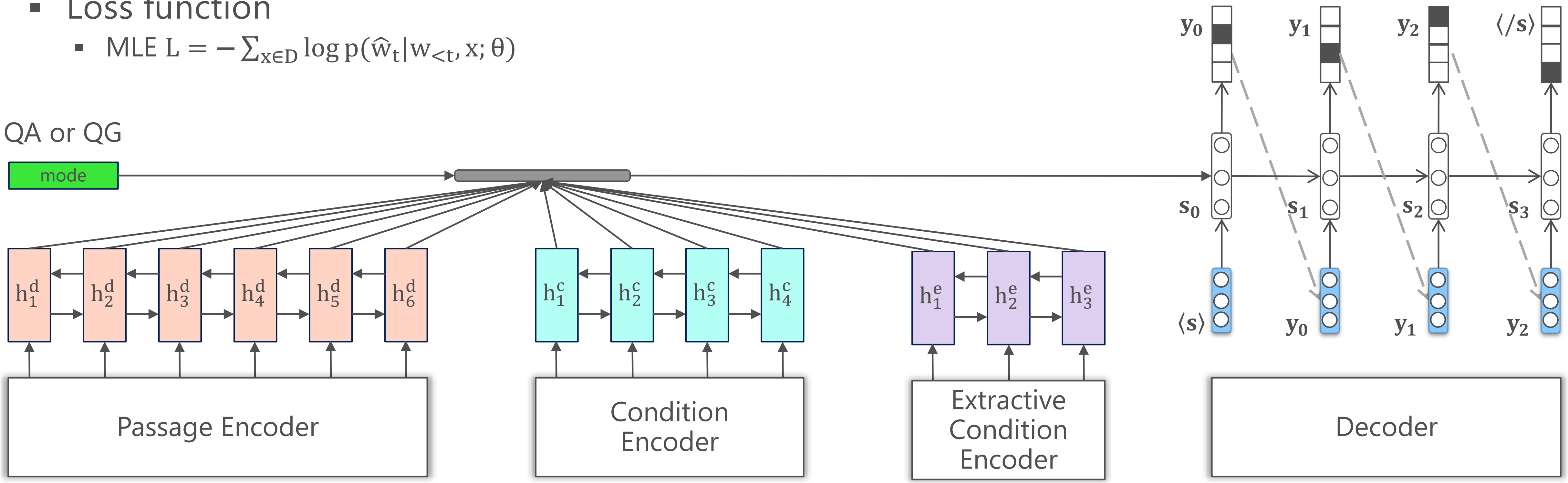
$$\hat{A} = \arg\max_{A} P_{QA}(A|Q) \quad \Rightarrow \quad \hat{A} = \arg\max_{A}\{P_{QA}(A|Q) + \boldsymbol{\beta} \cdot \boxed{\mathbf{P_{QG}(Q|A)}}\}$$

- Forced decoding
- Generate N-best questions of A first, and then compute question-question similarity

Interaction between QA and QG via a Joint Model

(Wang et al., 2017)

- Decoding
  - Take a passage as input
  - Generate a question (answer) as output, conditioned on an answer (question)
- Training
  - Feed answer-generation and question-generation data to the model in an alternating fashion between mini-batches
- Loss function
  - MLE $L = -\sum_{x \in D} \log p(\hat{w}_t | w_{<t}, x; \theta)$



Interaction between QA and QG via Dual Supervised Learning

(Tang et al., 2017)

Question Answering

Language model

Language model

$$P(Q, A) = P(Q)P(A|Q) = P(A)P(Q|A)$$

QA model

QG model

$$\boldsymbol{l_{dual}} = (\log P(A) + \log P(Q|A) - \log P(Q) - \log P(A|Q))^2$$

Question Generation

Update QA model: $\theta_{QA} = sgd(\theta_{QA},\ \nabla_{\theta_{QA}} l_{QA} + \boldsymbol{\lambda_{QA} \nabla_{\theta_{QA}} l_{dual}})$

Update QG model: $\theta_{QG} = sgd(\theta_{QG},\ \nabla_{\theta_{QG}} l_{QG} + \boldsymbol{\lambda_{QG} \nabla_{\theta_{QG}} l_{dual}})$

Interaction between QA and QG via GDAN

(Yang et al., 2017)

$$\max J_D(L, d_{true}) + J_D(U_G, d_{gen})$$

$$\max J_G(U_G, d_{true})$$

Discriminative Model

Generative Model

$$J_D(L, d_{true}) = \frac{1}{|L|} \sum_{(p^i,q^i,a^i)\in L} \log P_{D,d_{true}}(a_i|p^i, q^i)$$

$$J_D(U_G, d_{gen}) = \frac{1}{|U_G|} \sum_{(p^i,q^i,a^i)\in U_G} \log P_{D,d_{gen}}(a_i|p^i, q^i)$$

$$J_D(U_G, d_{true}) = \frac{1}{|U_G|} \sum_{(p^i,q^i,a^i)\in U_G} \log P_{D,d_{true}}(a_i|p^i, q^i)$$

- An adversarial training objective by append the tag d_{true} instead of d_{gen} for model-generated data to fool the discriminative model
- Pretrain with MLE is important

Algorithm 1 Training Generative Domain-Adaptive Nets

Input: labeled data L, unlabeled data U, #iterations T_G and T_D
Initialize G by MLE training on L
Randomly initialize D
while not stopping **do**
  for $t \leftarrow 1$ to T_D **do**
    Update D to maximize $J(L, \text{d\_true}, D) + J(U_G, \text{d\_gen}, D)$ with SGD
  end for
  for $t \leftarrow 1$ to T_G **do**
    Update G to maximize $J(U_G, \text{d\_true}, D)$ with Reinforce and SGD
  end for
end while
return model D

Our Findings

The improvement is small…

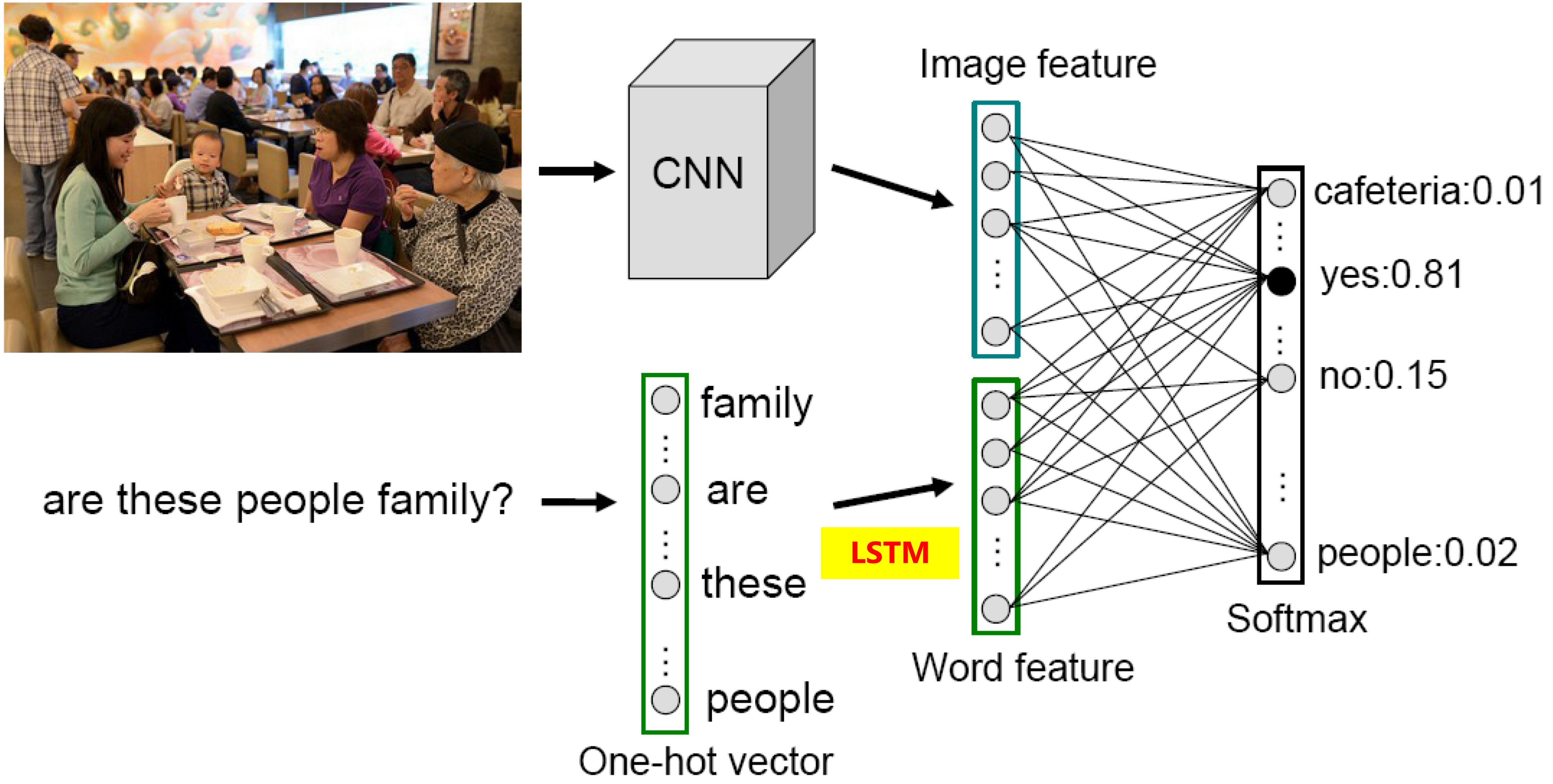
Some (possible) explanations

- Hard to predict whether a QA pair is good
- Hard to tune models like GAN
- QA and QG are two equally-difficult tasks

VisualQA

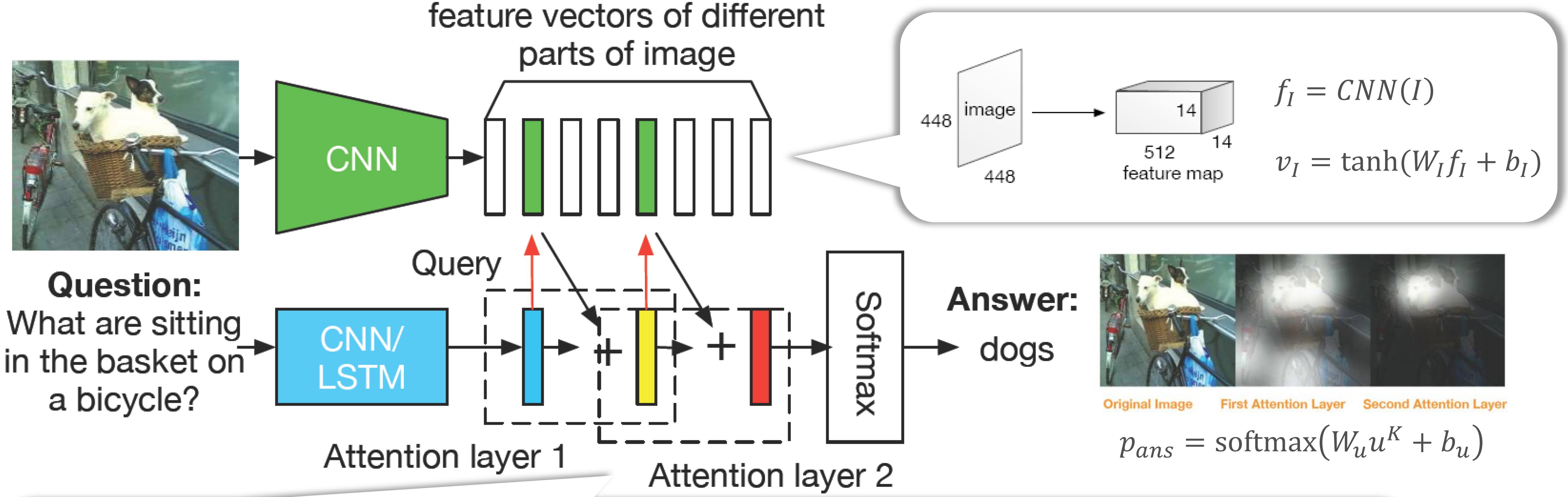
Simple Baseline

(Zhou et al., 2015)



Stacked Attention Networks (SANs)

(Yang et al., 2016)



$$f_I = CNN(I)$$

$$v_I = \tanh(W_I f_I + b_I)$$

$$p_{ans} = \mathrm{softmax}\left(W_u u^K + b_u\right)$$

$$h_A = \tanh\left(W_{I,A} v_I \oplus (W_{Q,A} v_Q + b_A)\right) \in \mathbb{R}^{k \times m}$$

$$p_A = \mathrm{softmax}(W_P h_A + b_P) \in \mathbb{R}^{1 \times m}$$

$$\tilde{v}_I = \sum_i p_i v_i \qquad u = \tilde{v}_I + v_Q$$

...

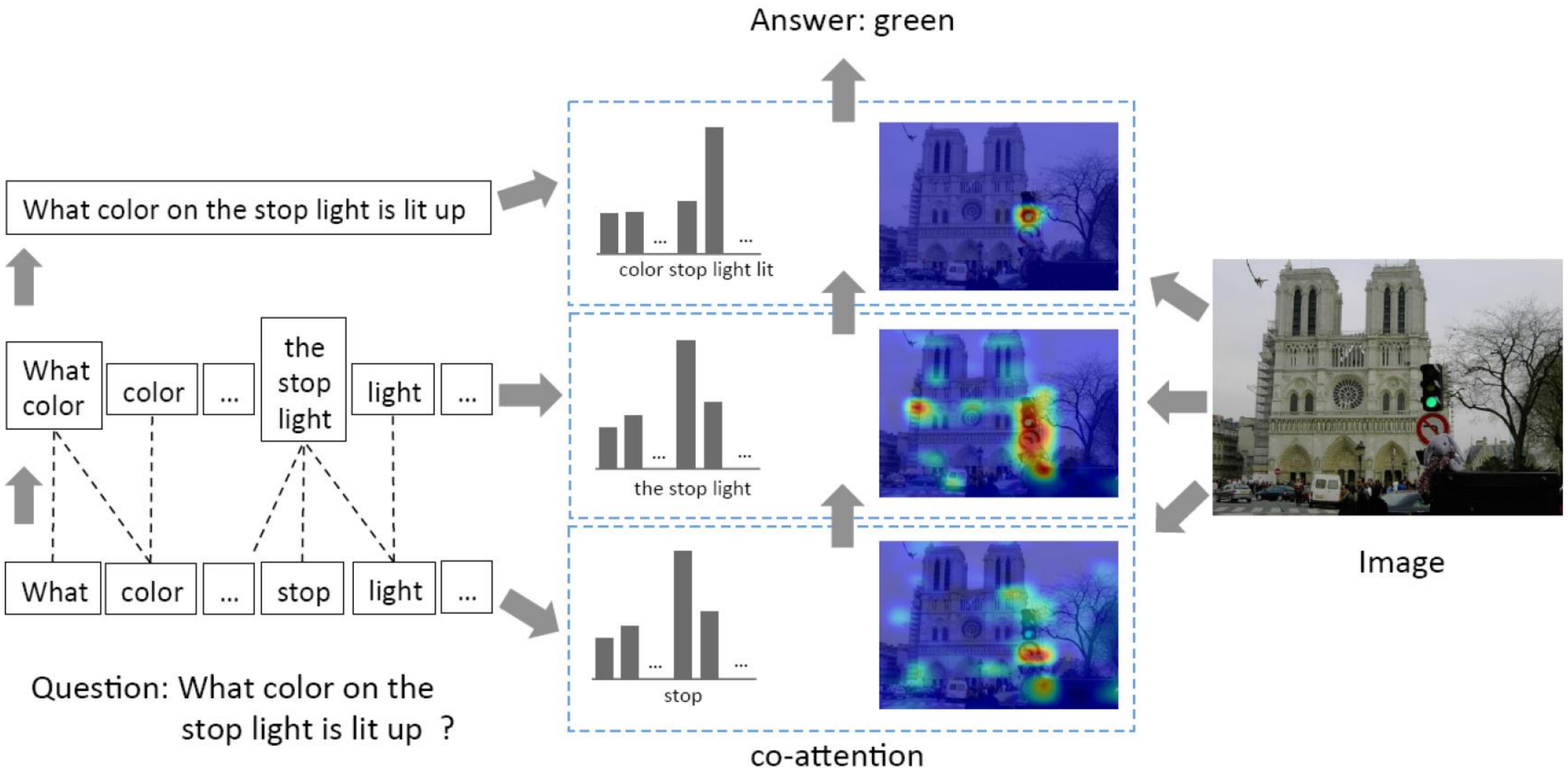
$$h_A^k = \tanh\left(W_{I,A}^k v_I \oplus (W_{Q,A}^k u^{k-1} + b_A^k)\right) \in \mathbb{R}^{k \times m}$$

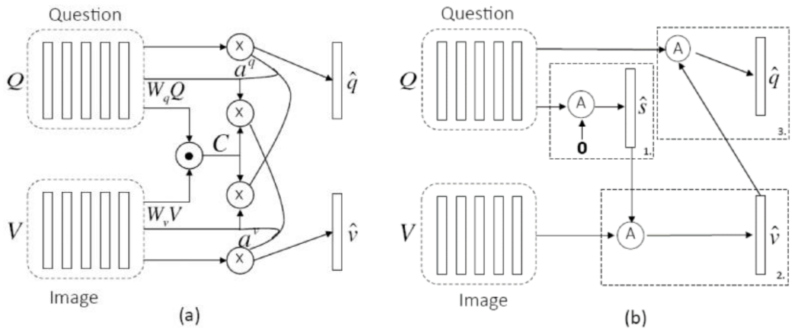
$$p_A^k = \mathrm{softmax}\left(W_P^k h_A^k + b_P^k\right) \in \mathbb{R}^{1 \times m}$$

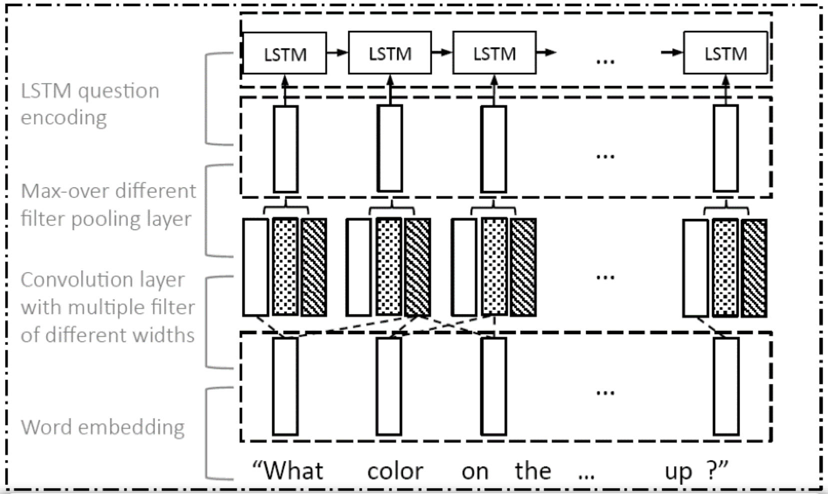
$$\tilde{v}_I^k = \sum_i p_i^k v_i \qquad u^k = \tilde{v}_I^k + u^{k-1}$$

Hierarchical Question-Image Co-Attention

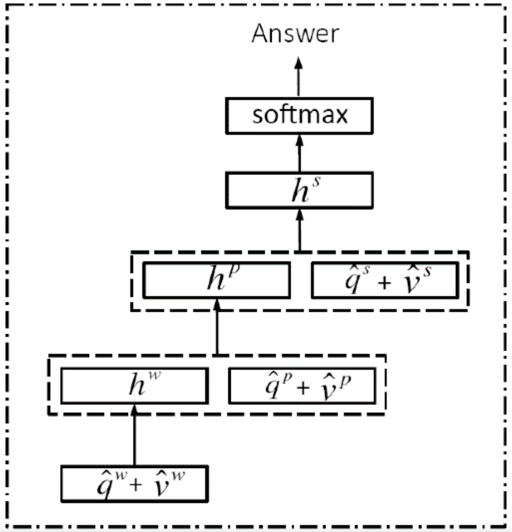
(Lu et al., 2016)







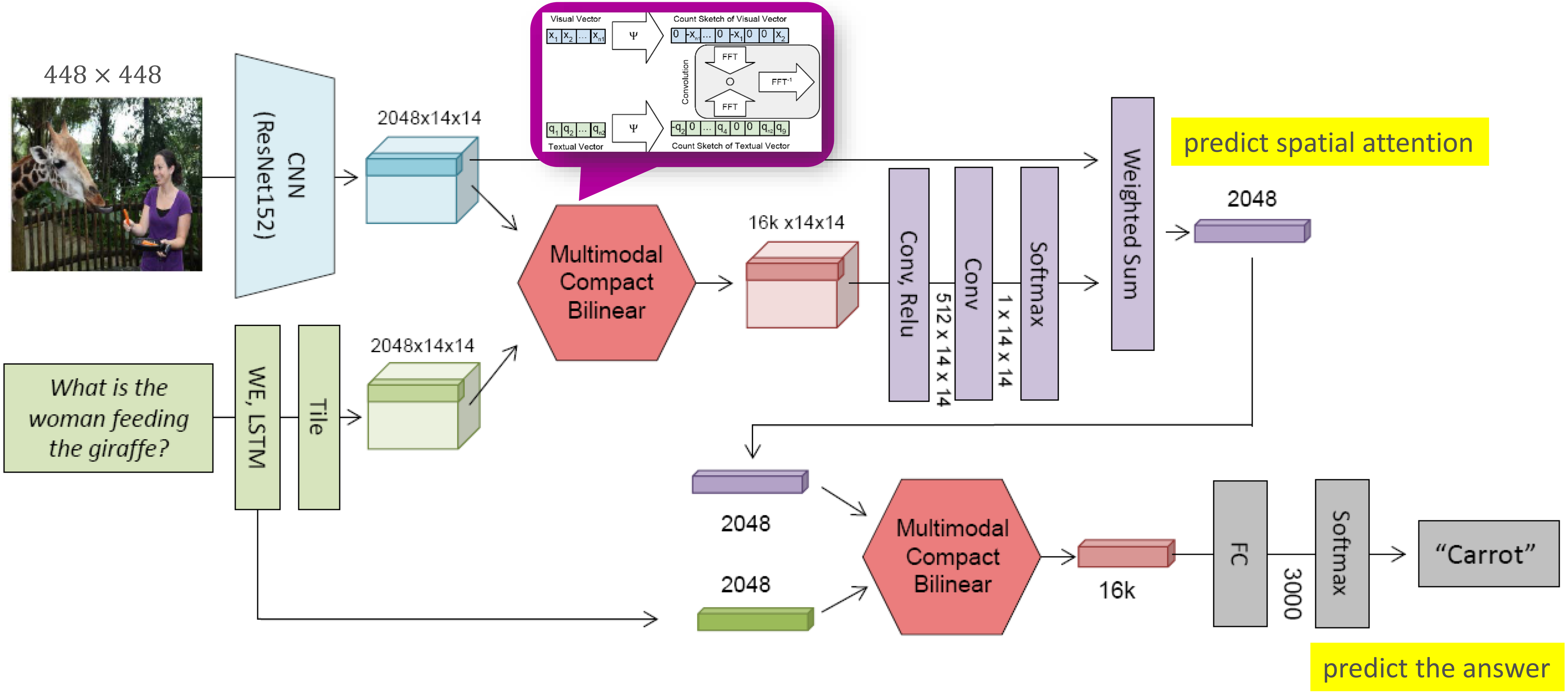
Hierarchical Question Encoding



Encoding for Answer Prediction

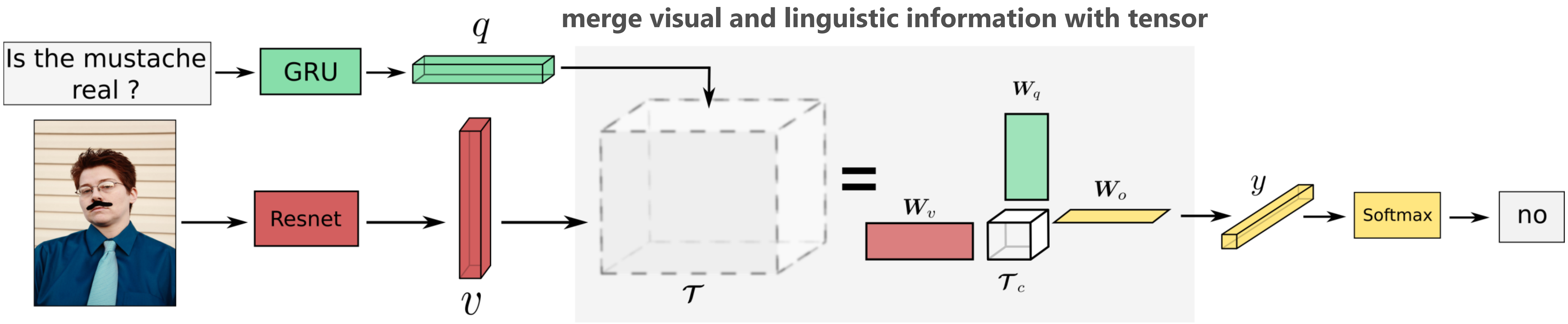
Multimodal Compact Bilinear pooling (MCB)

(Fukui et al., 2016) – Winner of CVPR 2016 VQA Workshop Challenge



Multimodal Tucker Fusion (MUTAN)

(Ben-Younes et al., 2017)



$y = (\mathcal{T} \times_1 q) \times_2 v$ → $\mathcal{T} = \left(\left(\mathcal{T}_c \times_1 W_q\right) \times_2 W_v\right) \times_3 W_o$ → $y = \left(\left(\mathcal{T}_c \times_1 \left(q^{\mathrm{T}} W_q\right)\right) \times_2 \left(v^{\mathrm{T}} W_v\right)\right) \times_3 W_o$

$\mathcal{T} \in \mathbb{R}^{d_q \times d_v \times |\mathcal{A}|}$

$q \in \mathbb{R}^{d_q}$

$v \in \mathbb{R}^{d_v}$

$\mathcal{T}_c \in \mathbb{R}^{t_q \times t_v \times t_o}$

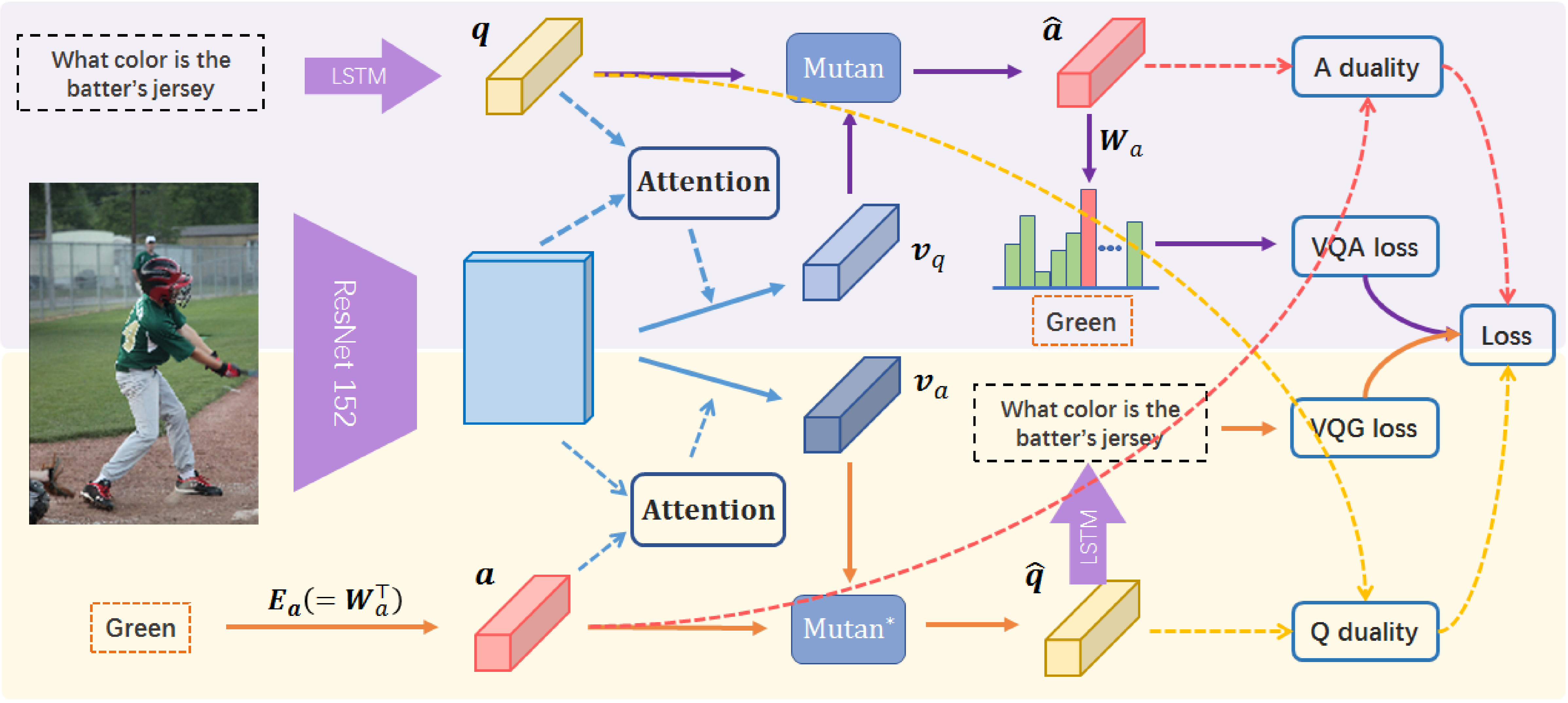
$W_q \in \mathbb{R}^{d_q \times t_q}$

$W_v \in \mathbb{R}^{d_v \times t_v}$

$W_v \in \mathbb{R}^{|\mathcal{A}| \times t_o}$

Visual QG as Dual Task of Visual QA

(Li et al., 2017)



Reference

- Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, Rob Fergus. **Simple Baseline for Visual Question Answering**. arXiv, 2015.
- Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Smola. **Stacked Attention Networks for Image Question Answering**. CVPR, 2016.
- Jiasen Lu, Jianwei Yang, Dhruv Batray, Devi Parikh. **Hierarchical Question-Image Co-Attention for Visual Question Answering**. NIPS, 2016.
- Hedi Ben-Younes, R´emi Cadene, Matthieu Cord, Nicolas Thome. **MUTAN: Multimodal Tucker Fusion for Visual Question Answering**. CVPR, 2017.
- Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang. **Visual Question Generation as Dual Task of Visual Question Answering**. arXiv, 2017.