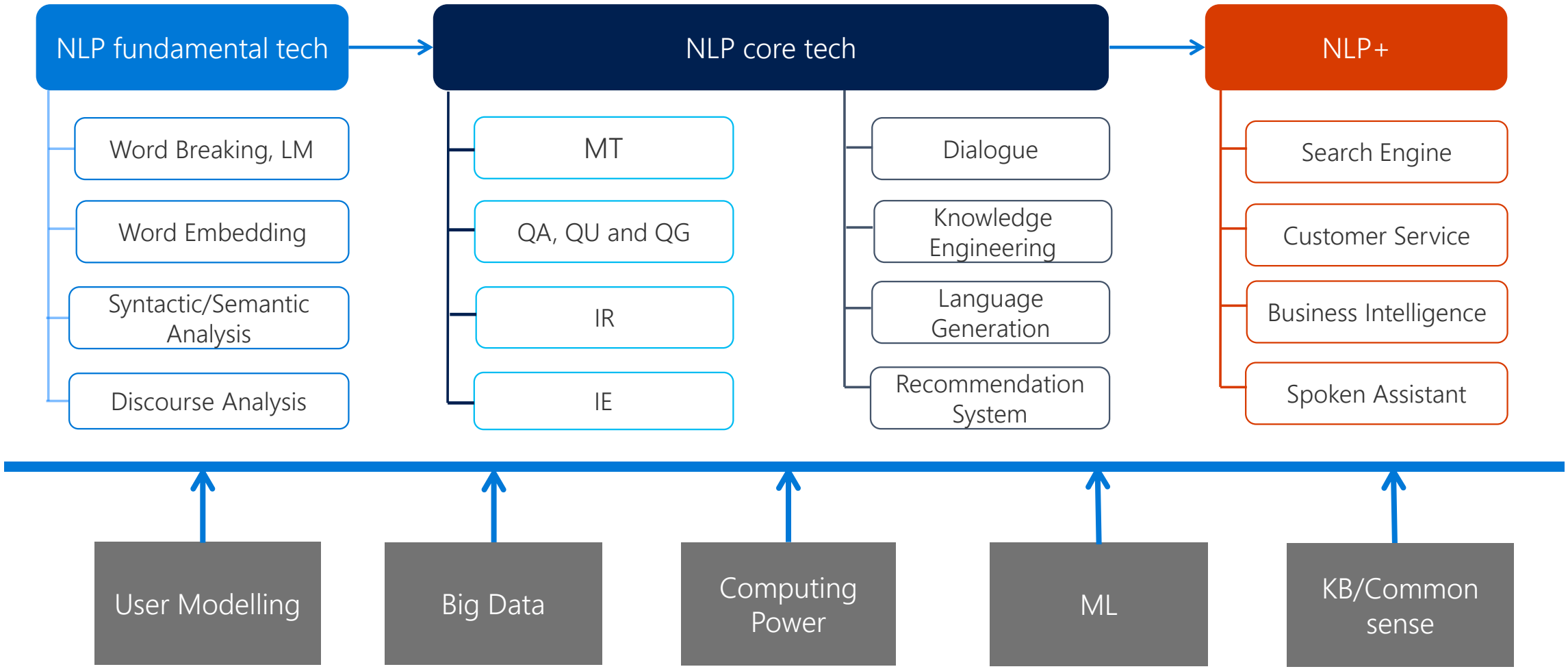
Knowledge-enhanced NLP: Progress and Trend

Nan Duan
Microsoft Research Asia
2018-11-17

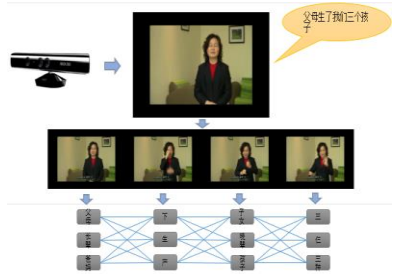
Microsoft

NLP Technologies

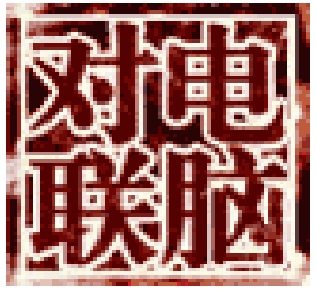


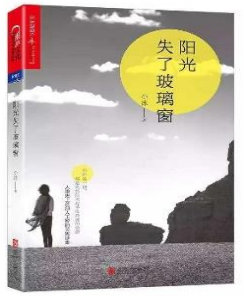
Slides from Dr. Ming Zhou

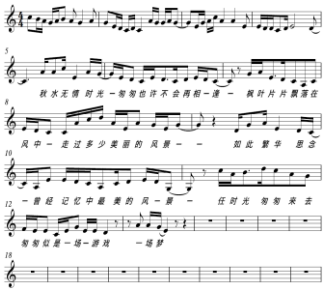
MSRA NLP Research Achievements



Bing







Slides from Dr. Ming Zhou

Importance of Knowledge to NLP

- Supervised learning with enough training data
- DNN works successfully

Multi-turn
(rich-context)

Single-turn
(low-context)

Conversational AIs
(task-oriented)

Conversational AIs
(chit-chat)

QA

Semantic Parsing

Machine
Translation

Search Engine

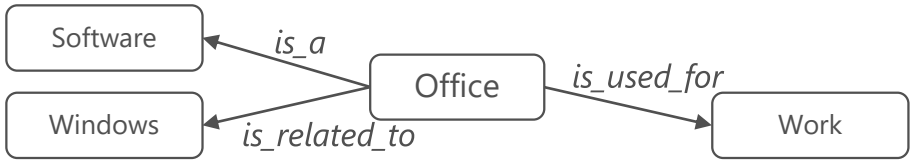
- Lack of training data
- Knowledge is a must

Low-resource

Rich-resource

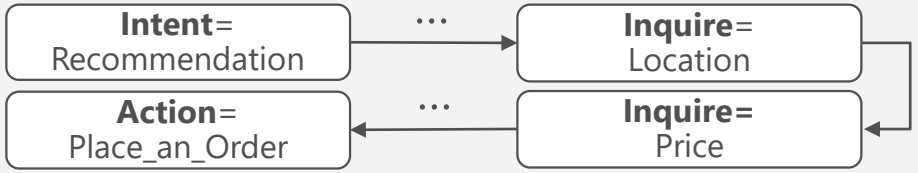
Knowledge Pyramid in NLP

Commonsense



ConceptNet

Knowledge Base (Specific Domain)



Knowledge Base (Open Domain)

Microsoft Office	**Developer**	Microsoft
Microsoft Office	**Initial release**	19 November 1990
Microsoft Office	**Operating system**	Microsoft Windows
Microsoft Office	**Website**	office.com
...	...	...

Supervised Pre-trained Embeddings

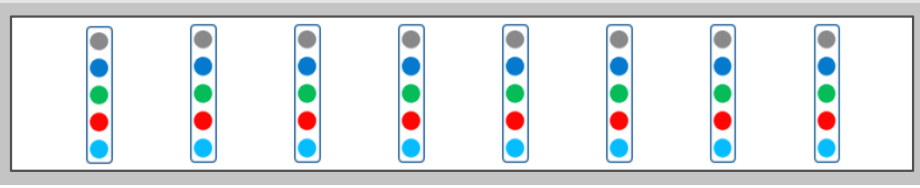
MT Model

QA Model

MRC Model

...

Unsupervised Pre-trained Embeddings



which version of Office is right for student

Today's Agenda

- Deep Learning Basics for NLP
- NLP with Pre-trained Embeddings
- NLP with Knowledge Bases
- NLP with Commonsense
- Summary and Trend

Today's Agenda

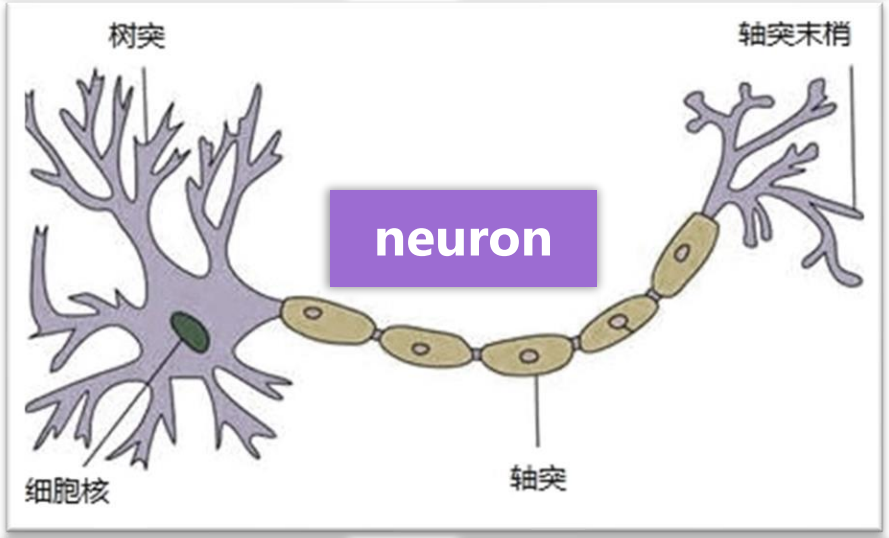
- **Deep Learning Basics for NLP**
- NLP with Pre-trained Embeddings
- NLP with Knowledge Bases
- NLP with Commonsense
- Summary and Trend

Neuron (神经元)

(McCulloch and Pitts, 1943)

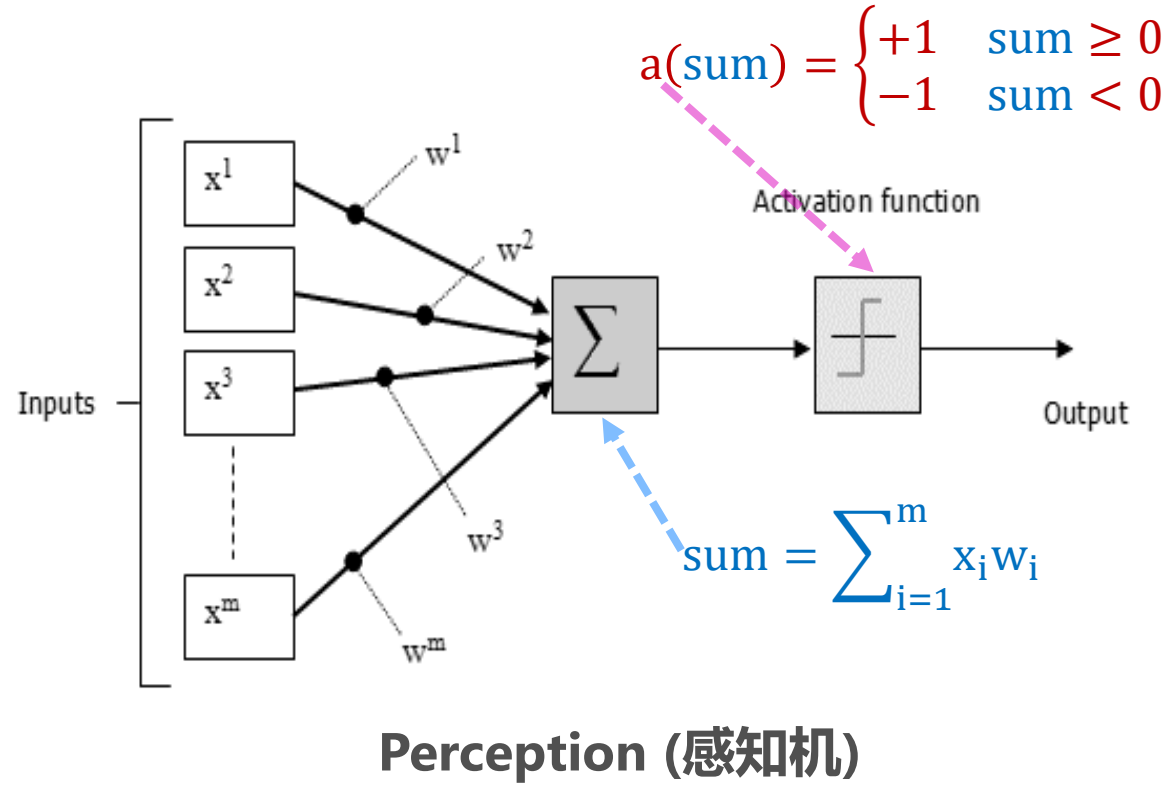
accept input

connected with other neurons



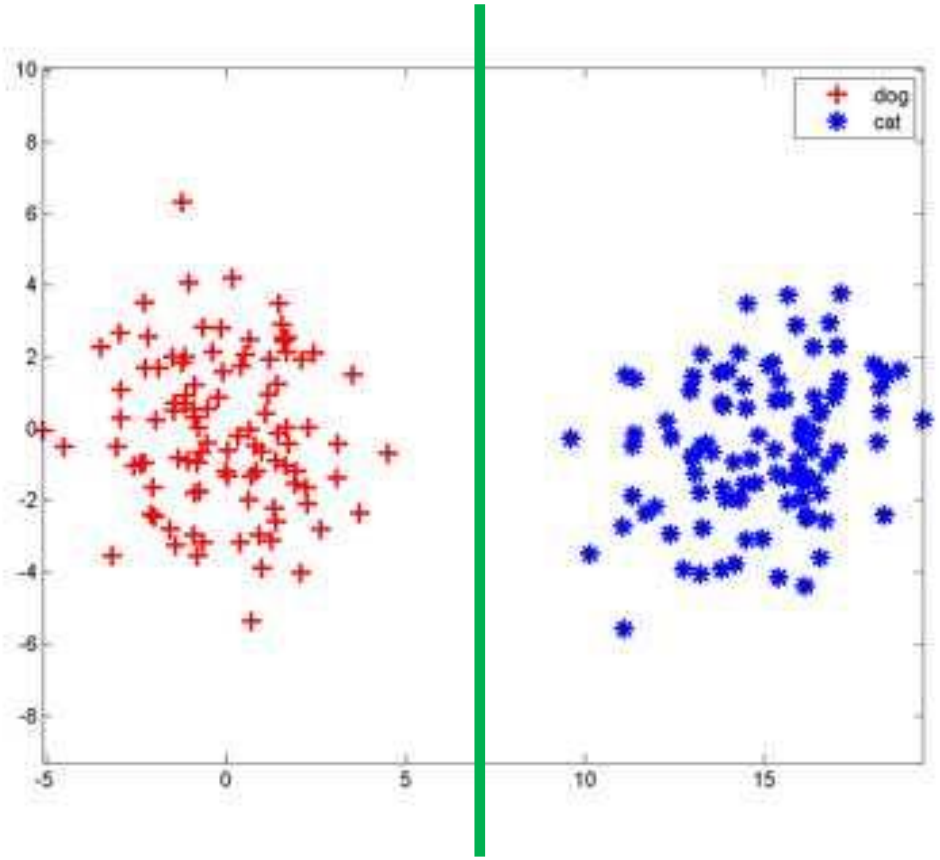
compute output

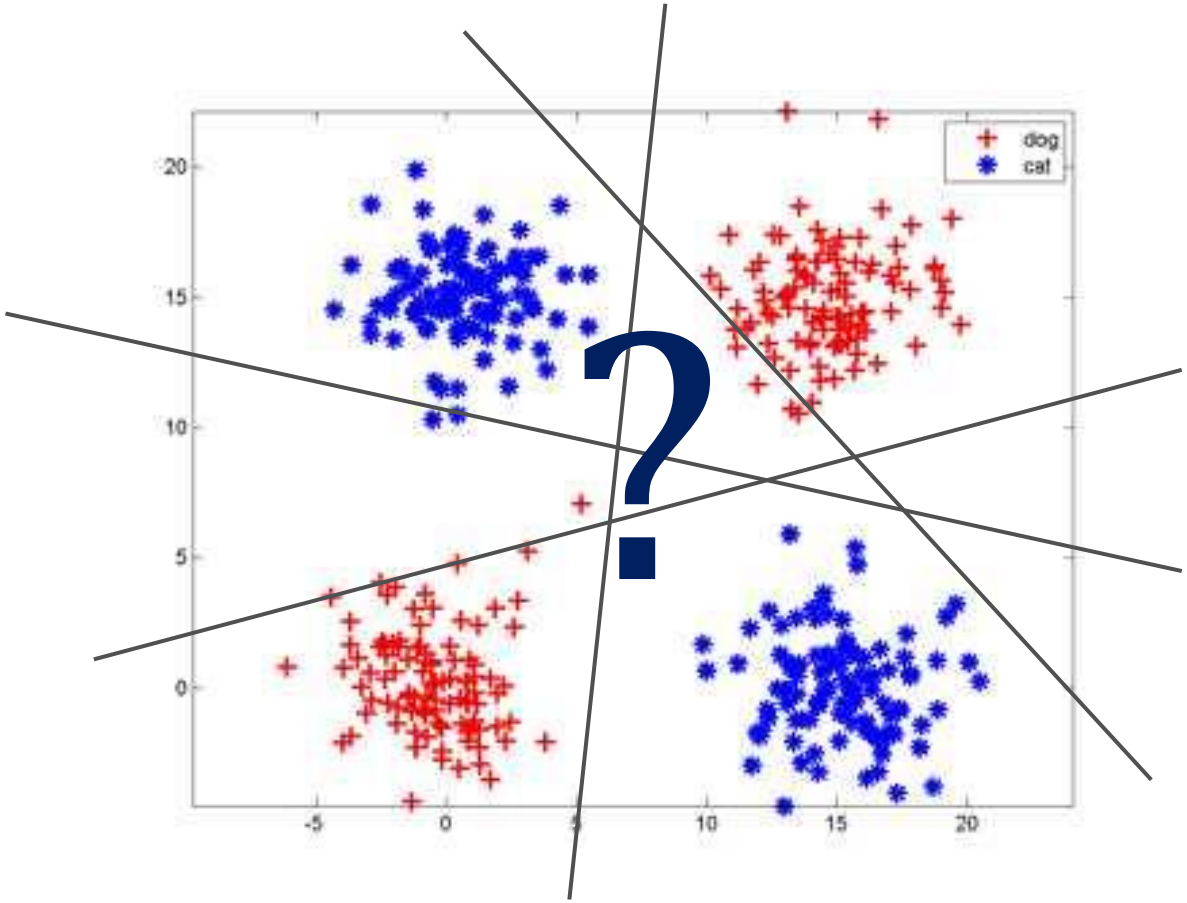
transfer output, if activated



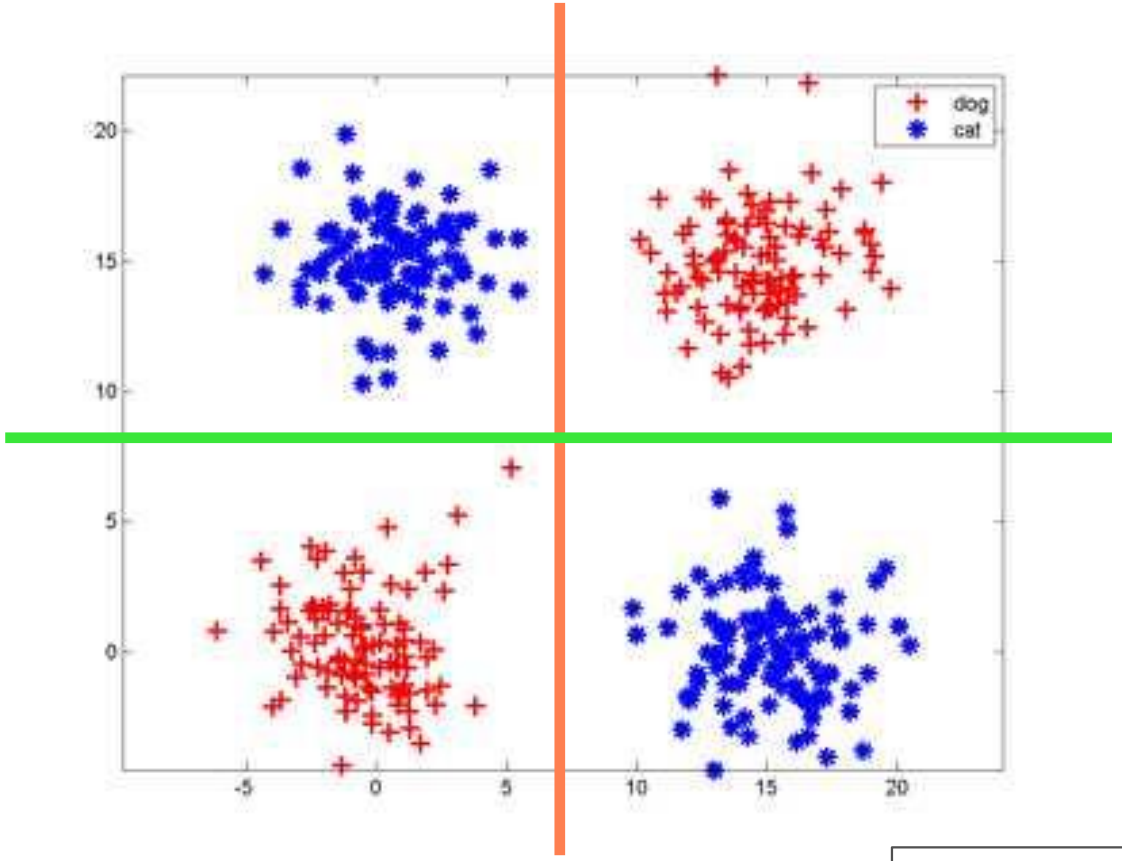
Perception (感知机)

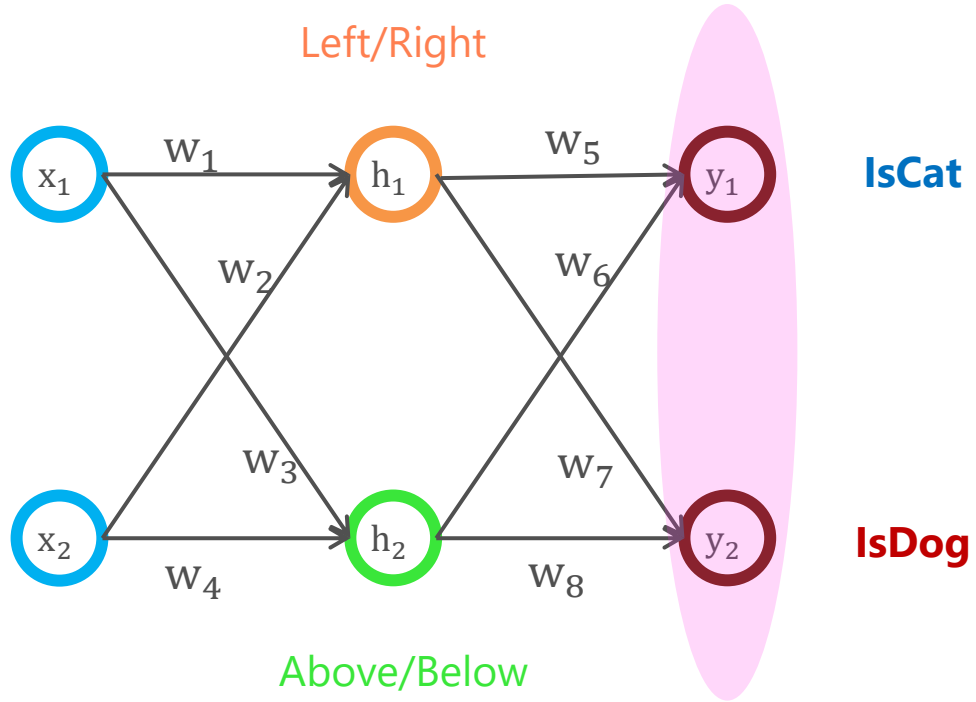
Separate Cats and Dogs with Neuron





Neural Network (神经网络)





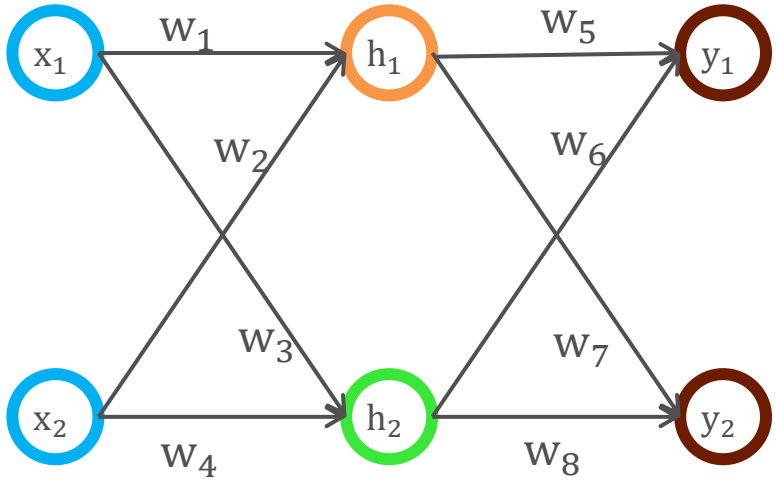
softmax function squashes a K-dimensional vector Z of arbitrary real values to a K-dimensional vector σ(Z) of real values in the range [0, 1] that add up to 1.

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

Feed Forward

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

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



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

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

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

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

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

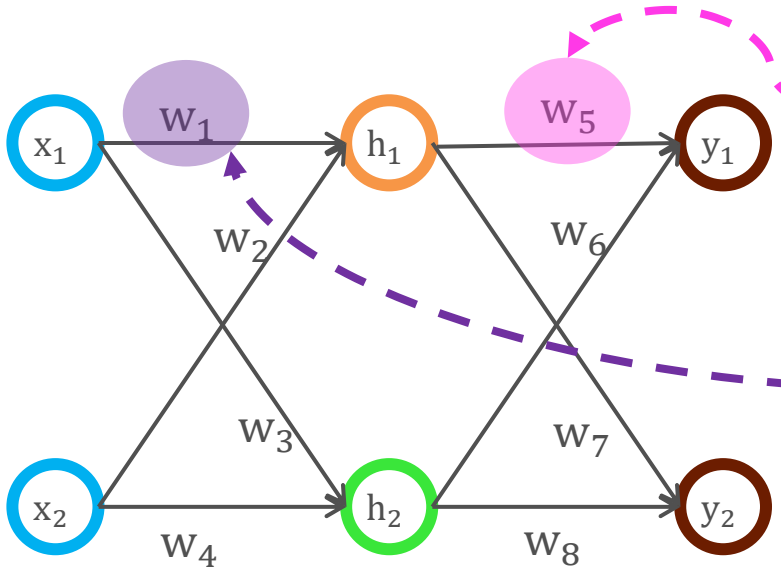
$$\text{out}_{y2} = \frac{1}{1 + e^{-\text{net}_{y2}}} \neq \text{target}_{y2}$$

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

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

$$E = \frac{1}{2}(target_{y1} - out_{y1})^2 + \frac{1}{2}(target_{y2} - out_{y2})^2$$

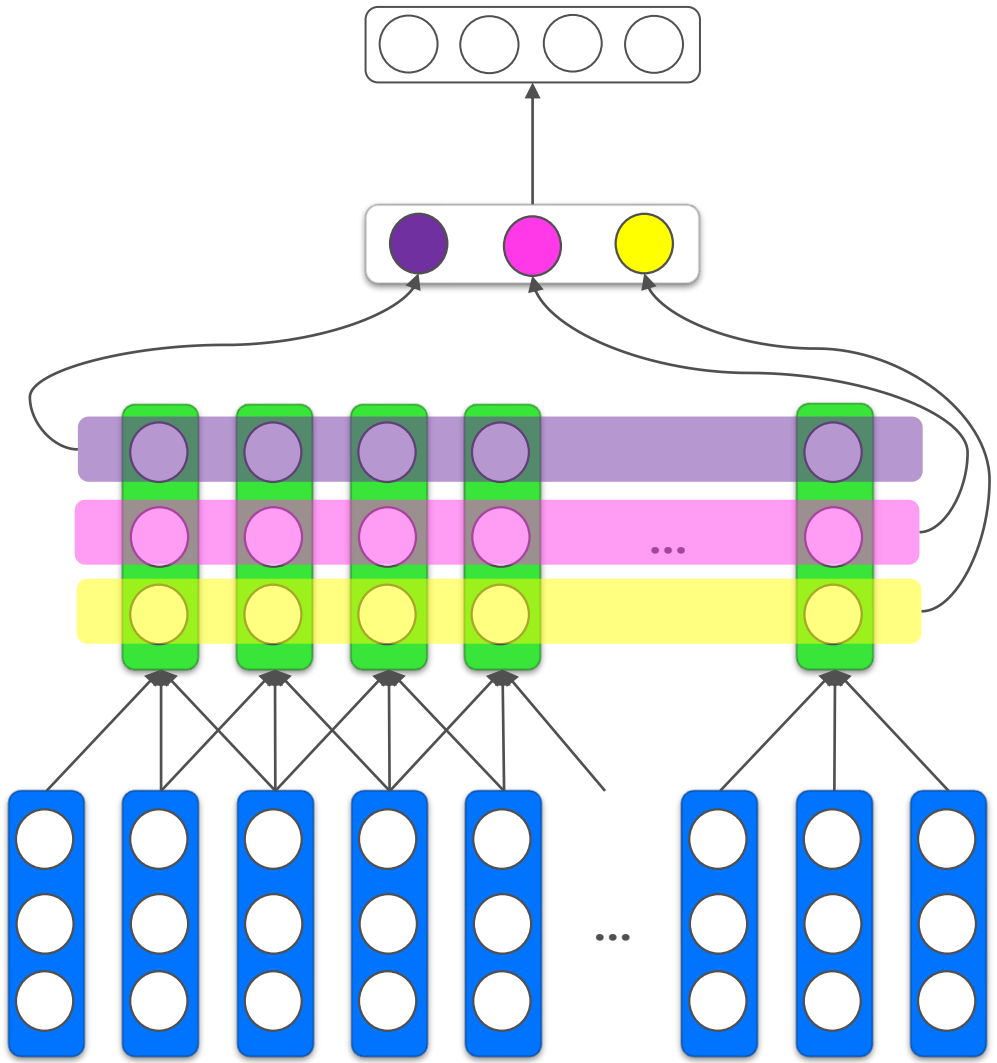


$$\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial out_{y1}} * \frac{\partial out_{y1}}{\partial net_{y1}} * \frac{\partial net_{y1}}{\partial w_5}$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

Chain Rule

Convolution Neural Network (CNN)



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

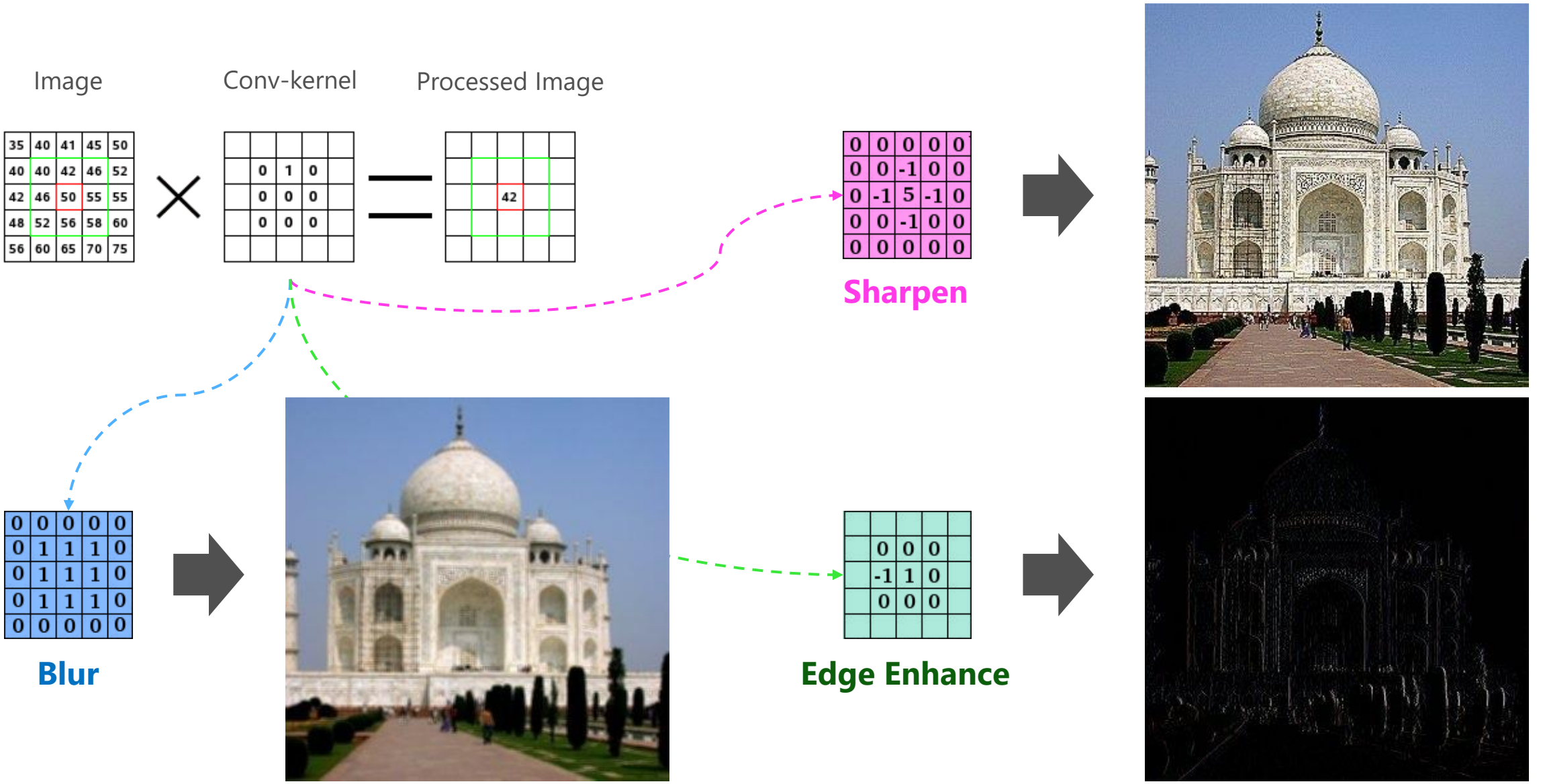
$$c_i = f(W_C \cdot x_{i:i+d-1} + b_C)$$

Local Receptive Field

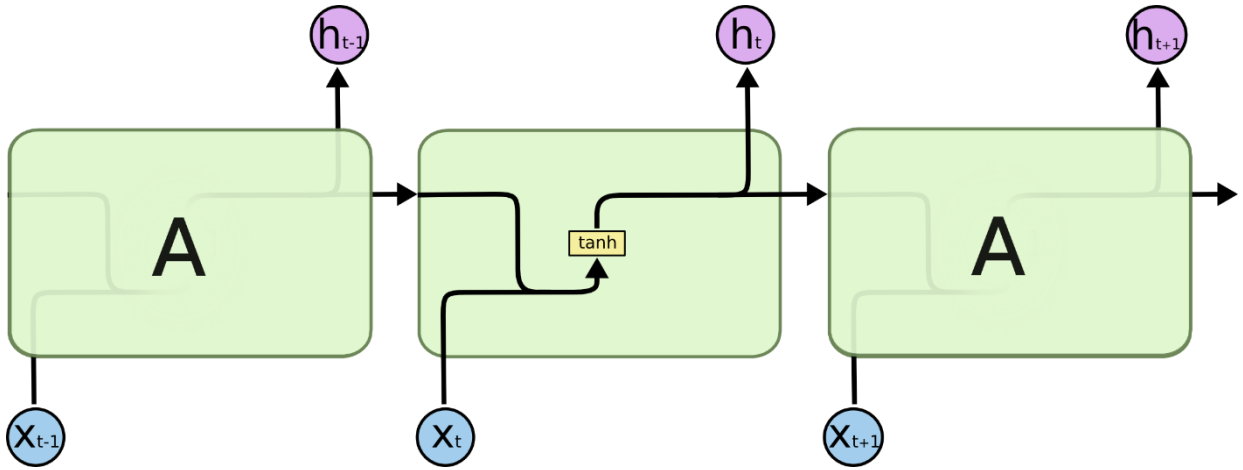
Weight Sharing

Input Layer

Convolution is Feature Extraction

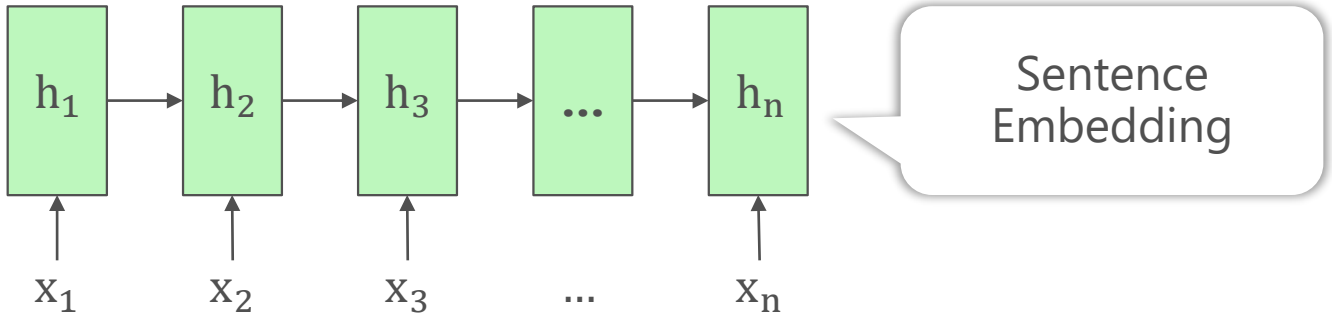


Recurrent Neural Network (RNN)

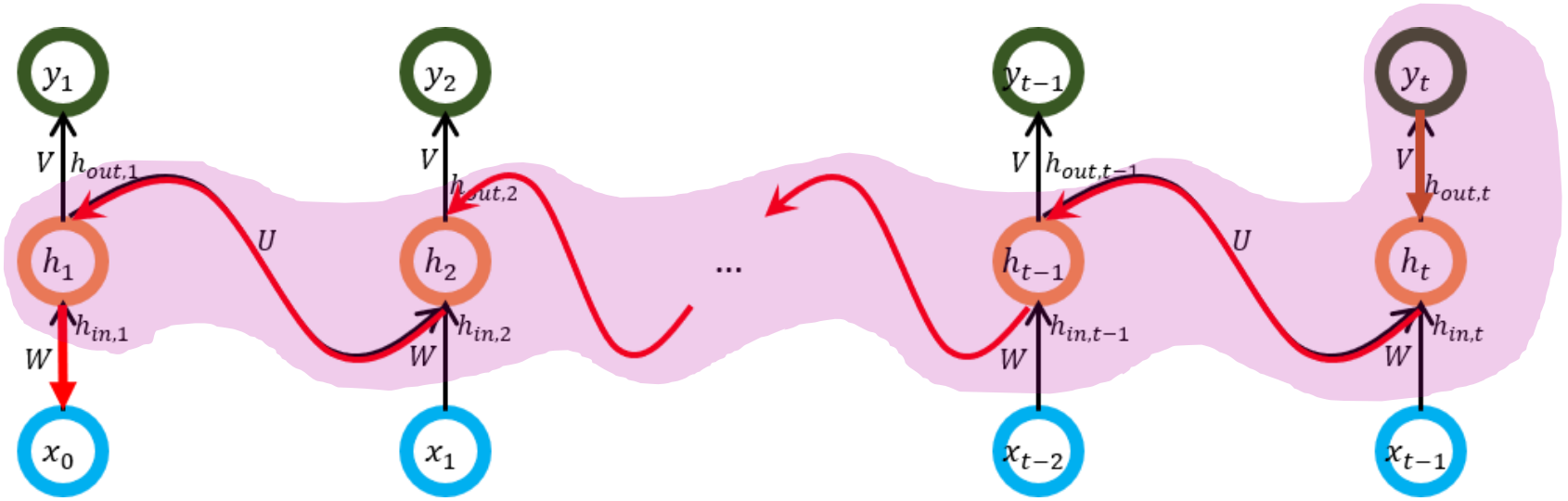


$$h_t = \tanh(W_h \cdot h_{t-1} + W_x \cdot x_t + b) = \tanh(W \cdot [h_{t-1}; x_t] + b)$$

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

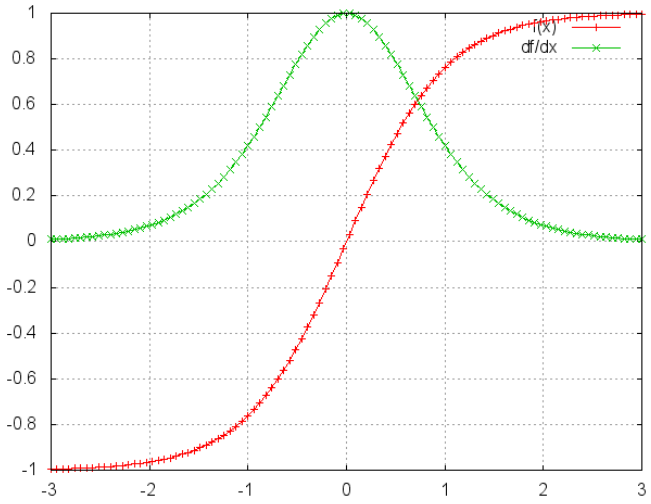


Vanishing Gradients

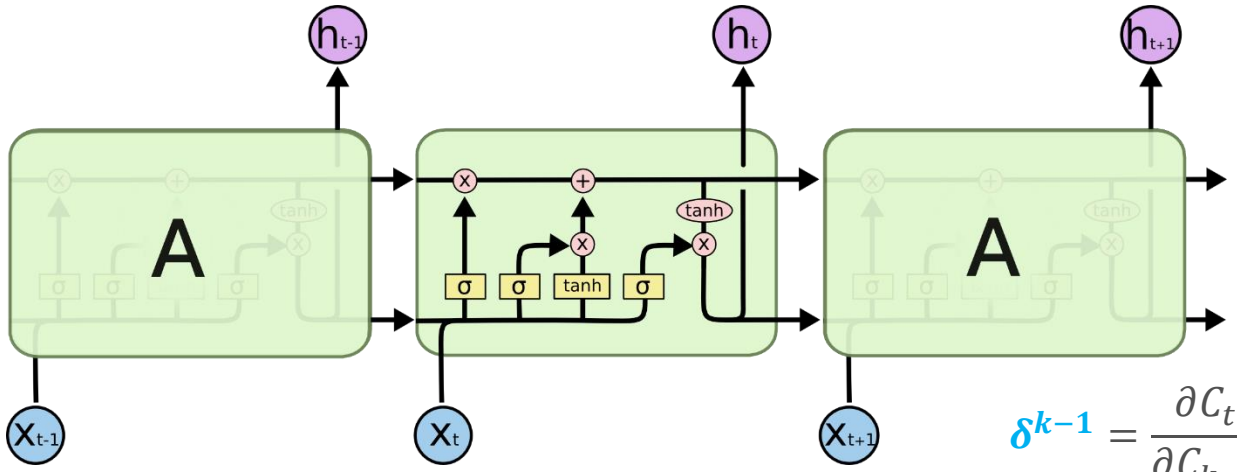


$$\delta_{in,1} = \delta_{out,t} * \frac{\partial h_{out,t}}{\partial h_{in,t}} * \frac{\partial h_{in,t}}{\partial h_{out,t-1}} * \cdots * \frac{\partial h_{in,2}}{\partial h_{out,1}} * \frac{\partial h_{out,1}}{\partial h_{in,1}}$$

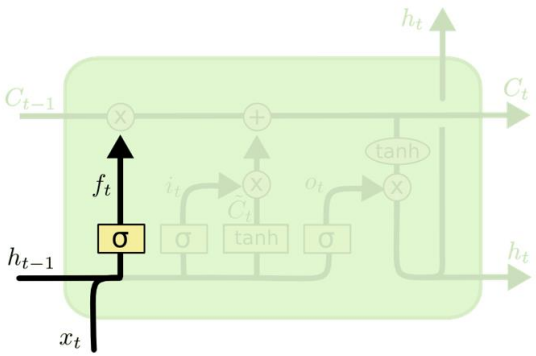
Value range of
$\tanh(x)$ and $\tanh'(x)$



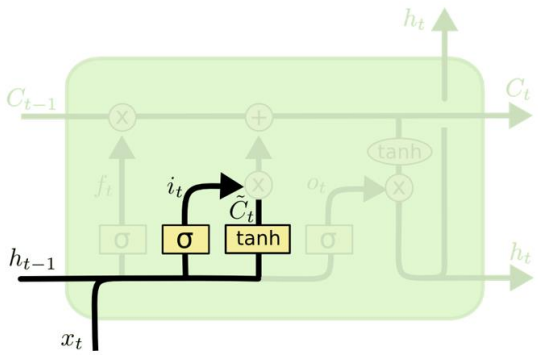
Long Short-Term Memory (LSTM)



$$\delta^{k-1} = \frac{\partial C_t}{\partial C_{k-1}} = \frac{\partial C_t}{\partial C_k} * \frac{\partial C_k}{\partial C_{k-1}} = \delta^k * \frac{\partial C_k}{\partial C_{k-1}}$$
$$= \delta^k * (f_k + \cdots)$$

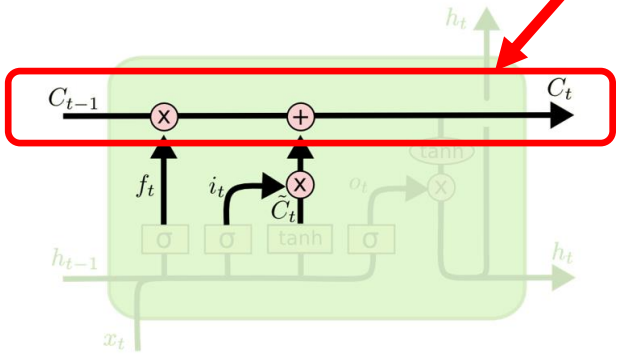


$$f_t = \sigma(W_f \cdot [h_{t-1}; x_t] + b_f)$$

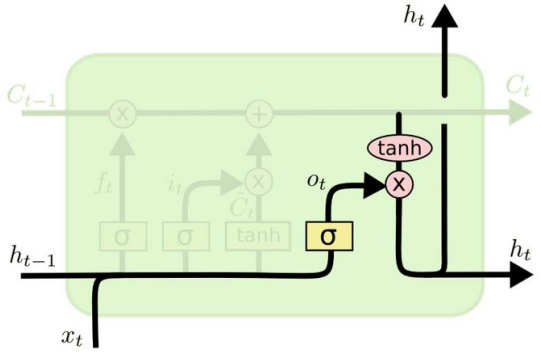


$$i_t = \sigma(W_i \cdot [h_{t-1}; x_t] + b_i)$$

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



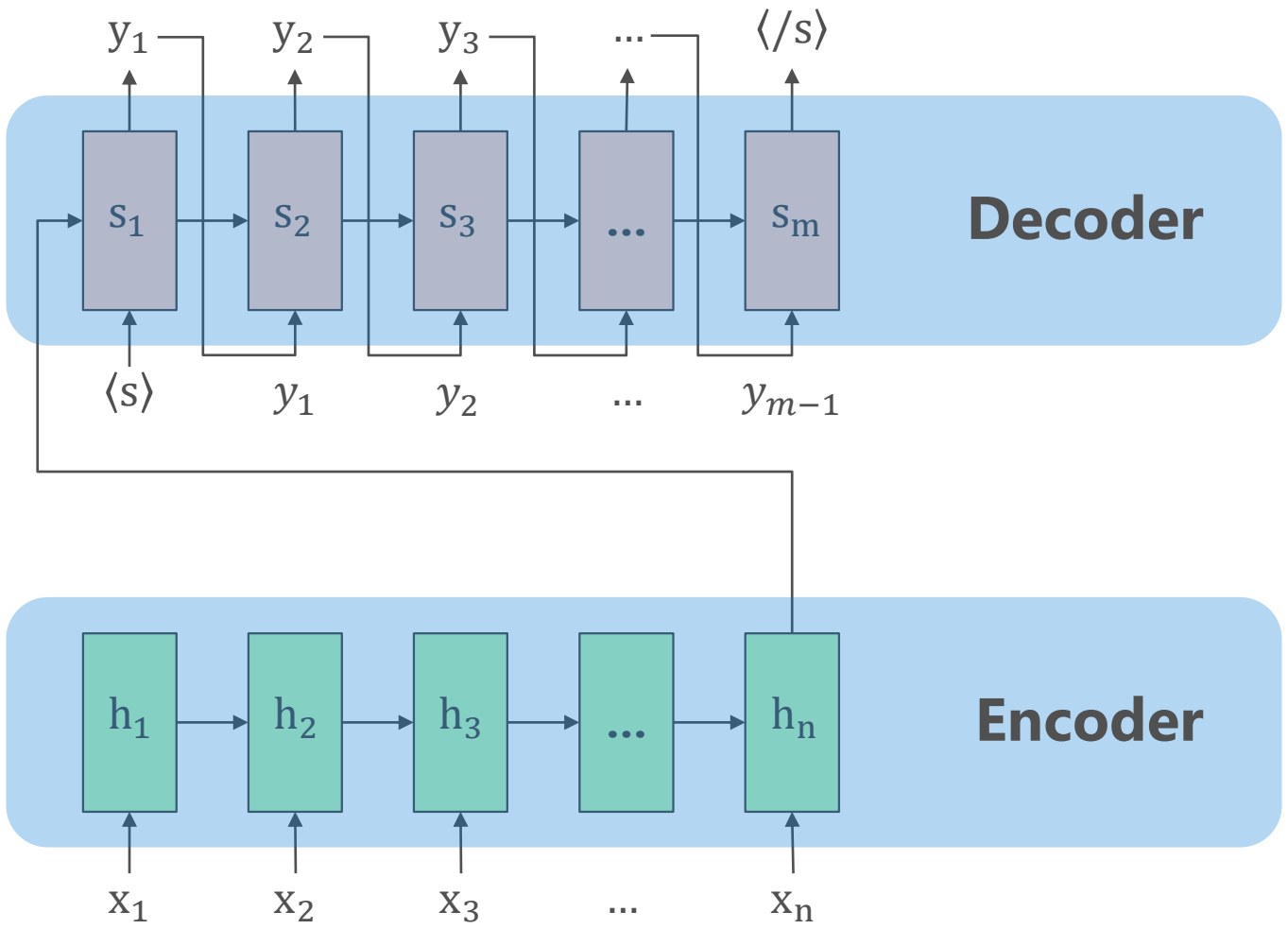
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$



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

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

Encoder-Decoder Framework for Sequence Generation

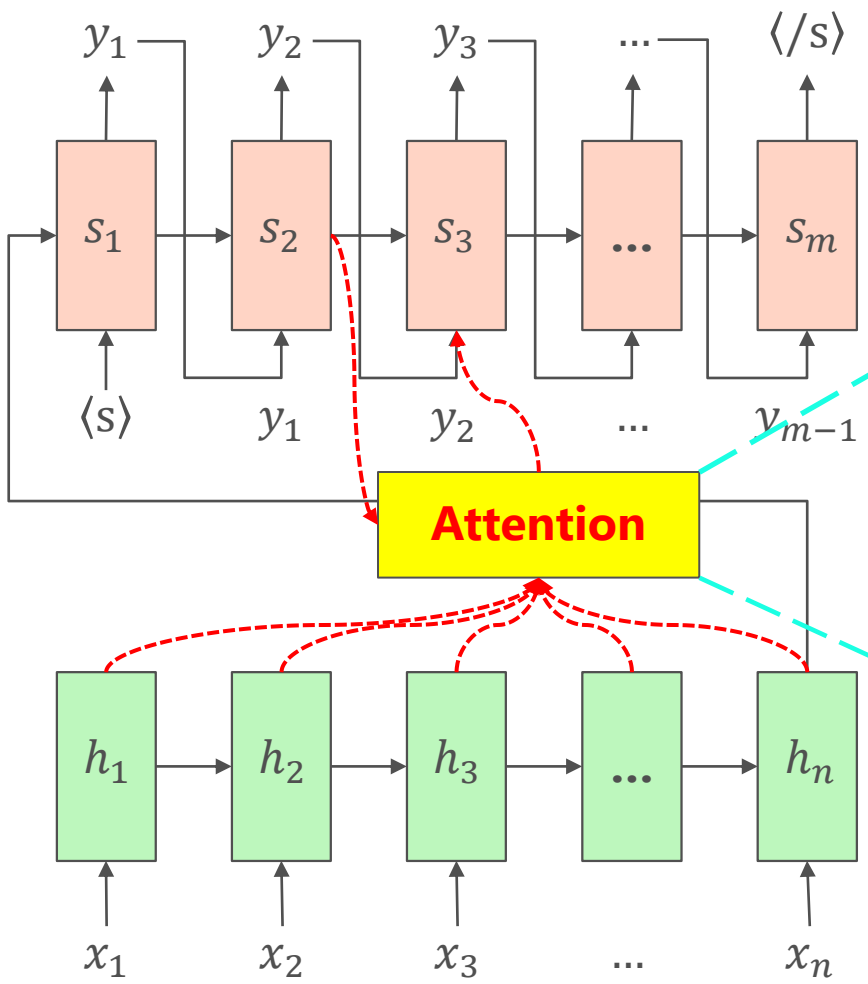


$$s_i = \tanh(W \cdot y_{i-1} + U \cdot s_{i-1})$$

Encoder-Decoder Framework with Attention

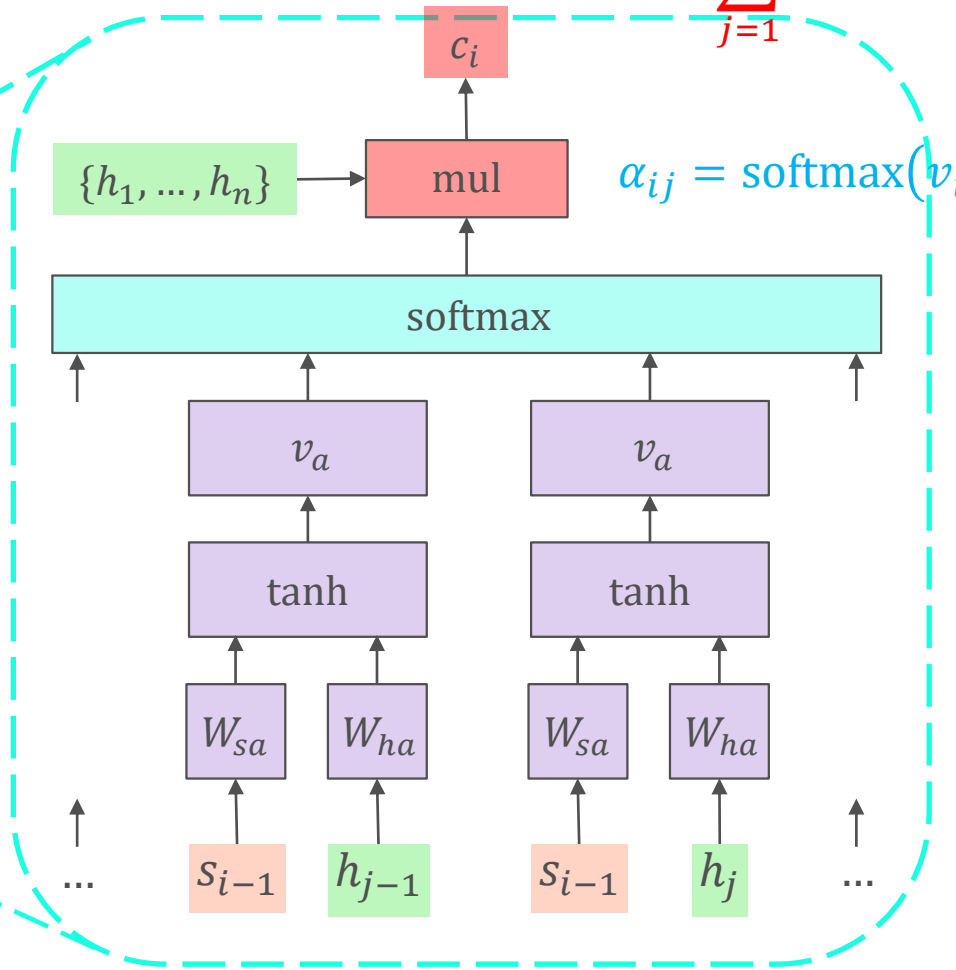
$$s_i = \tanh(W \cdot y_{i-1} + U \cdot s_{i-1} + \boldsymbol{C \cdot c_i})$$

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



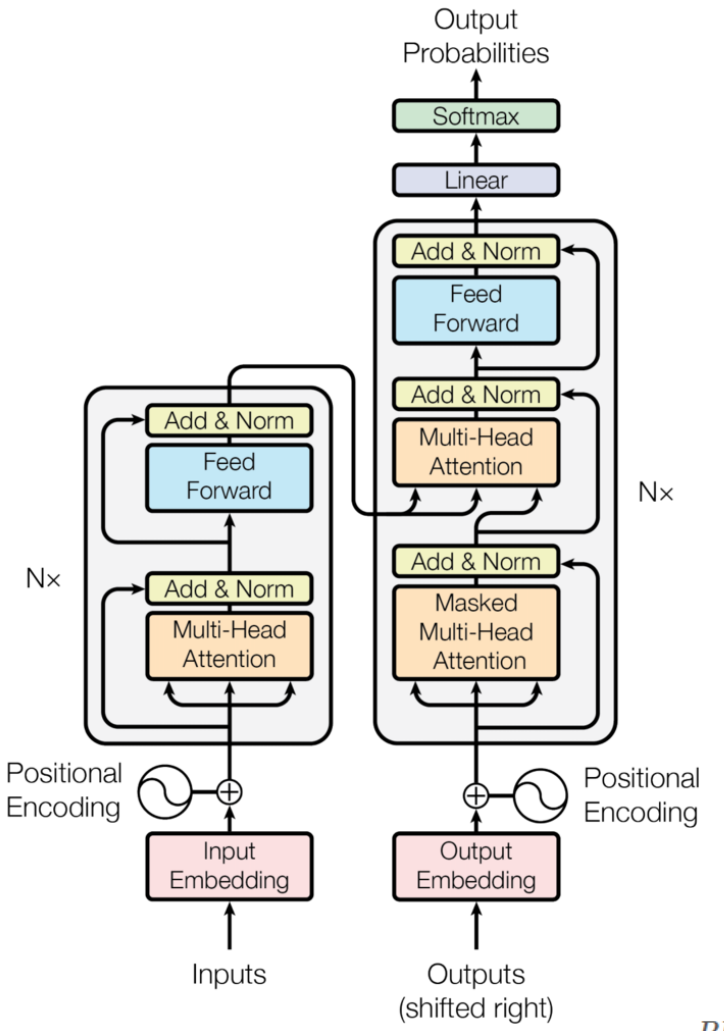
Decoder

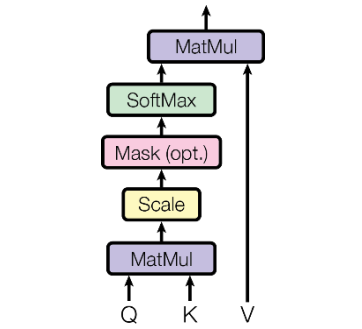
Encoder

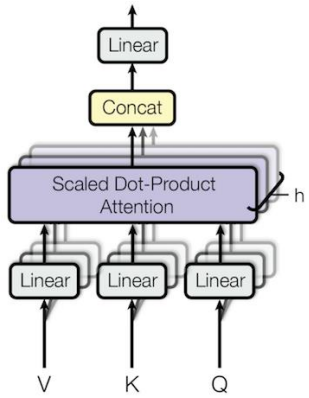


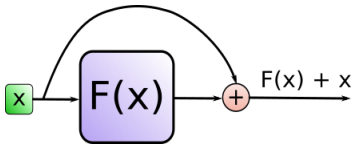
$$v_{ij} = v_a^T \tanh\left(W_{sa} s_{i-1} + W_{ha} h_j\right)$$

Transformer









Transformer

$$PE(pos, 2i) = \sin(pos/10^{4 \cdot 2i/d_{model}})$$

$$PE(pos, 2i+1) = \cos(pos/10^{4 \cdot 2i/d_{model}})$$

Today's Agenda

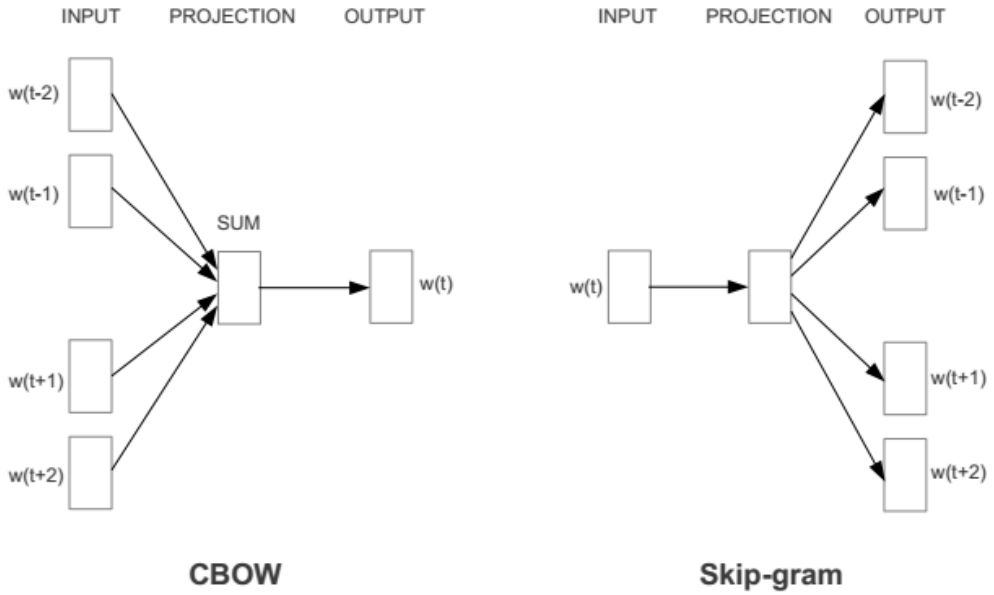
- Deep Learning Basics for NLP
- **NLP with Pre-trained Embeddings**
- NLP with Knowledge Bases
- NLP with Commonsense
- Summary and Trend

Bag-of-words

- A text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

Word2Vec

(Mikolov et al., 2013)



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

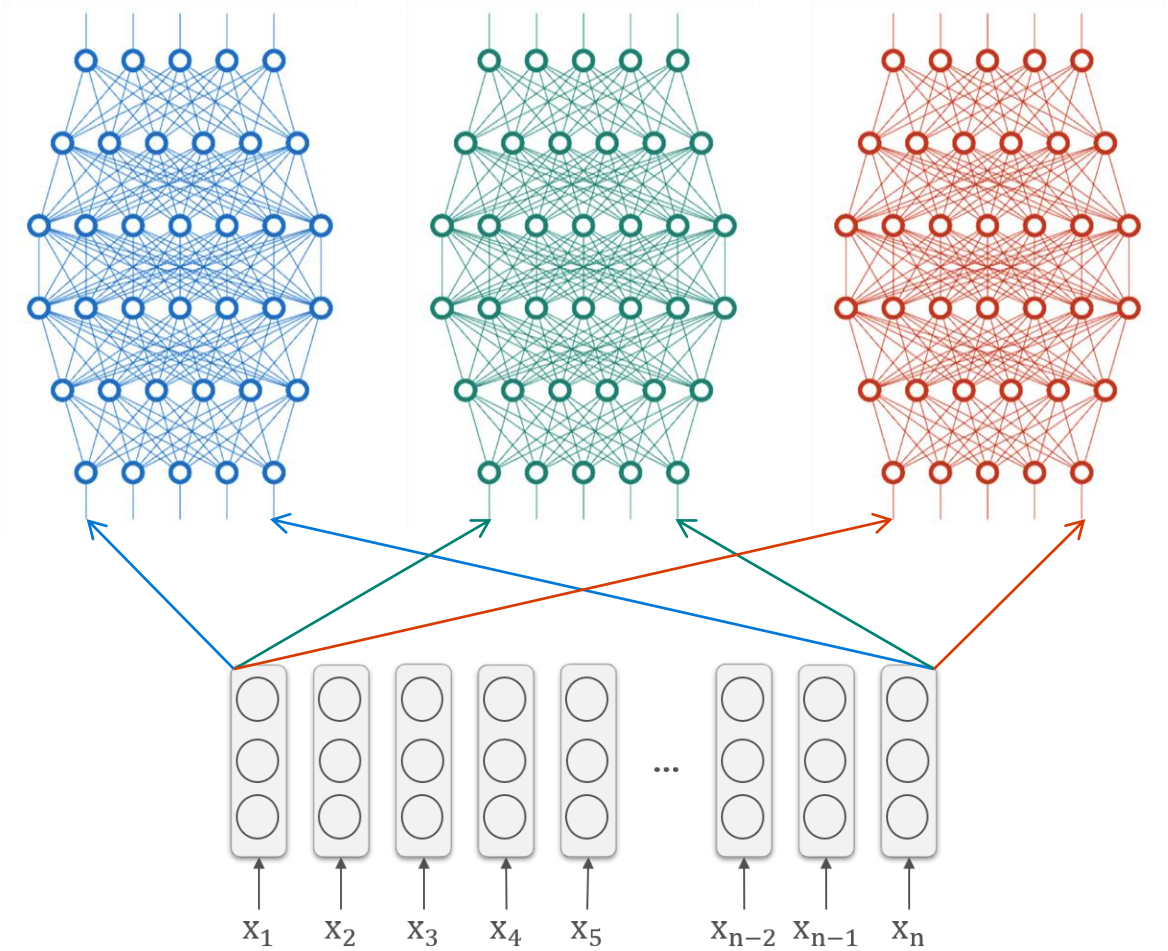
Objective function of CBOW: minimize $-\frac{1}{T}\sum_{t=1}^{T}\log p^{CBOW}(w_t|w_{t-n},\dots,w_{t-1},w_{t+1},\dots,w_{t+n})$

Objective function of Skip-gram: minimize $-\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c,j\neq 0}\log p^{Skip-gram}(w_{t+j}|w_t)$

Word2Vec in NLP

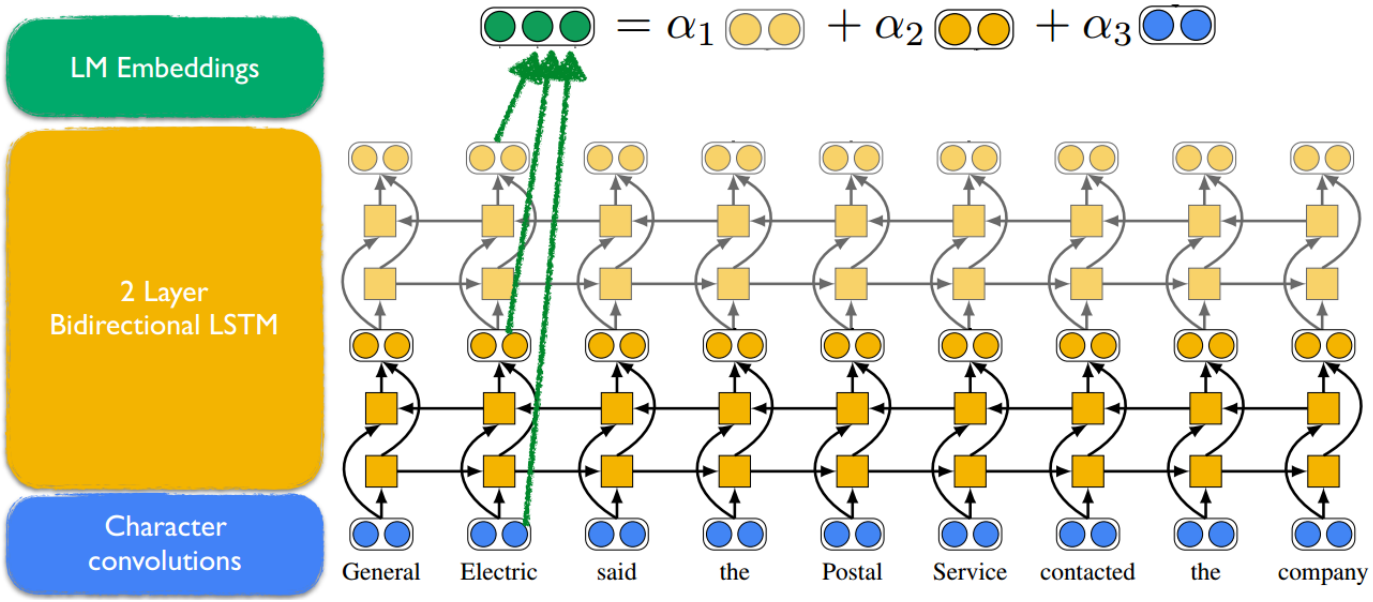
(Mikolov et al., 2013)

- Initialize the first layer of a neural network



ELMo

(Peters et al., 2018)



Forward LM: $p(t_1, \dots, t_N) = \prod_{k=1}^{N} p(t_k | t_1, \dots, t_{k-1})$

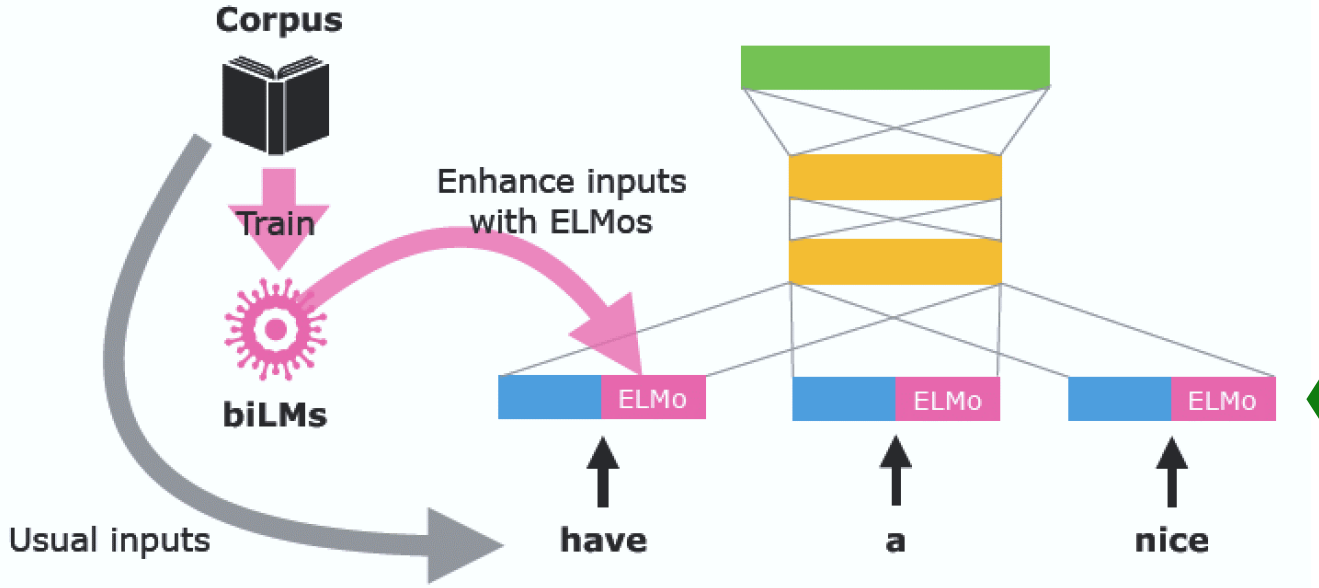
Backward LM: $p(t_1, \dots, t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, \dots, t_N)$

Objective function: maximize $\sum_{t=1}^{T} (\log p(t_k | t_1, \dots, t_{k-1}; \Theta_x, \overrightarrow{\Theta_{LSTM}}, \Theta_{softmax}) + \log p(t_k | t_{k+1}, \dots, t_N; \Theta_x, \overleftarrow{\Theta_{LSTM}}, \Theta_{softmax}))$

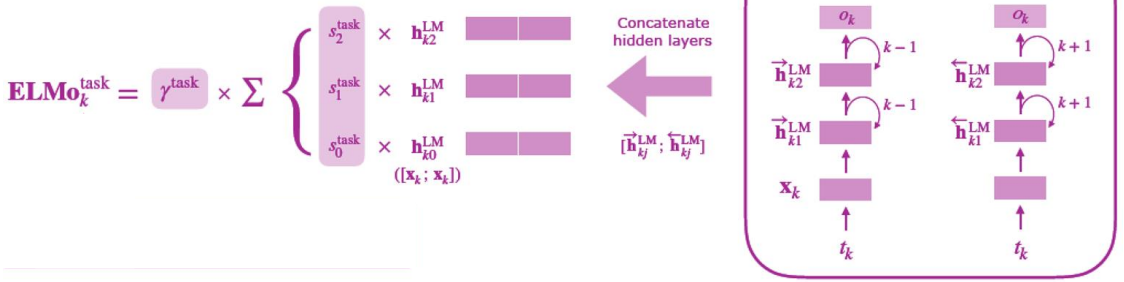
ELMo in NLP

(Peters et al., 2018)

ELMo can be integrated to almost all neural NLP tasks with simple concatenation to the embedding layer

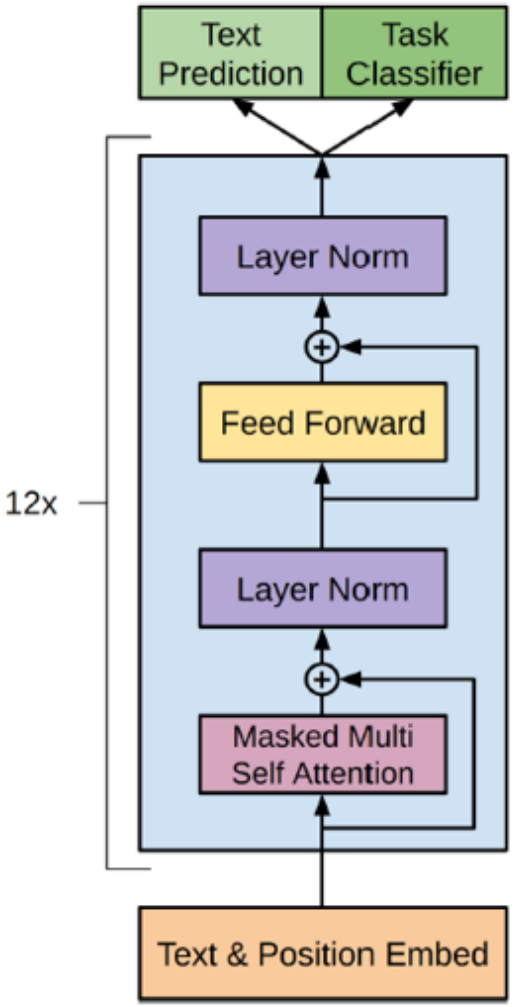


Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%



OpenAI GPT

(Radford et al., 2018)



$$h_0 = UW_e + W_p$$

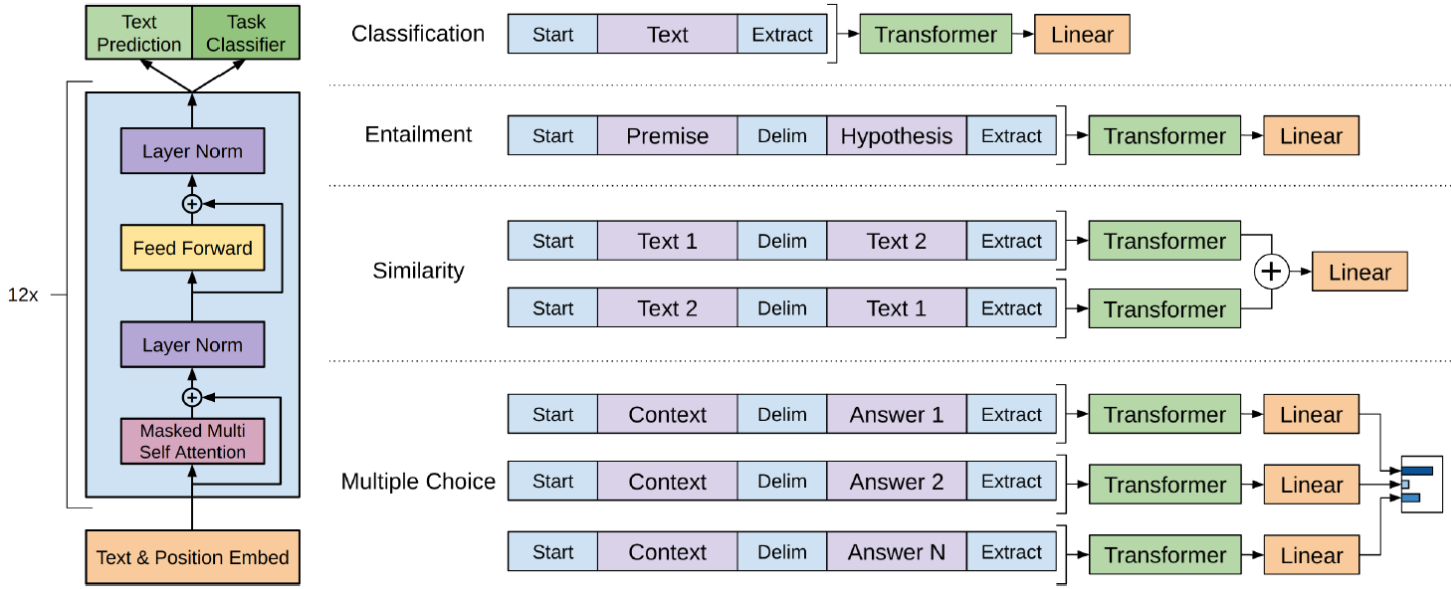
$$h_l = \text{transformer}_{\text{block}(h_{l-1})} \forall i \in [1, n]$$

$$p(u) = \text{softmax}(h_n W_e^{\text{T}})$$

Objective function: maximize $\sum_i \log p(u_i | u_{i-k}, \ldots, t_{i-1}; \Theta)$

OpenAI GPT in NLP

(Radford et al., 2018)



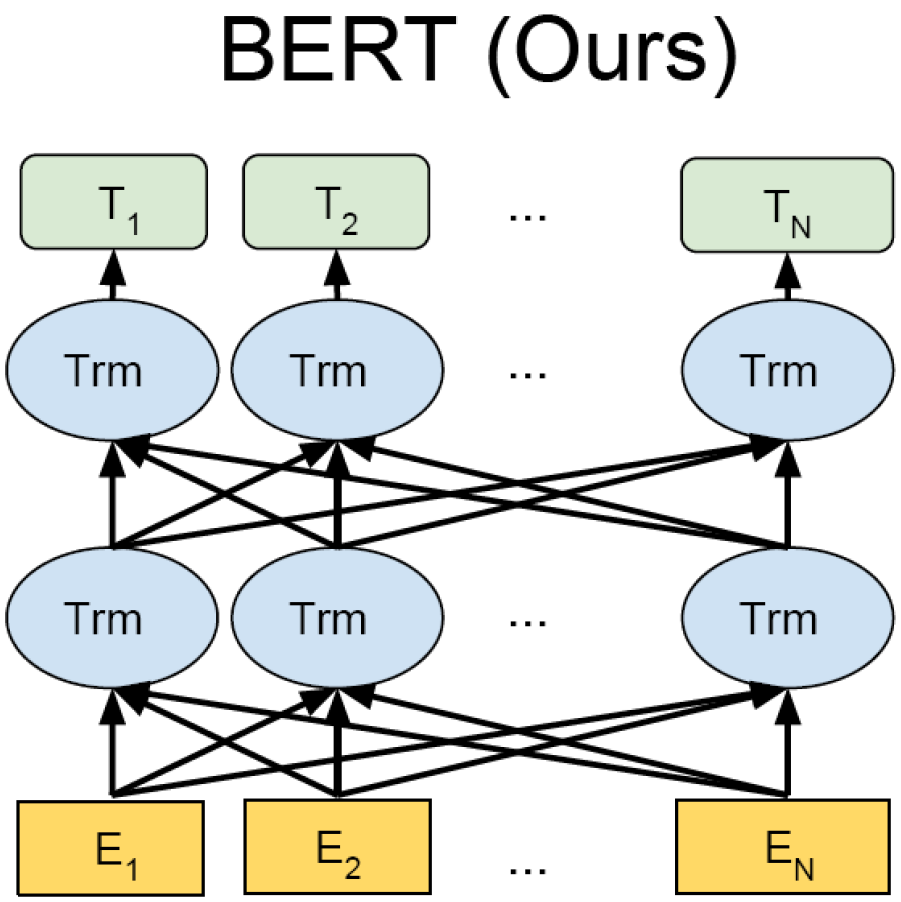
Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

Method	**Classification**		**Semantic Similarity**			**GLUE**
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

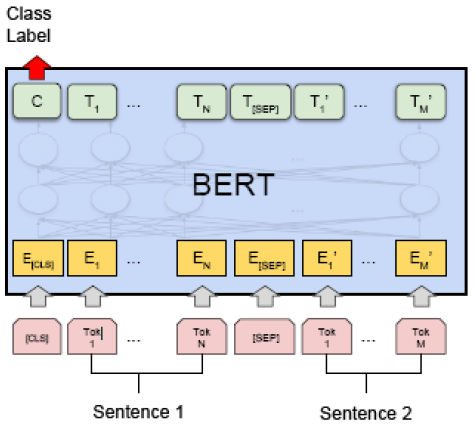
BERT

(Devlin et al., 2018)

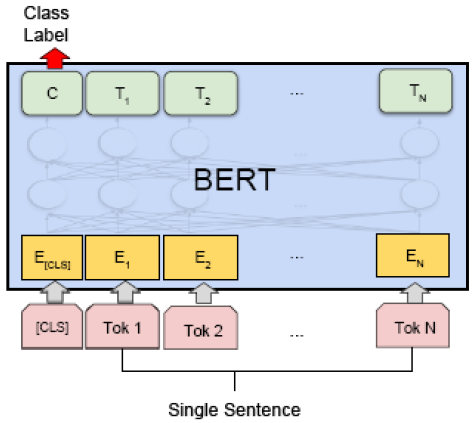


BERT in NLP

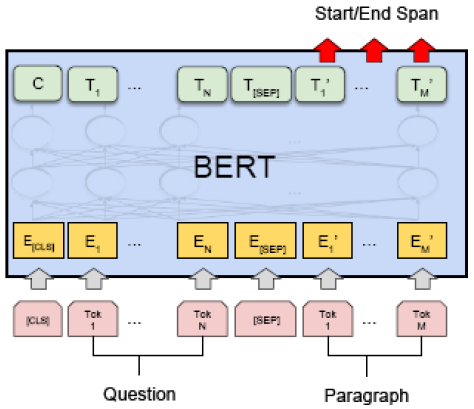
(Devlin et al., 2018)



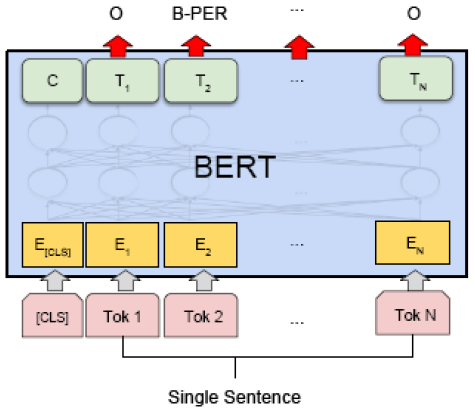
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



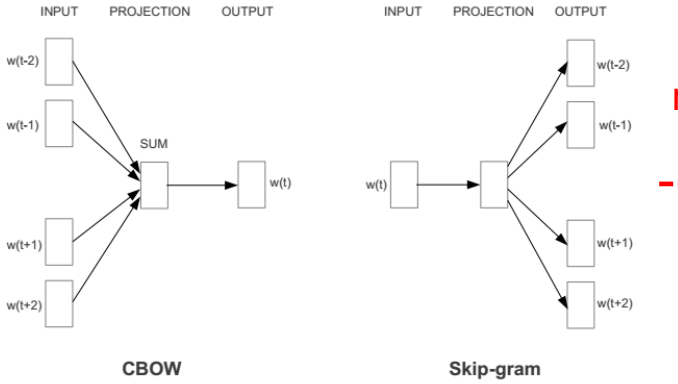
(c) Question Answering Tasks: SQuAD v1.1



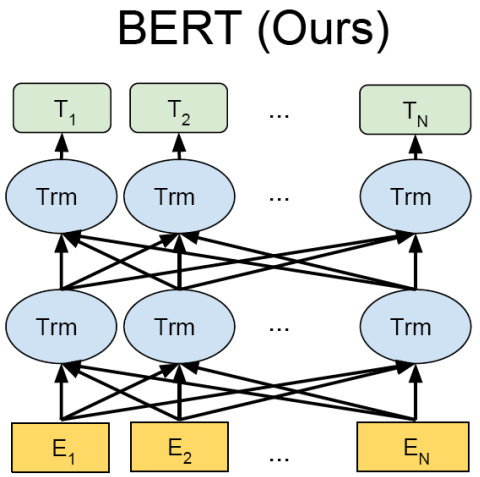
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

System	Dev		Test	
	EM	F1	EM	F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

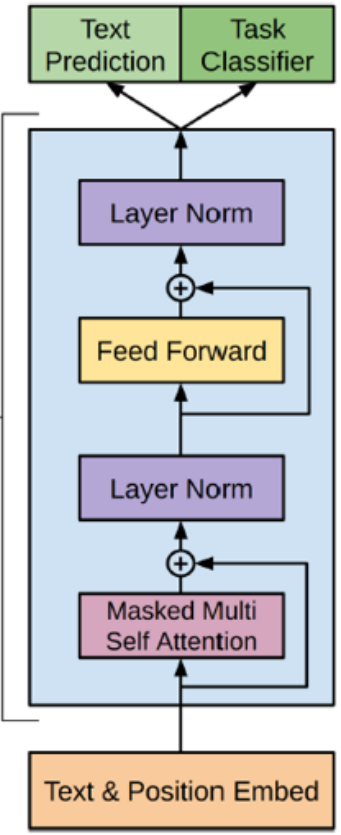
Comparison

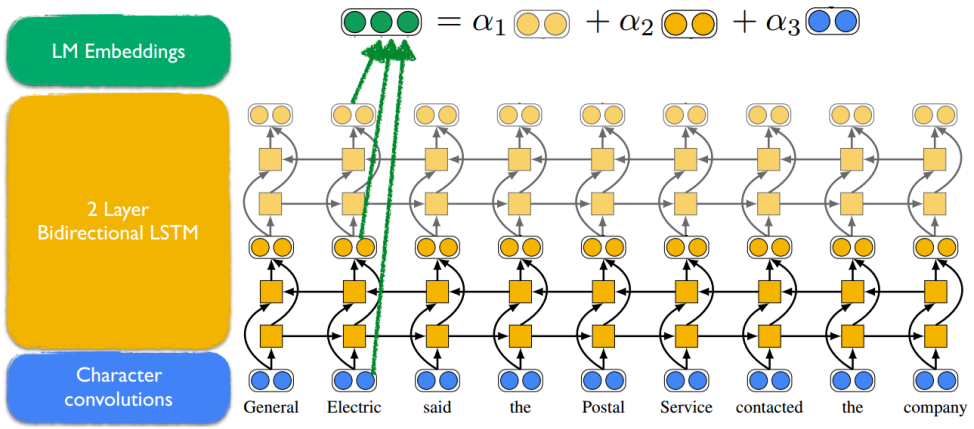


same idea, but more complicated network structure



from single-directional to bi-directional





from LSTM to Transformer

Conclusion (1)

- Summary
  - Transformer is better than RNN
  - Bi-direction is better than single-direction
  - Size of training data, the larger the better

- Trend
  - Task-aware
  - Knowledge-aware

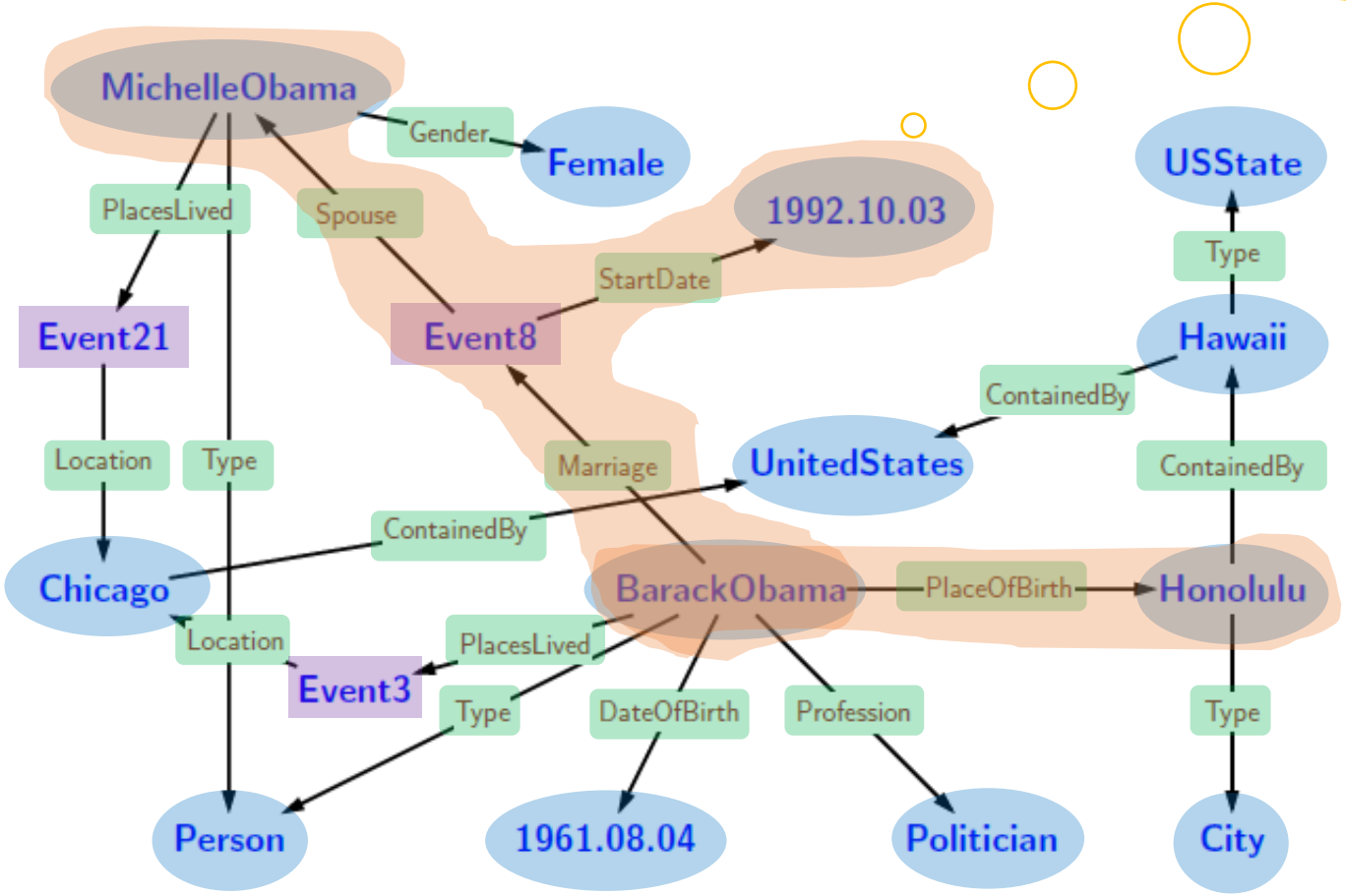
Break ^_^

Today's Agenda

- Deep Learning Basics for NLP
- NLP with Pre-trained Embeddings
- **NLP with Knowledge Bases**
- NLP with Commonsense
- Summary and Trend

Knowledge Graph

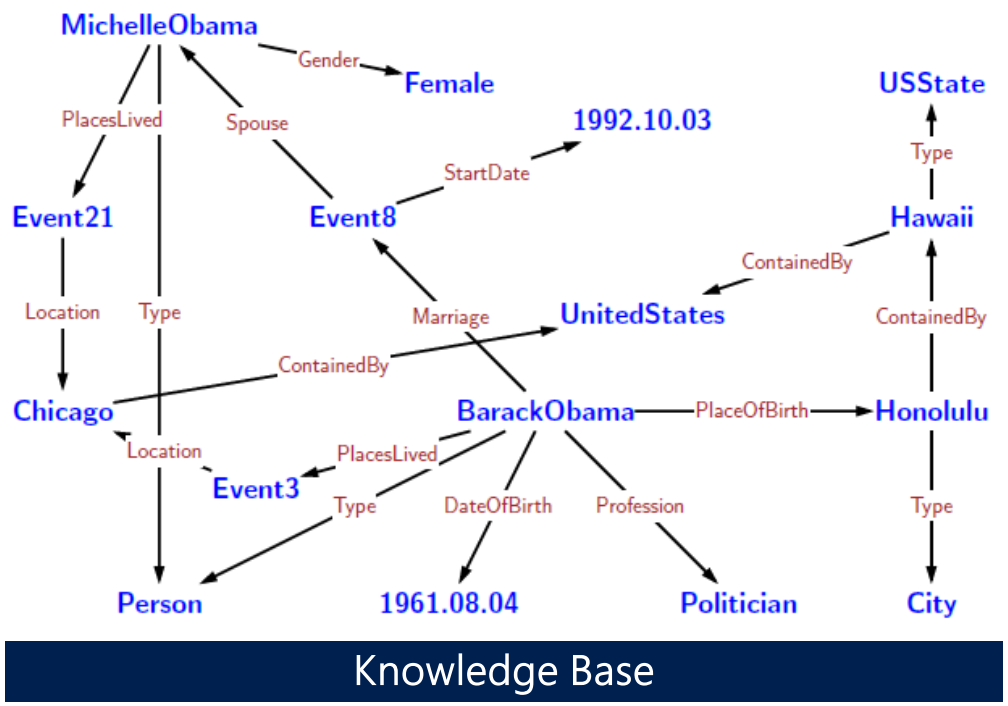
- Curated database with well-defined ontology



- **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

Semantic Parsing with Knowledge Graph

- Convert NL utterances into machine executable LFs based on knowledge base



When was Barack Obama born? → $\lambda x. DateOfBirth(BarackObama, x)$

Who is his wife? → $\lambda x \lambda y. Marriage(BarackObama, y) \wedge Spouse(y, x)$

Single-turn Semantic Parsing

Multi-turn Semantic Parsing

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

⟨expression⟩ ≔ ⟨constant⟩ | ⟨variable⟩ | ⟨function⟩ | ⟨application⟩

⟨function⟩ ≔ λ ⟨variable⟩. ⟨expression⟩

⟨application⟩ ≔ ⟨expression⟩ ⟨expression⟩

Lambda Calculus: Constant

- Represent objects in the world

China, Bill Gates, Mount Everest, 2017, …

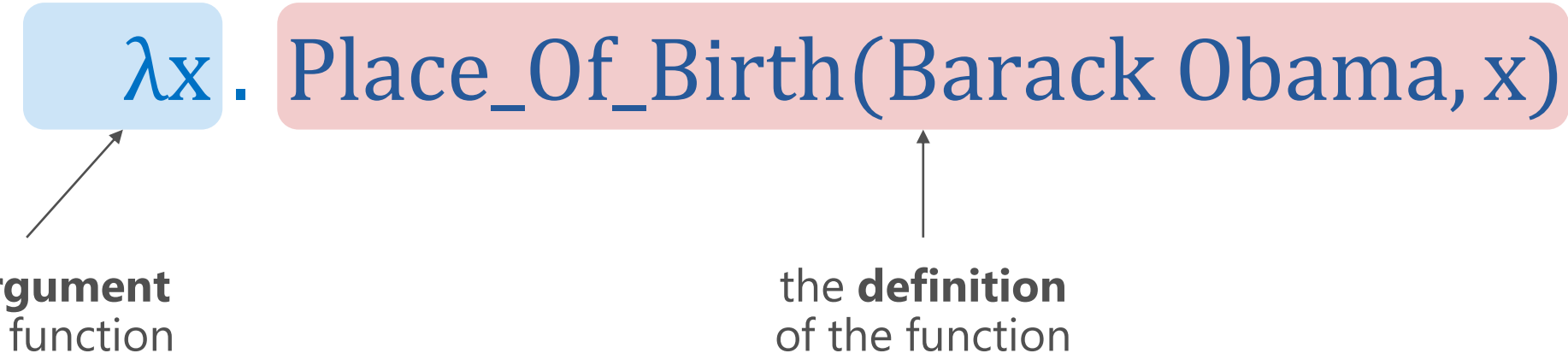
Lambda Calculus: Variable

- Represent object variables

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

Lambda Calculus: Function

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



Lambda Calculus: Application

- Apply the first expression to the second expression

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

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

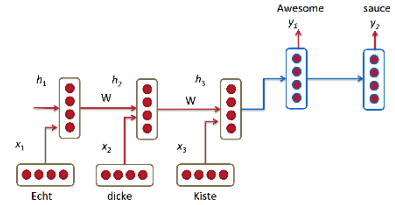
Transforming Natural Language into Logical Form (λ-Calculus)

Natural Language

What city was Obama born ?

Semantic Parsing

Grammar-based Semantic Parsing



Neural Network-based Semantic Parsing

Logical Form

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

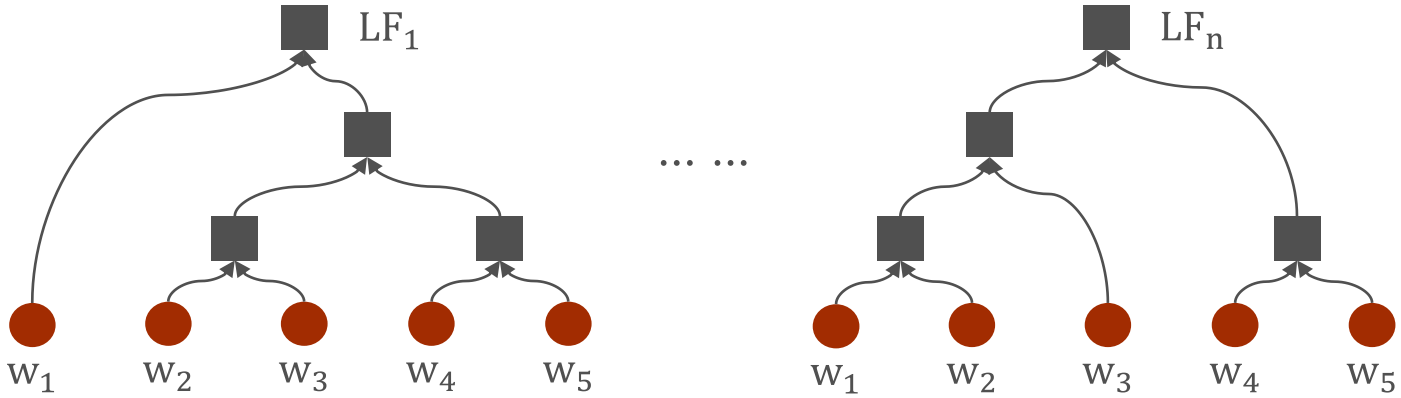
Generic Framework of Grammar-based Semantic Parsing

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

Logical Form Ranking

Rank different logical forms

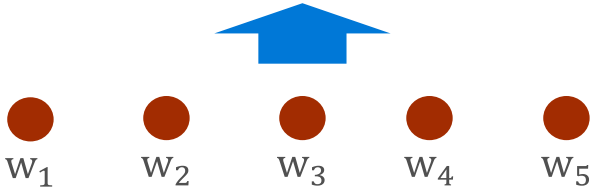
- Q: a question
- LF: a logical form of Q
- D: a semantic parsing derivation
- $\bar{\theta}$: model parameters



Semantic Rule-Driven CYK Parsing

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

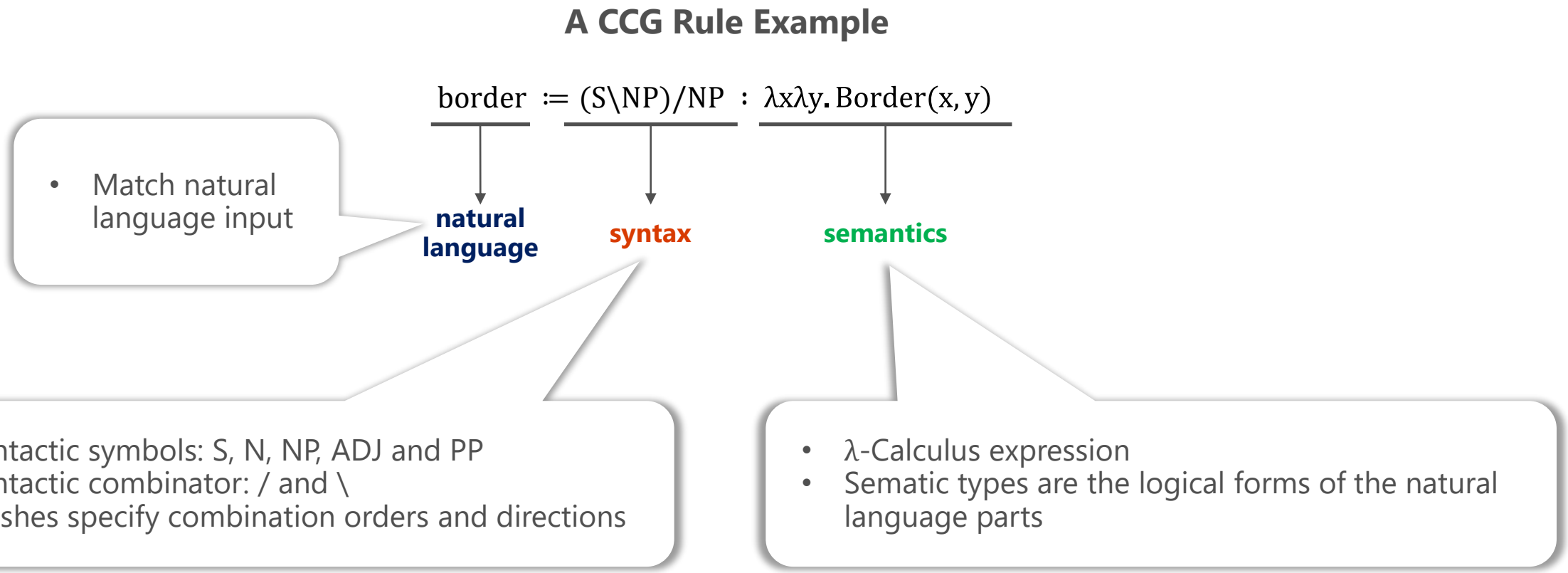
- CCG
- SCFG
- λ-DCS
- …



Natural Language Question

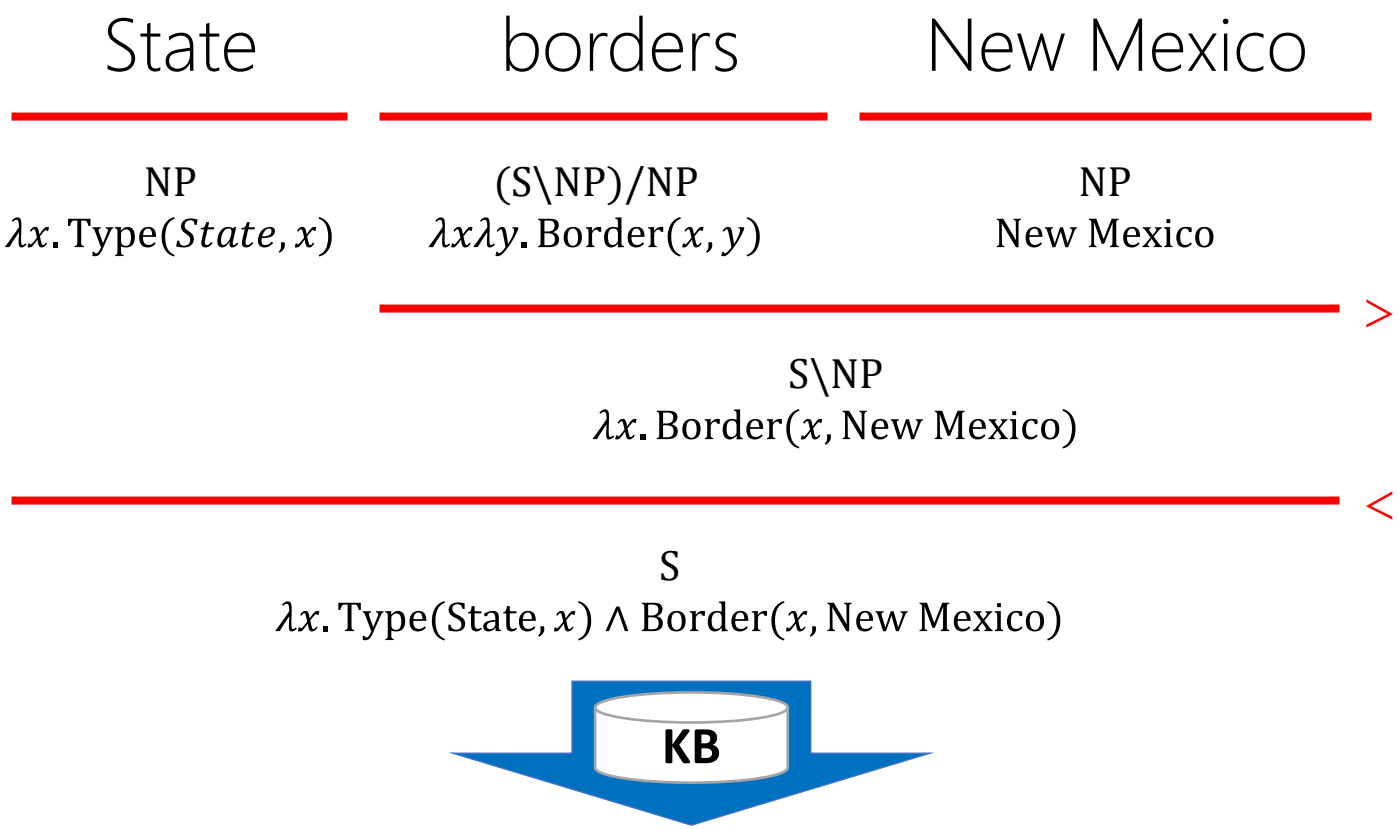
Combinatorial Categorial Grammar (CCG)

- CCG captures syntactic and semantic information jointly



Semantic Parsing with CCG

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



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

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)

- Output (CCG rules)

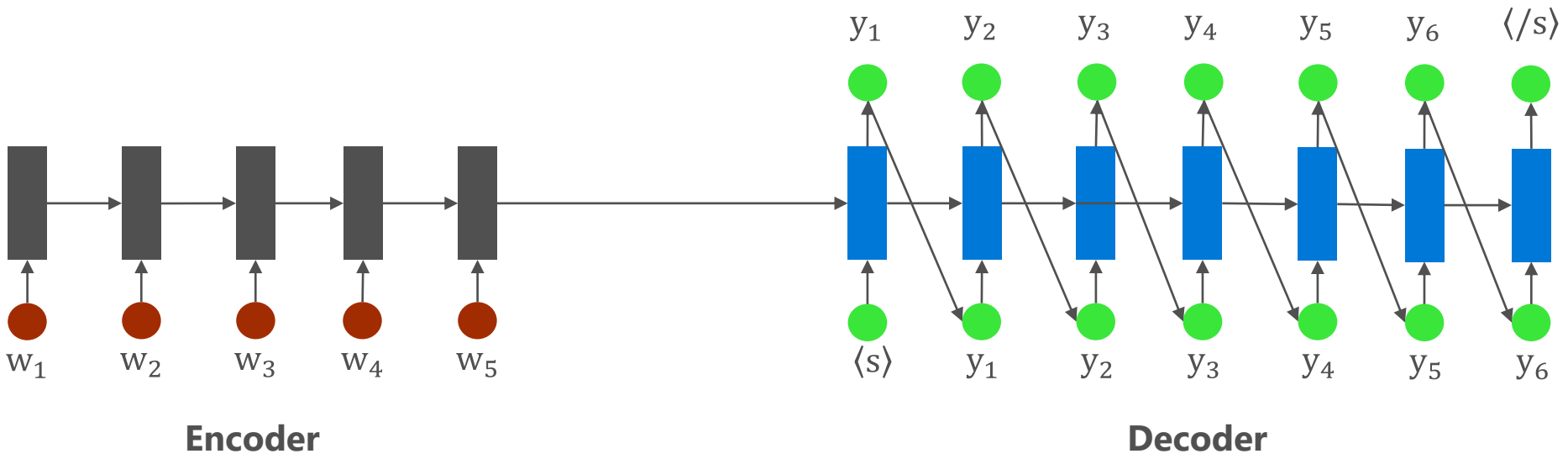
Texas := NP : texas
borders := (S \ NP) / NP : λx.λy.borders(y, x)
New Mexico := NP : new_mexico

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**

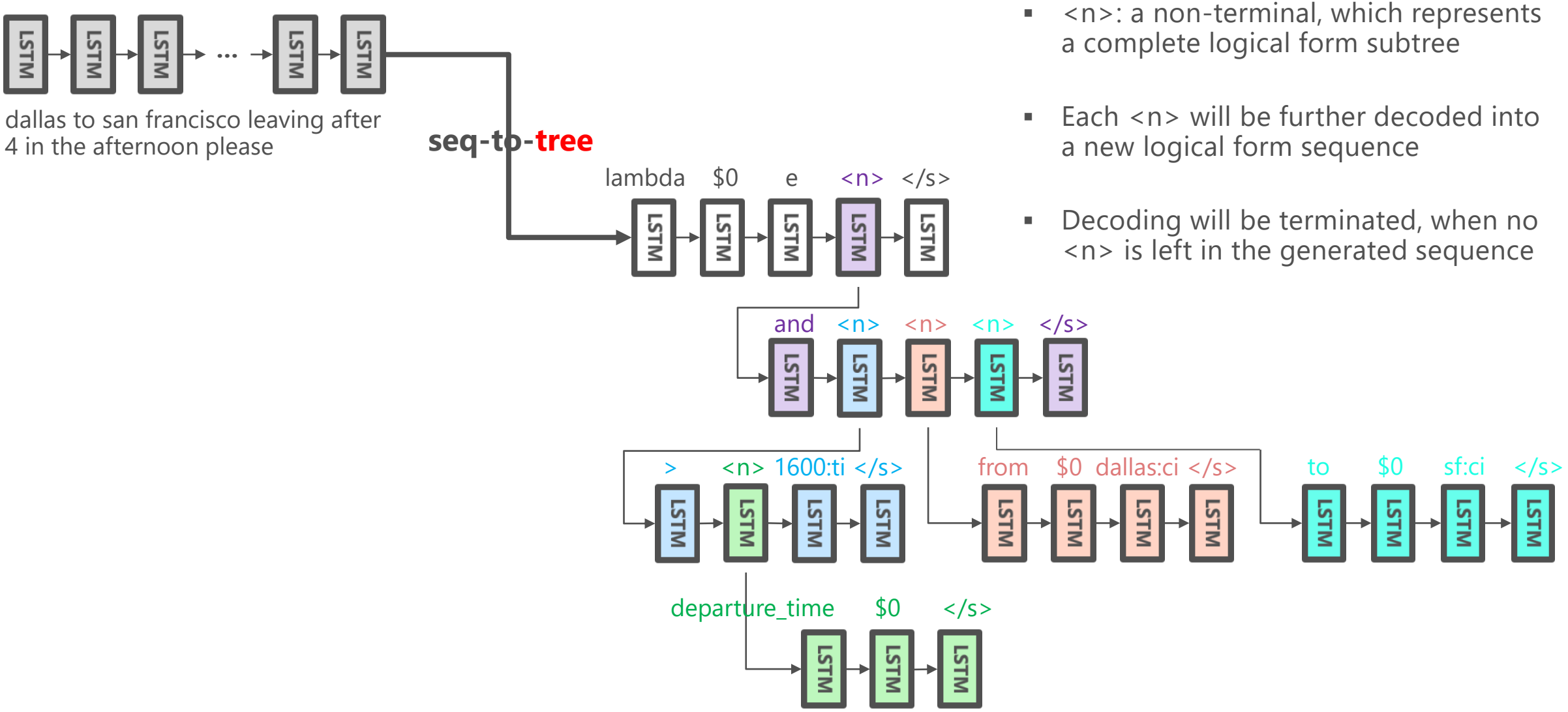
Generic Framework of Neural Network-based Semantic Parsing

- Perform semantic parsing as neural machine translation
- **Encoder** encodes each question into hidden states using RNN
- **Decoder** generates a logical form word-by-word based on question encoding using RNN



Single-turn Semantic Parsing with Sequence-to-Tree

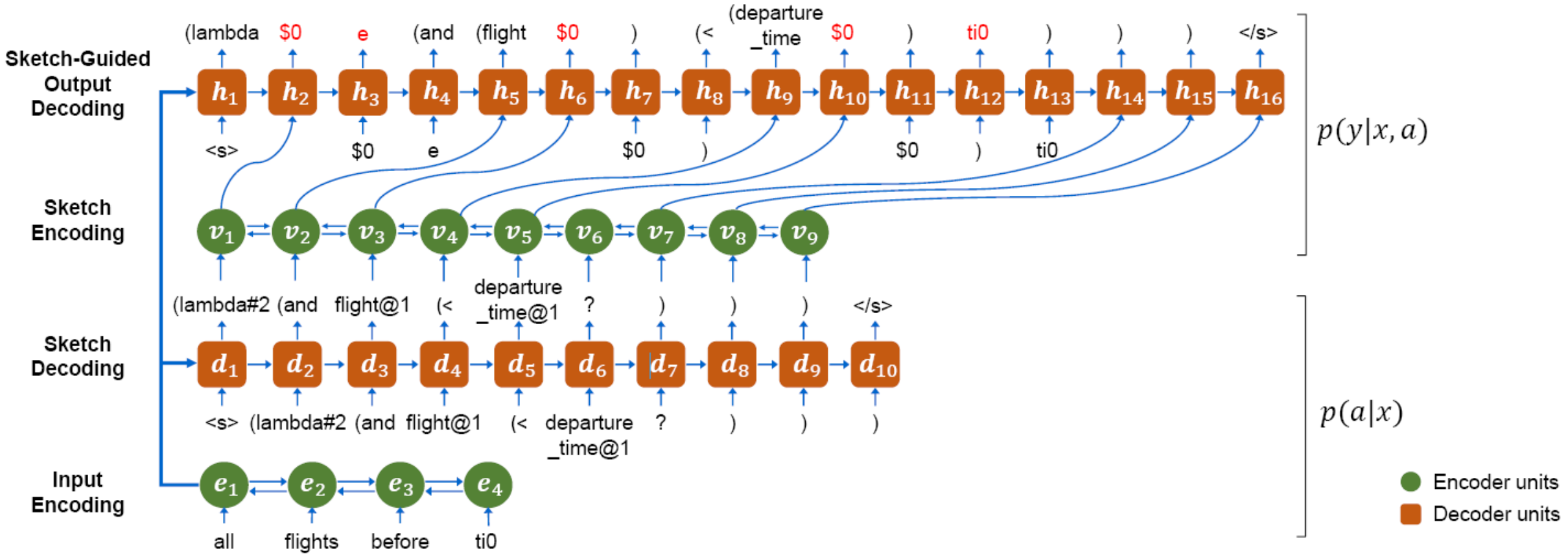
(Dong and Lapata, 2016; Jia and Liang, 2016)



- <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

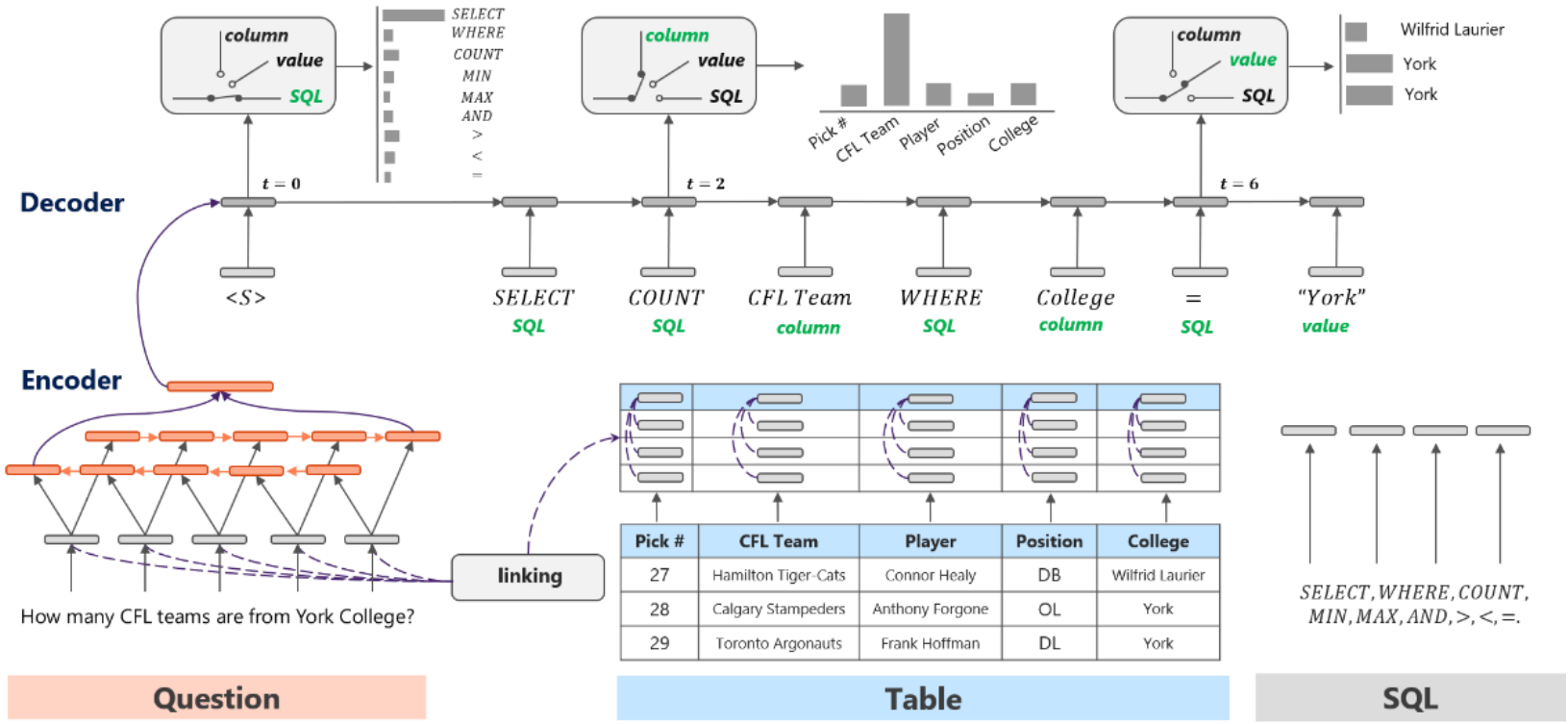
Single-turn Semantic Parsing with Sketch Decoding

(Dong and Lapata, 2018)



Single-turn Semantic Parsing with Multiple Gates

(Sun et al., 2018)



Datasets from Simple Questions to Complex Questions

The SimpleQuestions dataset

This section proposes SimpleQuestions, a dataset collected for research in automatic question answering with human generated questions. Details and baseline results on this dataset can be found in the paper:

Antoine Bordes, Nicolas Usunier, Sumit Chopra and Jason Weston. Large-Scale Simple Question answering with Memory Networks, arXiv:1506.02075.

The SimpleQuestions dataset consists of a total of 108,442 questions written in natural language by human English-speaking annotators each paired with a corresponding fact, formatted as (subject, relationship, object), that provides the answer but also a complete explanation. Facts have been extracted from the Knowledge Base Freebase. We randomly shuffle these questions and use 70% of them (75910) as training set, 10% as validation set (10845), and the remaining 20% as test set.

Here are some examples of questions and facts:

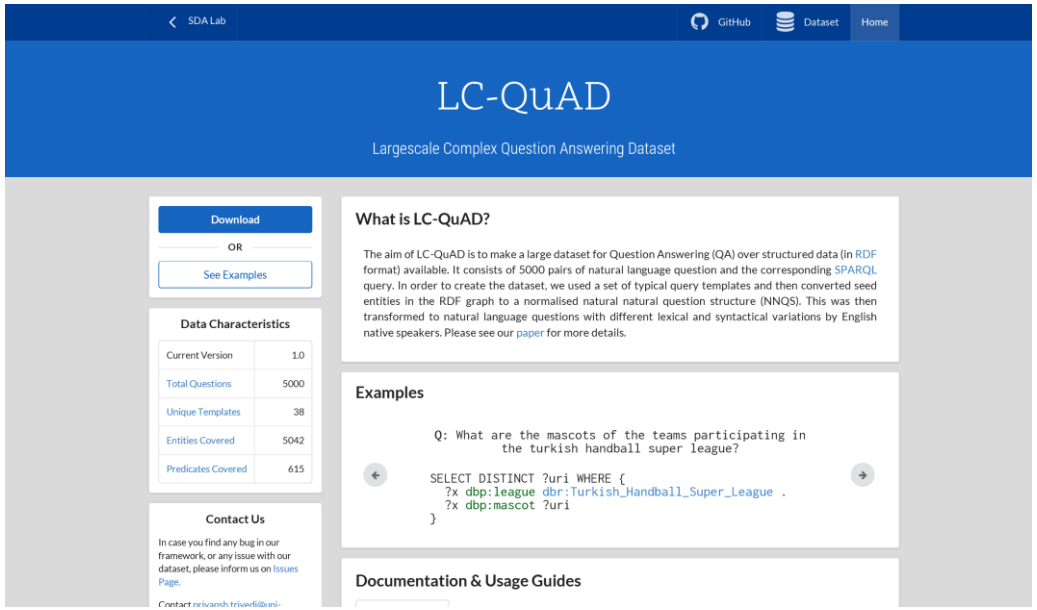
```
* What American cartoonist is the creator of Andy Lippincott?
  Fact: (andy_lip
* Which forest is
  Fact: (fires_cr
* What does Jimmy
  Fact: (jimmy_ne
* What dietary re
  Fact: (kimchi,
```

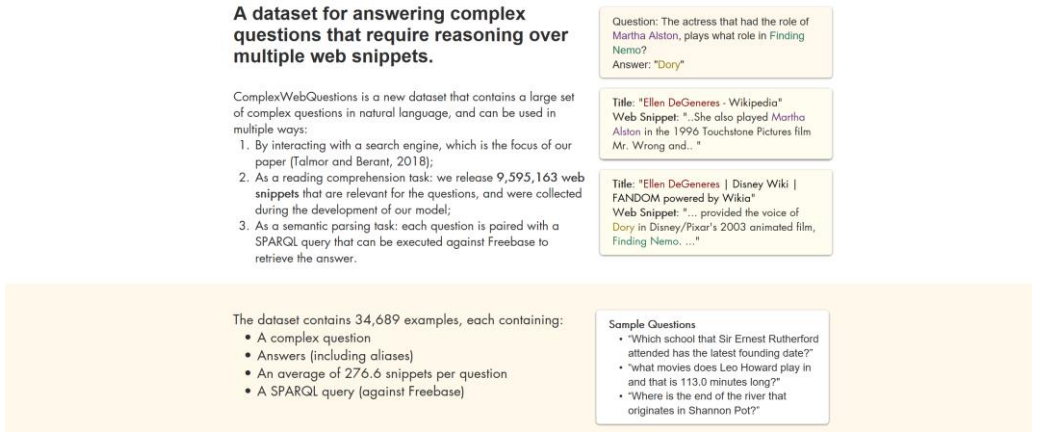
SimpleQuestions Nearly Solved:
A New Upperbound and Baseline Approach

Michael Petrochuk
University of Washington Department of Computer Science & Engineering
mikep5@cs.washington.edu

Luke Zettlemoyer
University of Washington Department of Computer Science & Engineering
lsz@cs.washington.edu

Because of data ambiguity, the upper-bound performance on this benchmark at 83.4% (Petrochuk and Zettlemoyer, 2018).





Coreference and Ellipsis Phenomena in Multi-turn Scenarios

Question Entity Coreference

- **Q1:** Who is the president of the United States?
- **A1:** Donald Trump
- **Q2:** what is its population?

Answer Entity Coreference

- **Q1:** Who is the president of the United States?
- **A1:** Donald Trump
- **Q2:** How many children does he have?

Question Subsequent Coreference

- **Q1:** Where did the president of the United States born?
- **A1:** New York City
- **Q2:** Where did he graduate from?

Answer Entity Ellipsis

- **Q1:** What movie did Leonardo DiCaprio won an Oscar for?
- **A1:** The Revenant
- **Q2:** who is the director?

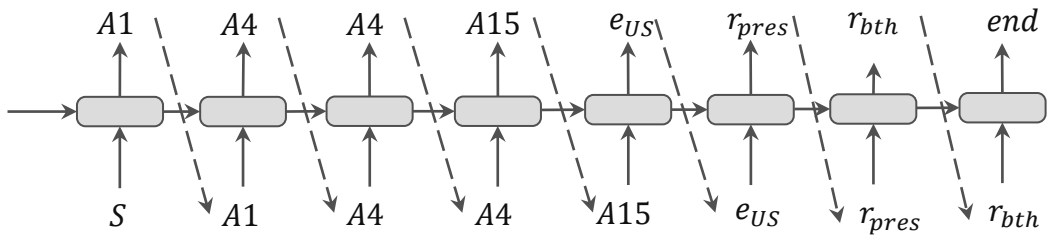
Question Predicate Ellipsis

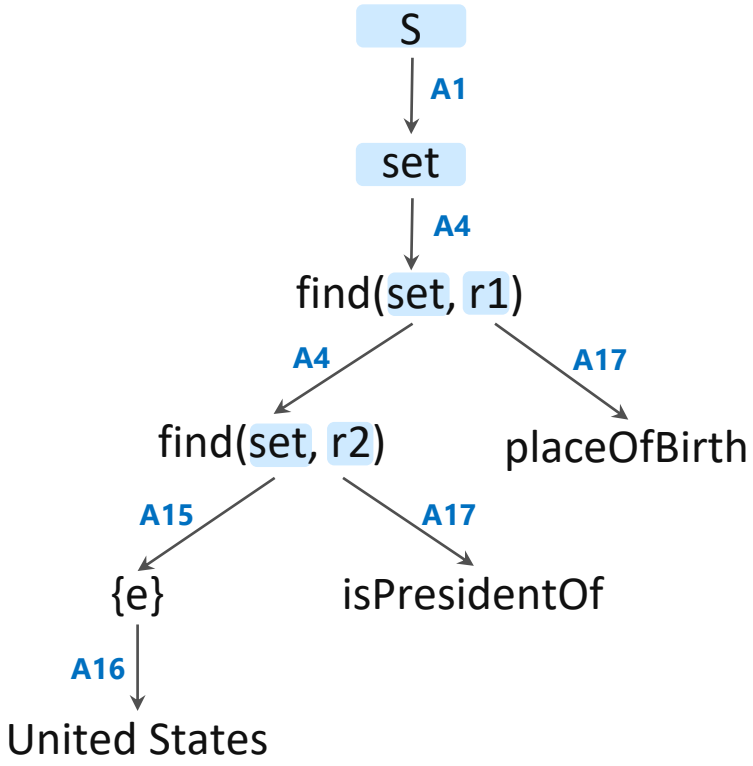
- **Q1:** Who is the president of the United States?
- **A1:** Donald Trump
- **Q2:** how about China?

Multi-turn Semantic Parsing with Dialogue Actions & Memory

(Guo et al., 2018)

Where did the president of the United States born?





A1: S → Set

A4: Set → find (set, r1)

A4: Set → find (set, r2)

A15: Set → {e}

A16: e → United States

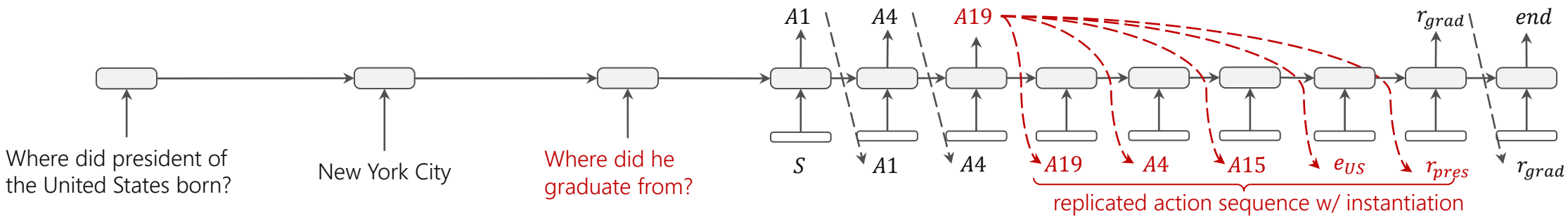
A17: r2 → isPresidentOf

A17: r1 → placeOfBirth

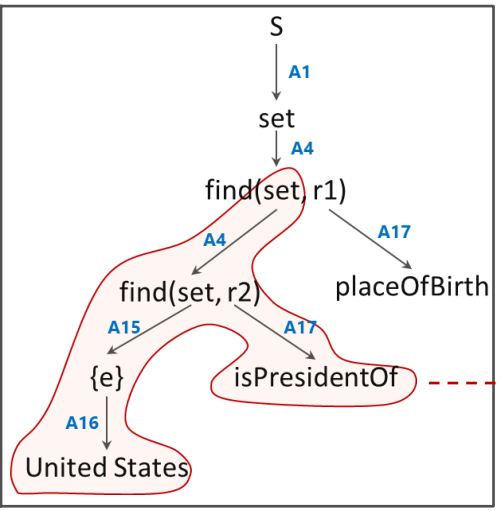
Action	Operation	Description
A1-A3	S—>Set | Num | Bool	S is start symbol
A4	Set—>Find(R, E)	Set of entities with a r edge to e
A5	Num—>Count(Set)	Total number of set
A6	Bool—>(∈, E, Set)	Whether E∈Set
A7	Set—>Set ∪ Set	Union of Sets
A8	Set—>Set ∩ Set	Intersection of Sets
A9	Set—>Set —Set	Difference of Sets
A10	Set->larger(set, r, num)	Entity from set linking to more than num entities with relation r
A11	Set->less(set, r, num)	Entity from set linking to less than num entities with relation r
A12	Set->equal(set, r, num)	Entity from set linking to num entities with relation r
A13	Set->arxmax(set, r, num)	Entity from set linking to most entities with relation r
A14	Set->argmin(set, r, num)	Entity from set linking to least entities with relation r
A15	Set->{e}	
A16-A18	e | r | num -> constant	instantiation for entity e, predicate r or number num
A19-A21	Set | Num | Bool ->action(i-1)	Replicate previous operation sequence

Multi-turn Semantic Parsing with Dialogue Actions & Memory

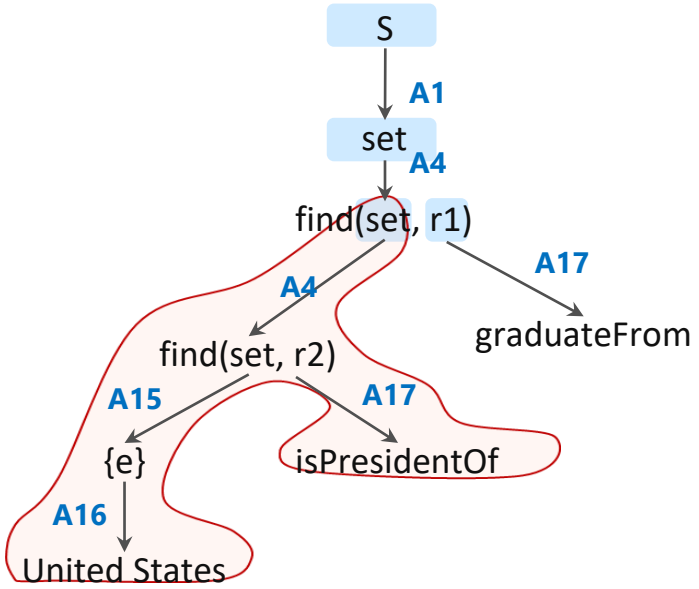
(Guo et al., 2018)



Dialog Memory	
Entity	{United States, tag=utterance} {New York City, tag=answer}
Predicate	{isPresidentOf} {placeOfBirth}
Action Subsequence	$set \rightarrow A4\ A15\ e_{US}\ r_{pres}$ $set \rightarrow A4\ A15$ $set \rightarrow A4\ A4\ A15\ e_{US}\ r_{pres}\ r_{bth}$ $set \rightarrow A4\ A4\ A15$



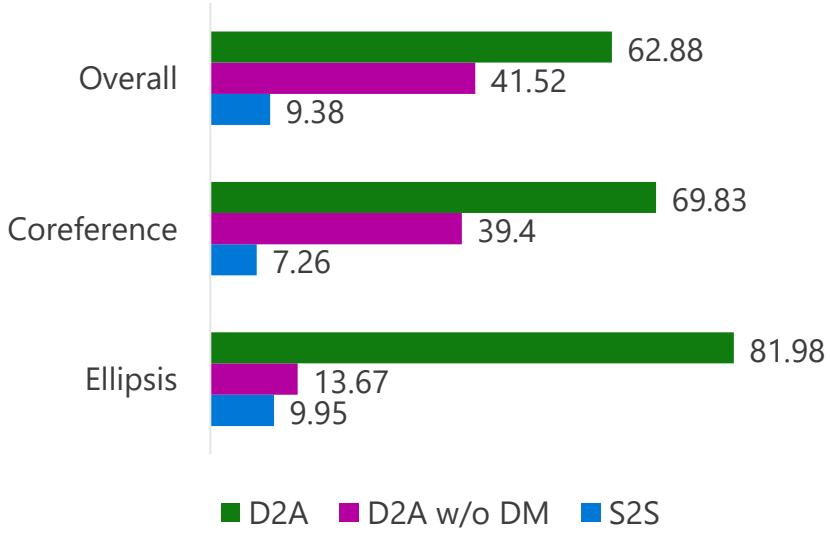
copy



Evaluation on CSQA Dataset

CSQA Dataset (IBM, 2018)

Dialogs	200,000
Turns	1.6M
Entities in KB	12.8M
Unique relations	330
KB Tuples	21.2M
Entity Types	642



id	question type	current question + previous turn	predicted logical form
1	Simple Question (Direct)	**Q1**: N/A **R1**: N/A **Q2**: Who was the dad of Jorgen Ottesen Brahe?	$find(\{Jorgen\ Ottesen\ Brahe\}, father)$
2	Simple Question (Coreferenced)	**Q1**: Who was the dad of Jorgen Ottesen Brahe? **R1**: Otte Brahe **Q2**: Who is the spouse of that one?	$find(\{Otte\ Brehe\}, spouse)$
3	Simple Question (Ellipsis)	**Q1**: What is the profession of Mkihail Beliaiev? **R1**: Military personnel **Q2**: And also tell me about Brett MacLean	$find(\{Brett\ MacLean\}, occupation)$
4	Logical Reasoning (All)	**Q1**: N/A **R1**: N/A **Q2**: Which administrative territories have diplomatic relations with Italy and are not Derikha present in?	$and(\ diff(\ find(\{Italy\}, reverse(diplomatic\ relation)),$ $find(\{Derikha\}, country),$ $find(\{administrative\ territories\}, isA)\ ))$
5	Quantitative Reasoning	**Q1**: N/A **R1**: N/A **Q2**: Which works did min number of people do the dubbing for?	$argmin(find(\{voice\ actor\}, isa), reverse(work)\ )$
6	Comparative Reasoning	**Q1**: N/A **R1**: N/A **Q2**: Which musical instruments are played by more number of people than electronic keyboard?	$larger(find(\{musical\ instruments\}, isA), reverse(instrument),$ $count(and(find(\{electronic\ keyboard\}, reverse(instrument)),$ $find(\{people\}, isA))\ ))$
7	Verification (Boolean)	**Q1**: N/A **R1**: N/A **Q2**: Is Arizona Coyotes present in United States of America?	$in(Arizona\ Coyotes\ ,$ $find(\{United\ States\ of\ America\}, reverse(country)\ ))$
8	Quantitative Reasoning (Count)	**Q1**: How many people have birthplace at Provence? **R1**: 15 **Q2**: And how about Peterborough?	$copy(\ count(\ find(\{Peterborough\}, reverse(place\ of\ birth)\ )))$
9	Comparative Reasoning (Count)	**Q1**: How many musical instruments are played by greater number of people than Body percussion ? **R1**: 30 **Q2**: And also tell me about timpani?	$copy(count(\ larger(\ find(\{musical\ instrument\}, isA),$ $reverse(instrument)\ \ ,$ $count(\ find(\ \{timpani\}\ , reverse(instrument))\ )\ )\ ))$

Conclusion (2)

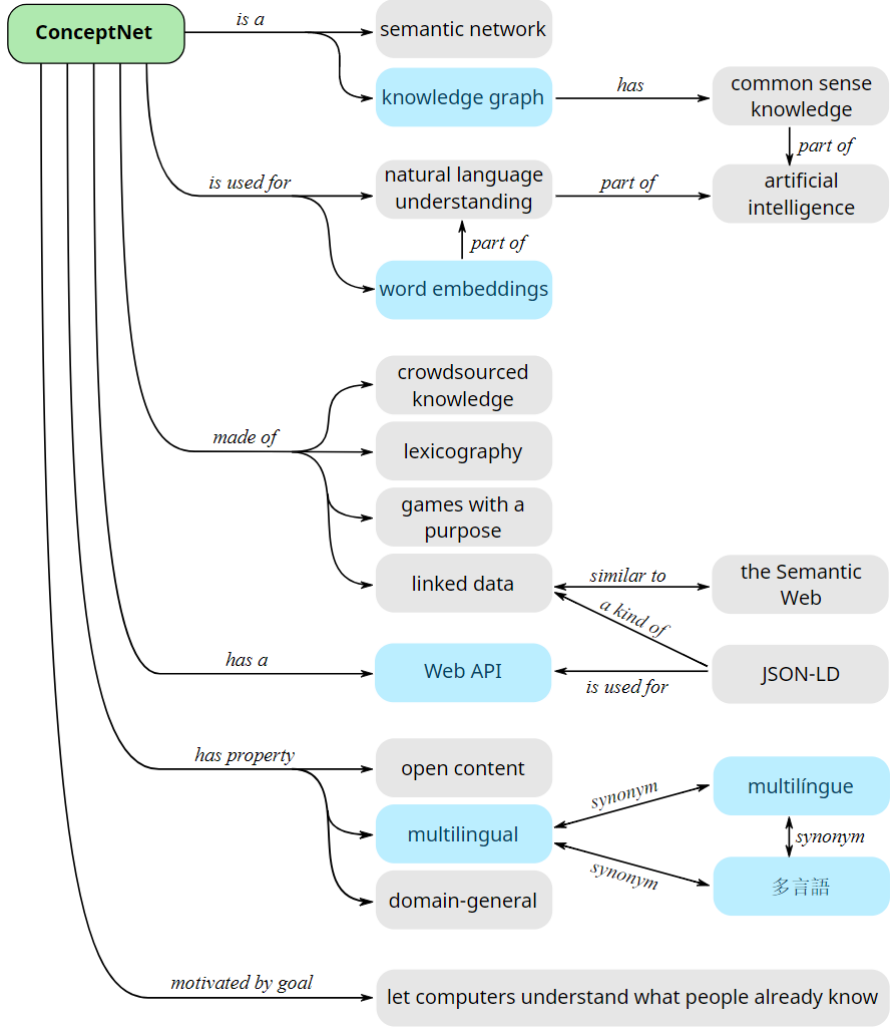
- Summary
  - Semantic parsing is the core task of NLP
  - Dataset set is a big issue for model training

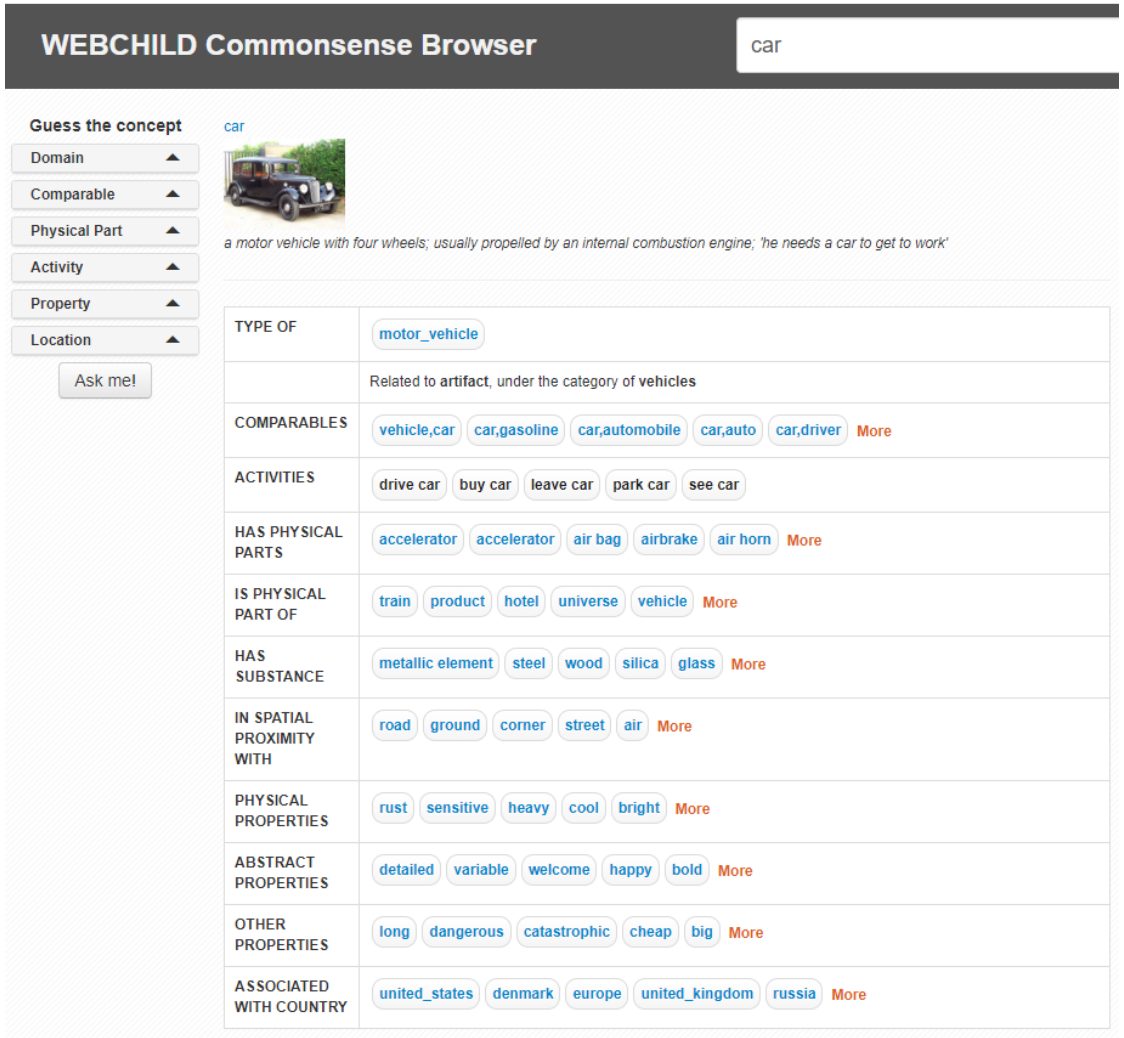
- Trend
  - From grammar to neural work
  - From single-relation to complex-relation
  - From single-turn to multi-tune
  - From single-pass to multi-pass
  - From knowledge graphs to web tables

Today's Agenda

- Deep Learning Basics for NLP
- NLP with Pre-trained Embeddings
- NLP with Knowledge Bases
- **NLP with Commonsense**
- Summary and Trend

Commonsense Knowledge





...

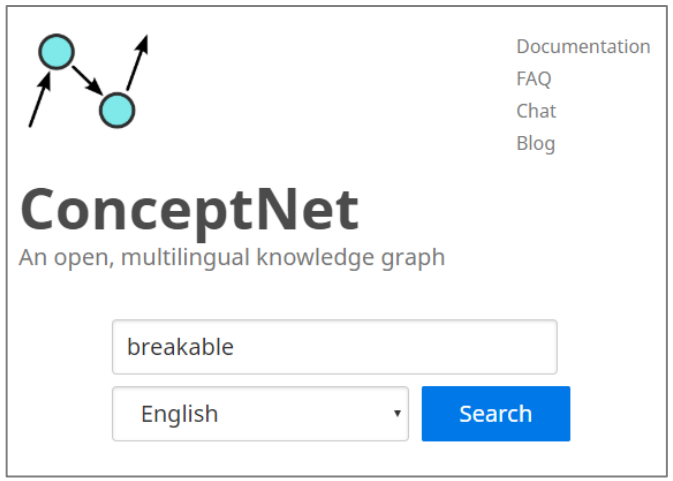
ConceptNet

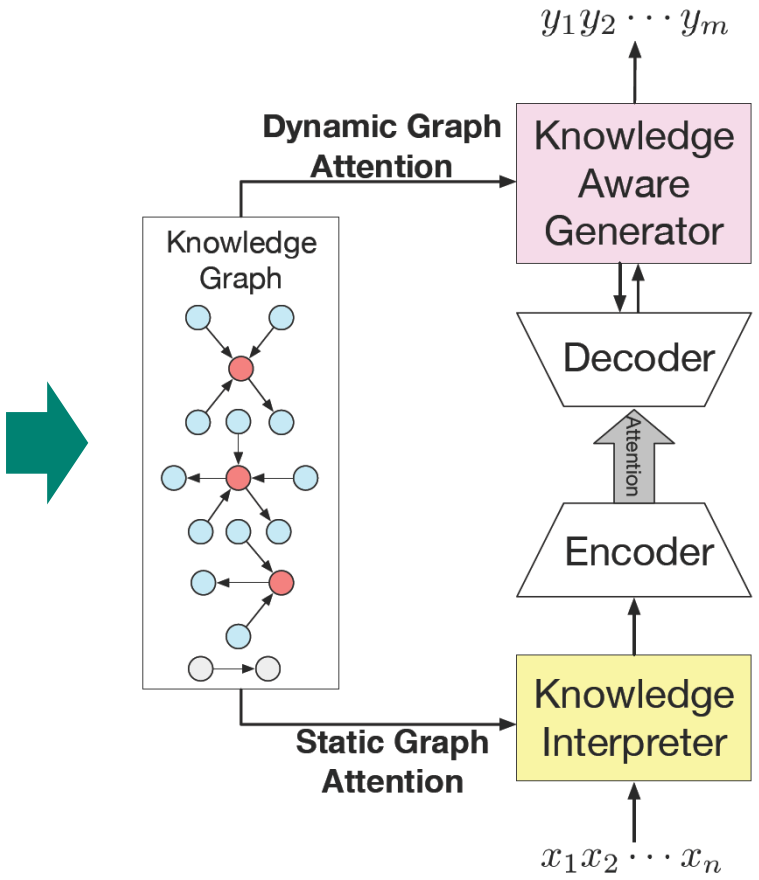
WebChild

...

Dialogue System with Commonsense Knowledge

- Use commonsense knowledge from **knowledge base**





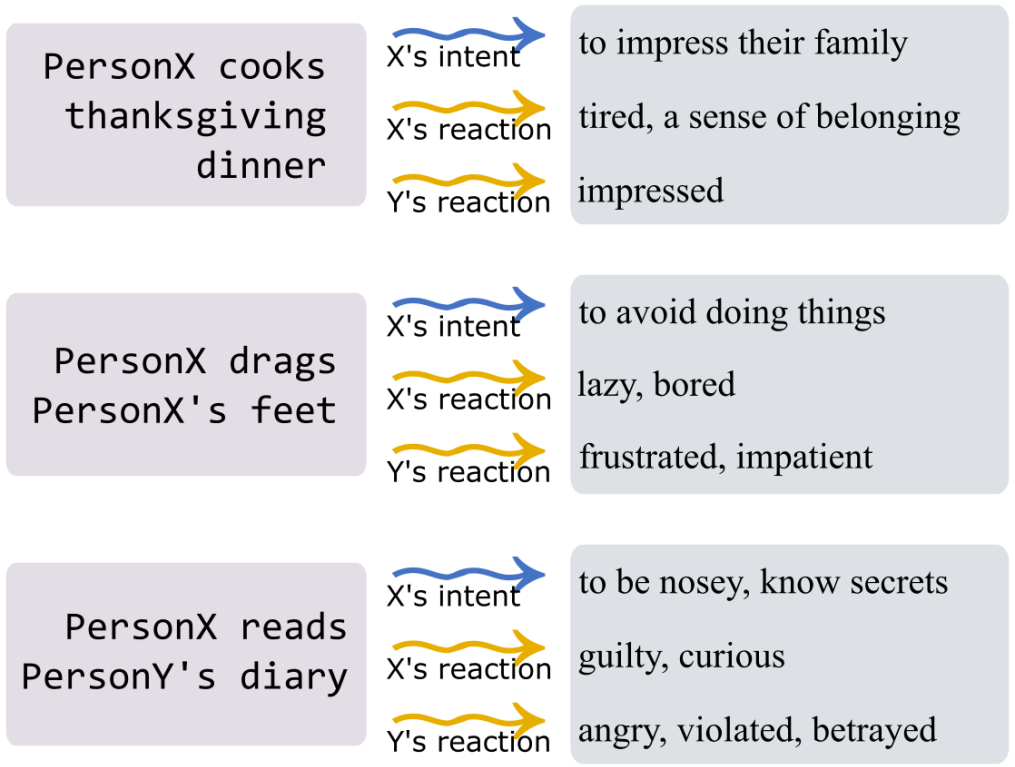
Output Example

Post	Why are you so **breakable**?
Knowledge	(**glass**, RelatedTo, **breakable**), (**brittle**, RelatedTo, **breakable**), (**rule**, RelatedTo, **breakable**)
Seq2Seq	I'm not a **OOV** , I'm just a **OOV**.
MemNet	I'm not **OOV**. I'm just a really nice person.
CopyNet	I'm not. I'm just a lurker.
CCM	Because I'm a **brittle** man .

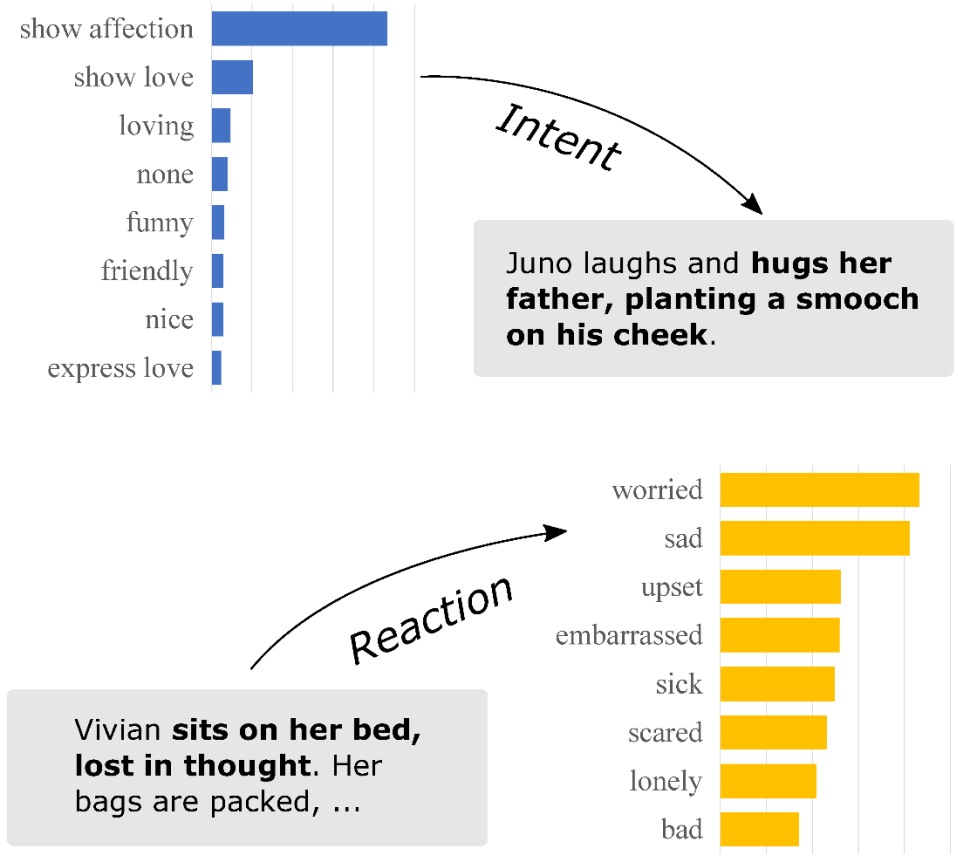
Commonsense aware Conversation Generation
(Zhou et al., 2018)

Intent and Reaction Prediction with Commonsense Knowledge

- Use commonsense knowledge learnt from **supervised data**



Event2Mind
(Rashkin et al., 2018)



QA and Reasoning with Commonsense Knowledge

*The service was poor, **but** the food was ___*

- Use commonsense knowledge learnt from **unsupervised data**

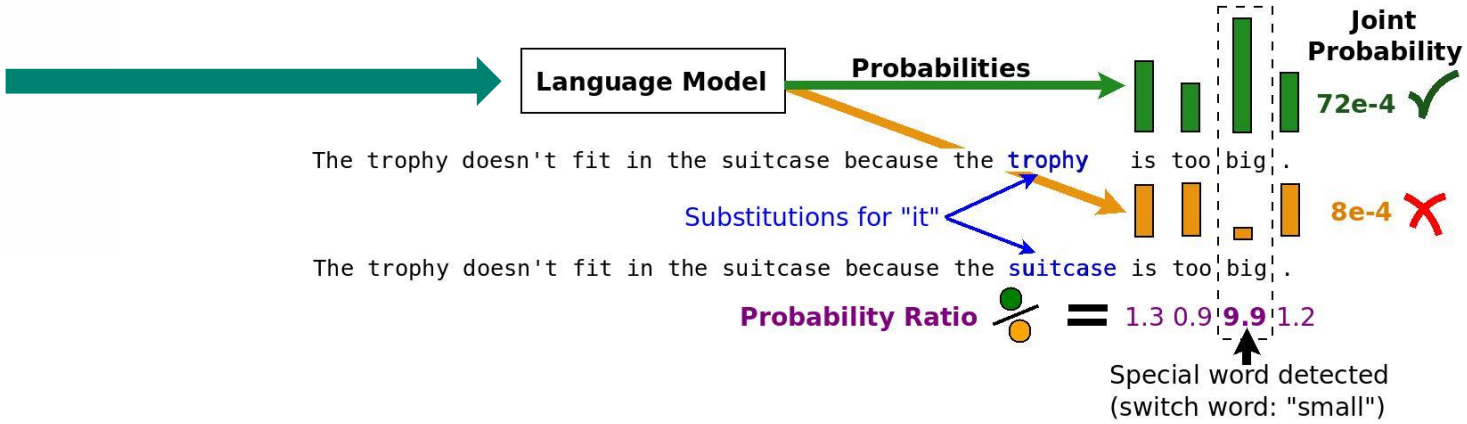
Monolingual Corpus

Winograd Schema Challenge

The trophy doesn't fit in the suitcase because **it** is too big.

Question: What is too big?

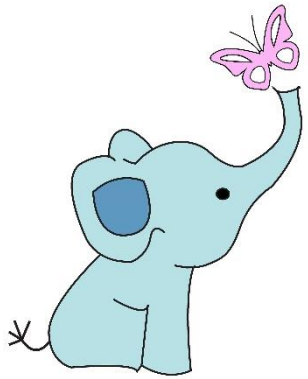
Answer: (a) **the trophy** (b) **the suitcase**



Language Model for Commonsense Reasoning
(Trinh and Le, 2018)

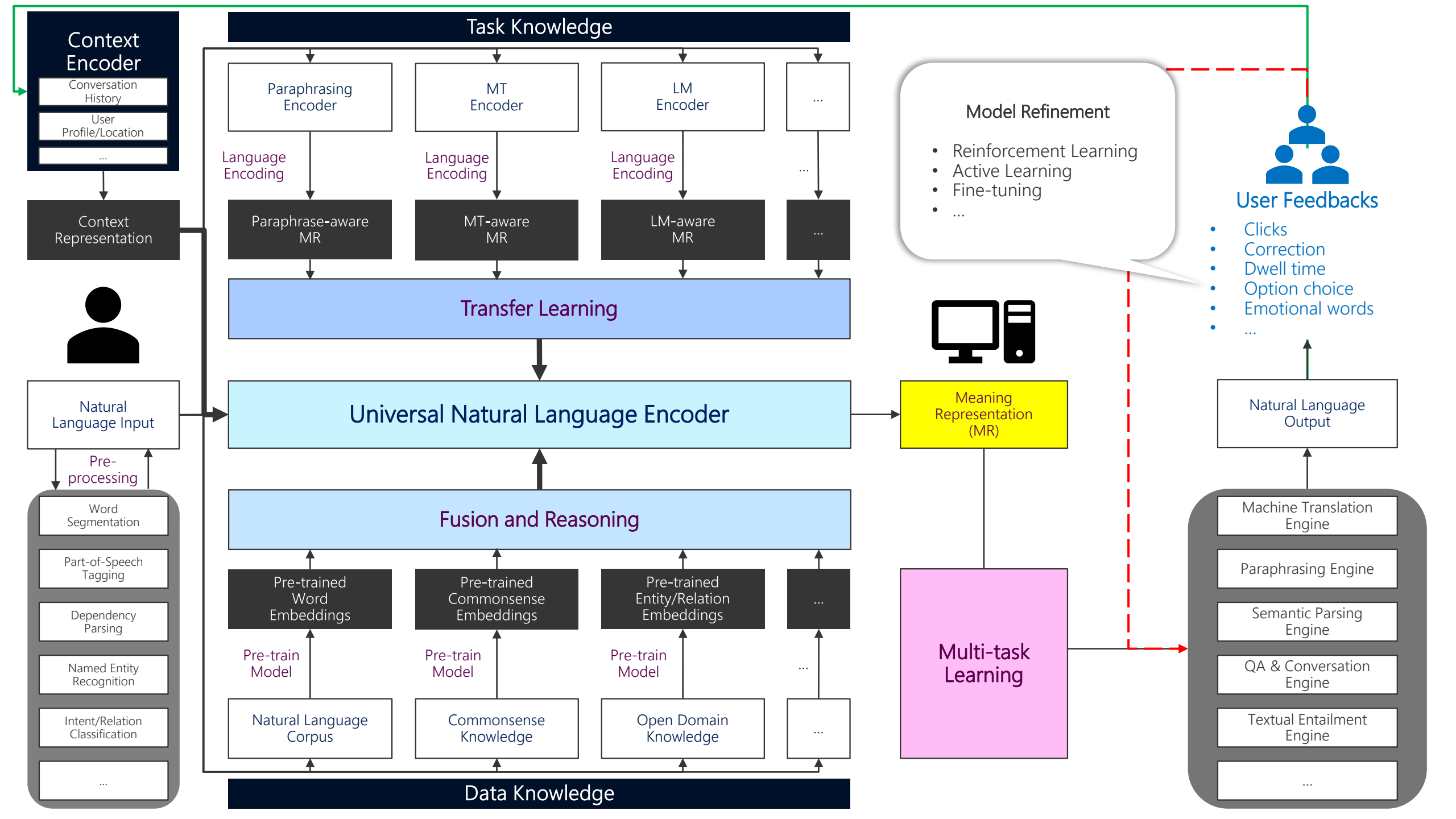
Conclusion (3)

- Commonsense is important to AI models, esp. chatbot and dialogue system
- Few datasets for model developments
- Take-ways
  - Commonsense is invisible most of time
  - But it is important to conversational AIs, such as chatbot and dialogue system

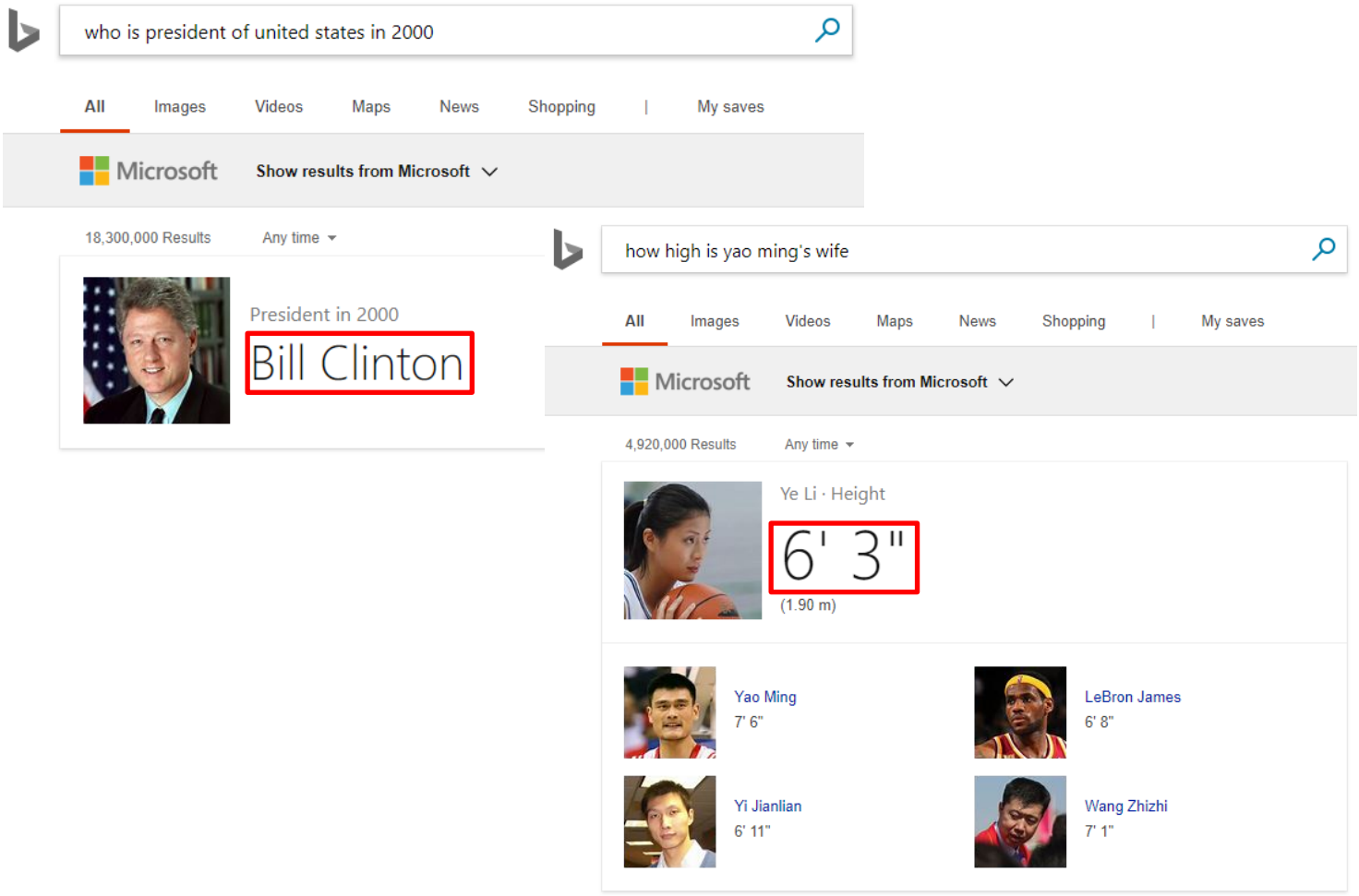


Today's Agenda

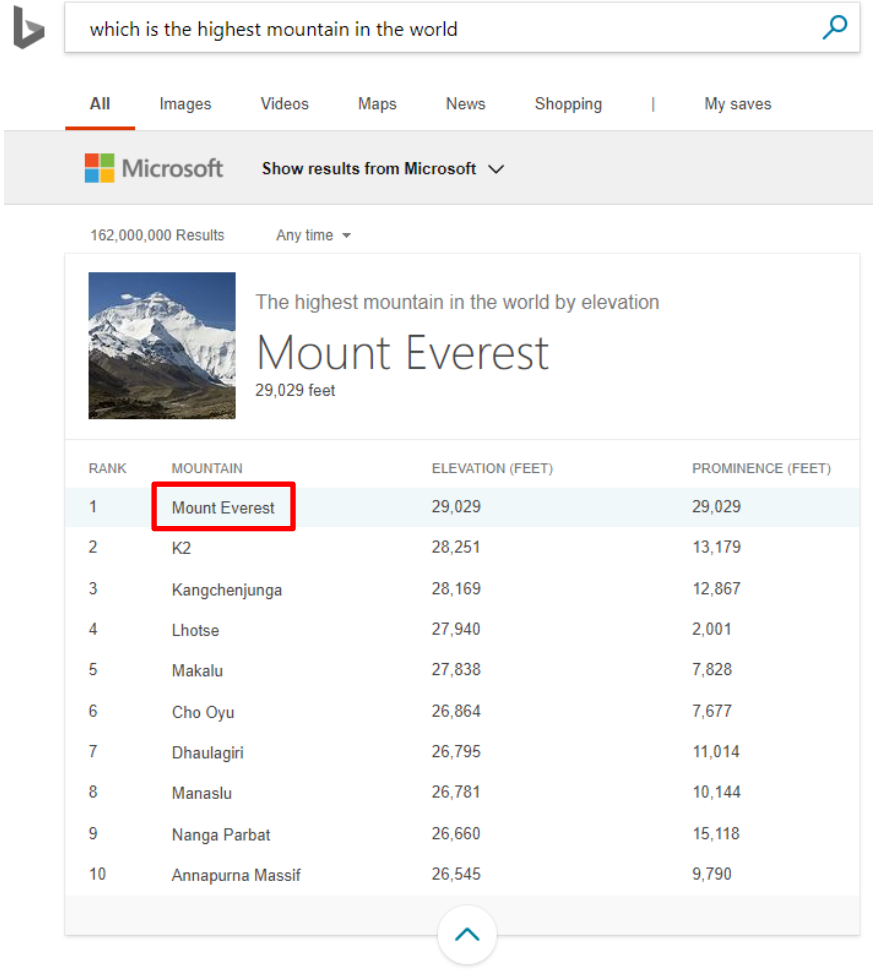
- Deep Learning Basics for NLP
- NLP with Pre-trained Embeddings
- NLP with Knowledge Bases
- NLP with Commonsense
- **Summary and Trend**


Context Encoder
Conversation History
User Profile/Location
...
Context Representation
Task Knowledge
Paraphrasing Encoder
MT Encoder
LM Encoder
...
Language Encoding
Language Encoding
Language Encoding
...
Paraphrase-aware MR
MT-aware MR
LM-aware MR
...
Transfer Learning
Natural Language Input
Pre-processing
Word Segmentation
Part-of-Speech Tagging
Dependency Parsing
Named Entity Recognition
Intent/Relation Classification
...
Universal Natural Language Encoder
Fusion and Reasoning
Pre-trained Word Embeddings
Pre-trained Commonsense Embeddings
Pre-trained Entity/Relation Embeddings
...
Pre-train Model
Pre-train Model
Pre-train Model
...
Natural Language Corpus
Commonsense Knowledge
Open Domain Knowledge
...
Data Knowledge
Model Refinement
Reinforcement Learning
Active Learning
Fine-tuning
...
User Feedbacks
Clicks
Correction
Dwell time
Option choice
Emotional words
...
Meaning Representation (MR)
Multi-task Learning
Natural Language Output
Machine Translation Engine
Paraphrasing Engine
Semantic Parsing Engine
QA & Conversation Engine
Textual Entailment Engine
...

Application in Bing



Bing Knowledge QA



RANK	MOUNTAIN	ELEVATION (FEET)	PROMINENCE (FEET)
1	Mount Everest	29,029	29,029
2	K2	28,251	13,179
3	Kangchenjunga	28,169	12,867
4	Lhotse	27,940	2,001
5	Makalu	27,838	7,828
6	Cho Oyu	26,864	7,677
7	Dhaulagiri	26,795	11,014
8	Manaslu	26,781	10,144
9	Nanga Parbat	26,660	15,118
10	Annapurna Massif	26,545	9,790

Bing Table QA

Application in Cortana

USER who is the guy directed movie inception SEND

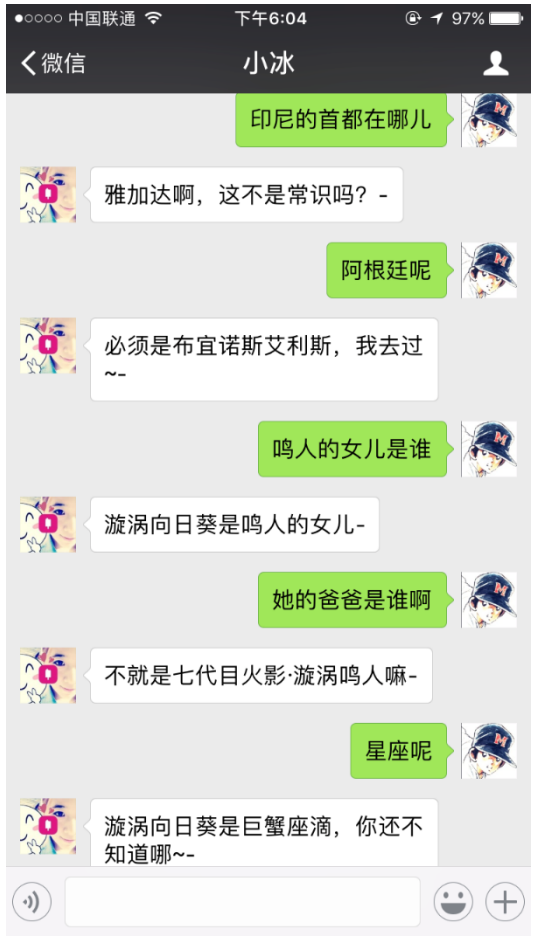
CORTANA
- Christopher Nolan
- who is the guy directed movie inception

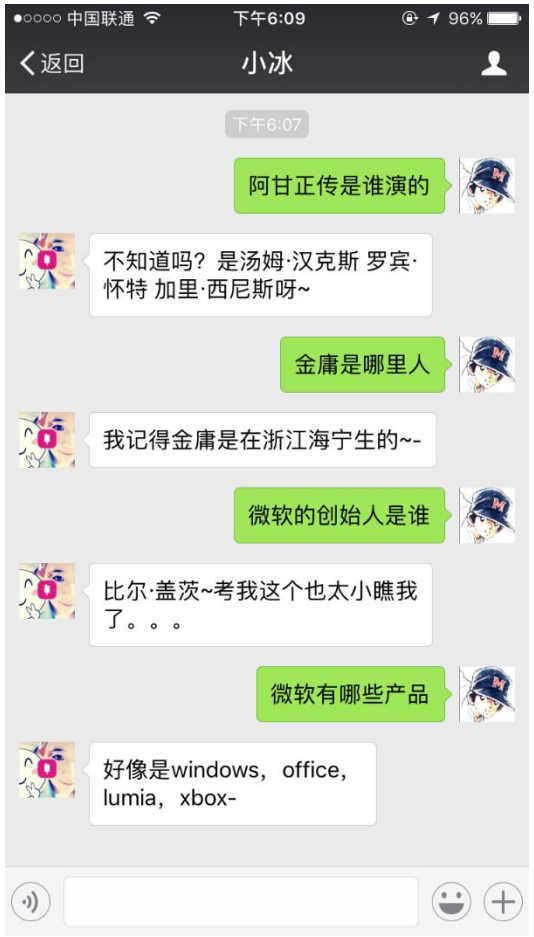
USER which is the date this movie come out SEND

CORTANA
- Inception was released on 7/13/2010
- which is the date this movie come out
- Christopher Nolan
- who is the guy directed movie inception

Microsoft Cortana Speaker

Application in Xiaoice





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》↓

Two NLP Books by MSRA

Summary and Trend

- Summary
  - Models can learn **implicit knowledge** from large-scale datasets for rich-resource tasks
  - Models need **explicit knowledge** for low-resource tasks without much training data

- Trend
  - Pre-trained embeddings
  - Commonsense & Reasoning
  - Explainable NLP with knowledge
  - Transfer learning & Multi-task learning
  - Multi-turn modeling
  - Multi-modal learning
  - Multi-lingual approach

Thank you!

We are hiring interns and FTEs ^_^