Introduction of Deep Learning for Language

2020/05/15 @tsujuifu

Tsu-Jui (Ray) Fu

- 1st PhD @ UCSB CS
- BS @ NTHU CS
- Vision-and-Language

Outline

- Natural Language Processing (NLP)
- Deep Learning for Language
- Research / Application on NLP

Outline

- Natural Language Processing (NLP)
  - What is NLP?
  - Sentiment Analysis
  - Traditional Text Feature for NLP
- Deep Learning for Language
- Research / Application on NLP

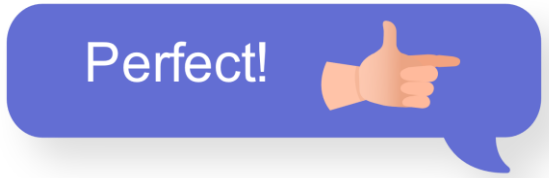
Outline

- Natural Language Processing (NLP)
  - What is NLP?
  - Sentiment Analysis
  - Traditional Text Feature for NLP
- Deep Learning for Language
- Research / Application on NLP

Natural Language Processing (NLP)

- Natural Language
  - speech and **text**
  - the most common way for human to **communicate**

speech



text

Natural Language Processing (NLP)

- Natural Language
- Machine Translation (**MT**), Question Answering (**QA**), Dialogue System (**DS**), …

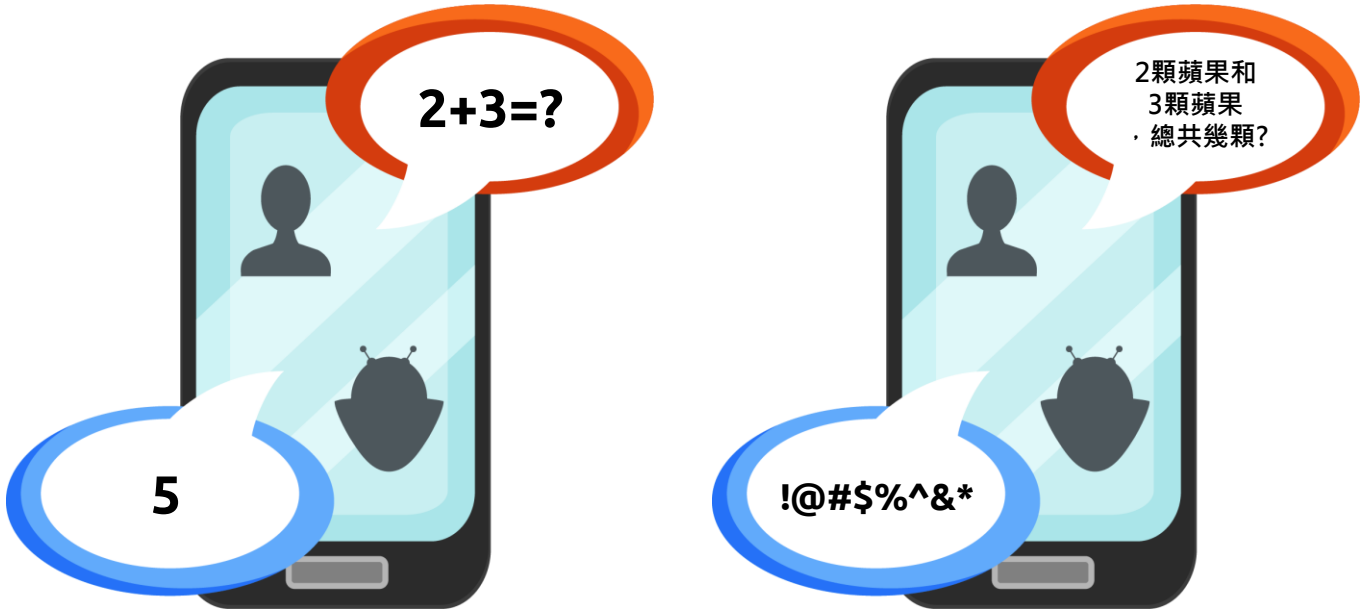
MT

QA

DS

Natural Language Processing (NLP)

- But NL is difficult for computer to “process”

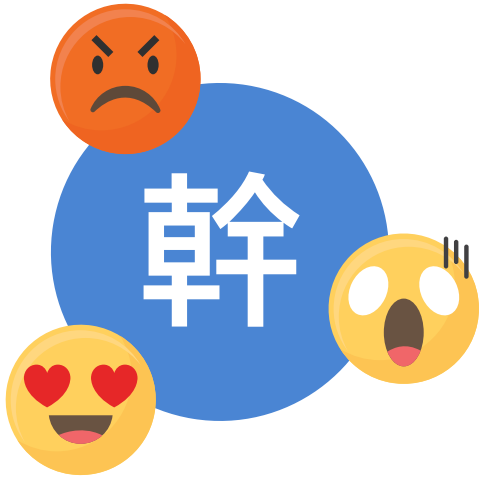


Natural Language Processing (NLP)

- But NL is difficult for computer to “process”

Natural Language Processing (NLP)

- But NL is difficult for computer to “process”
- Ambiguity, Homophonic, Humor, …



ambiguity

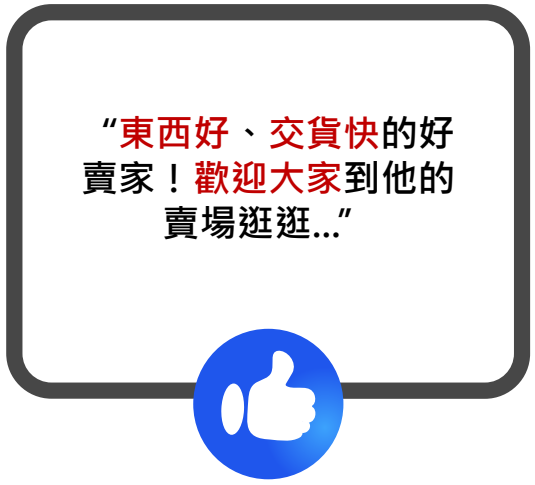
homophonic

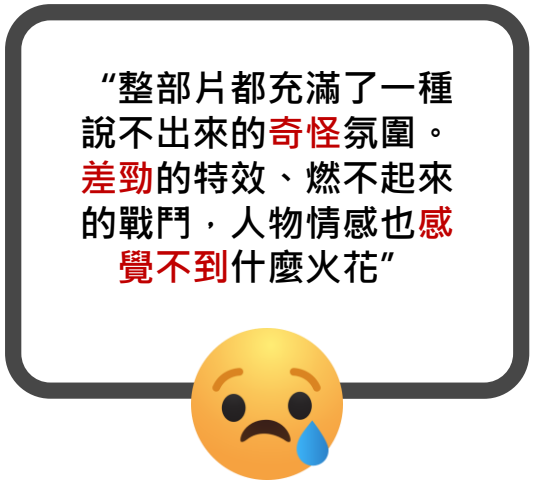
Outline

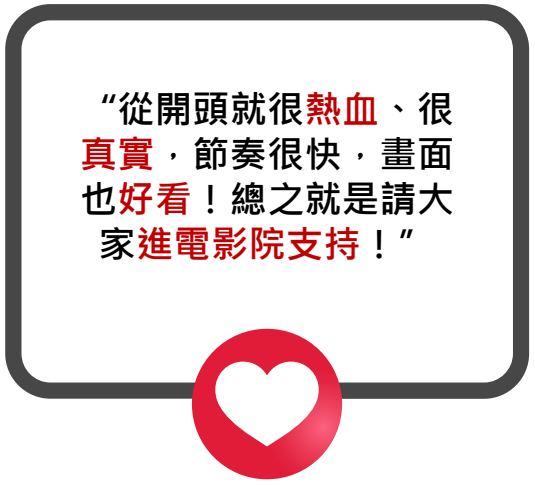
- Natural Language Processing (NLP)
  - What is NLP?
  - Sentiment Analysis
  - Traditional Text Feature for NLP
- Deep Learning for Language
- Research / Application on NLP

Sentiment Analysis

- **Emotion** of text data
  - identify customer sentiment toward products
  - analyze opinions on the social media

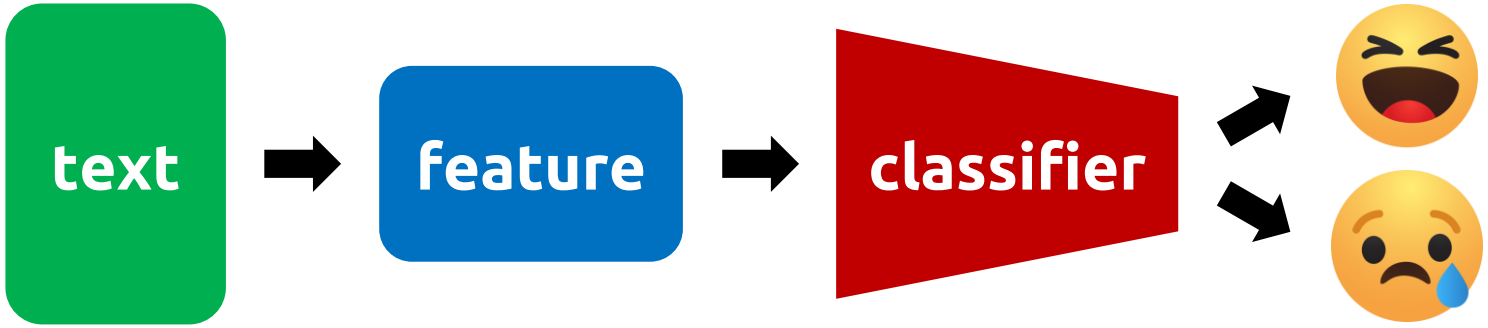






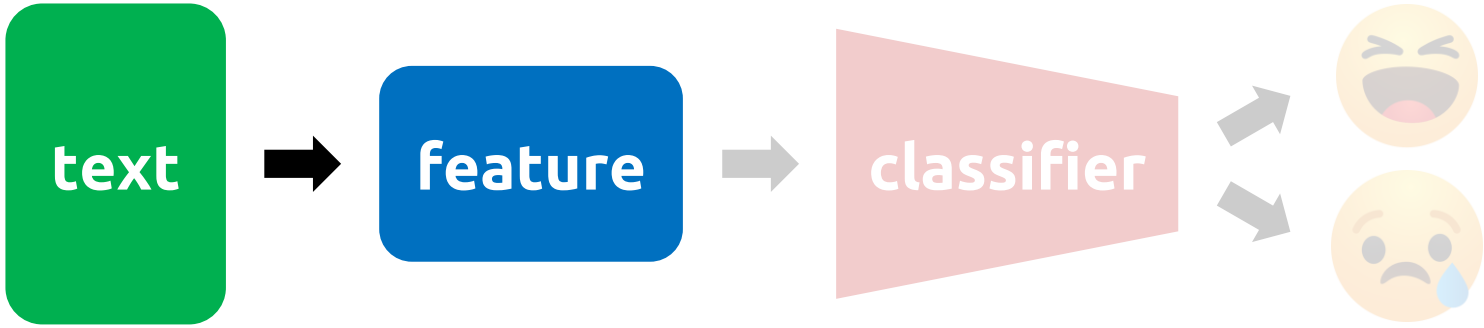
Sentiment Analysis

- Let's consider **binary** emotion (👍 vs 👎)



Sentiment Analysis

- Let's consider **binary** emotion (👍 vs 👎)



- How to transform **text into feature**?

Outline

- Natural Language Processing (NLP)
  - What is NLP?
  - Sentiment Analysis
  - Traditional Text Feature for NLP
- Deep Learning for Language
- Research / Application on NLP

Text Feature

- How to use a **vector** to represent the text?

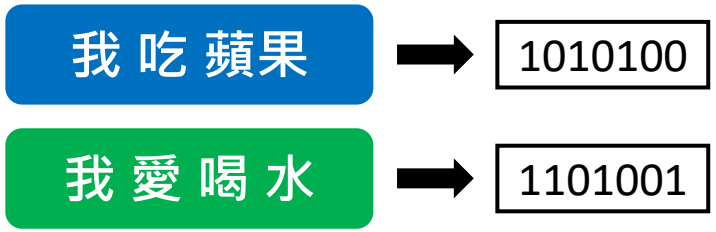
我 吃 蘋果

我 愛 喝 水

id	word	one-hot
0	我	1000000
1	愛	0100000
2	吃	0010000
3	喝	0001000
4	蘋果	0000100
5	香蕉	0000010
6	水	0000001

Text Feature

- How to use a **vector** to represent the text?
  - discrete **one-hot** vector (exist or not)



id	word	one-hot
0	我	1000000
1	愛	0100000
2	吃	0010000
3	喝	0001000
4	蘋果	0000100
5	香蕉	0000010
6	水	0000001

Text Feature

- How to use a **vector** to represent the text?
  - discrete **one-hot** vector (exist or not)

我 吃 蘋果 → 1010100

我 愛 喝 水 → 1101001

id	word	one-hot
0	我	1000000
1	愛	0100000
2	吃	0010000
3	喝	0001000
4	蘋果	0000100
5	香蕉	0000010
6	水	0000001

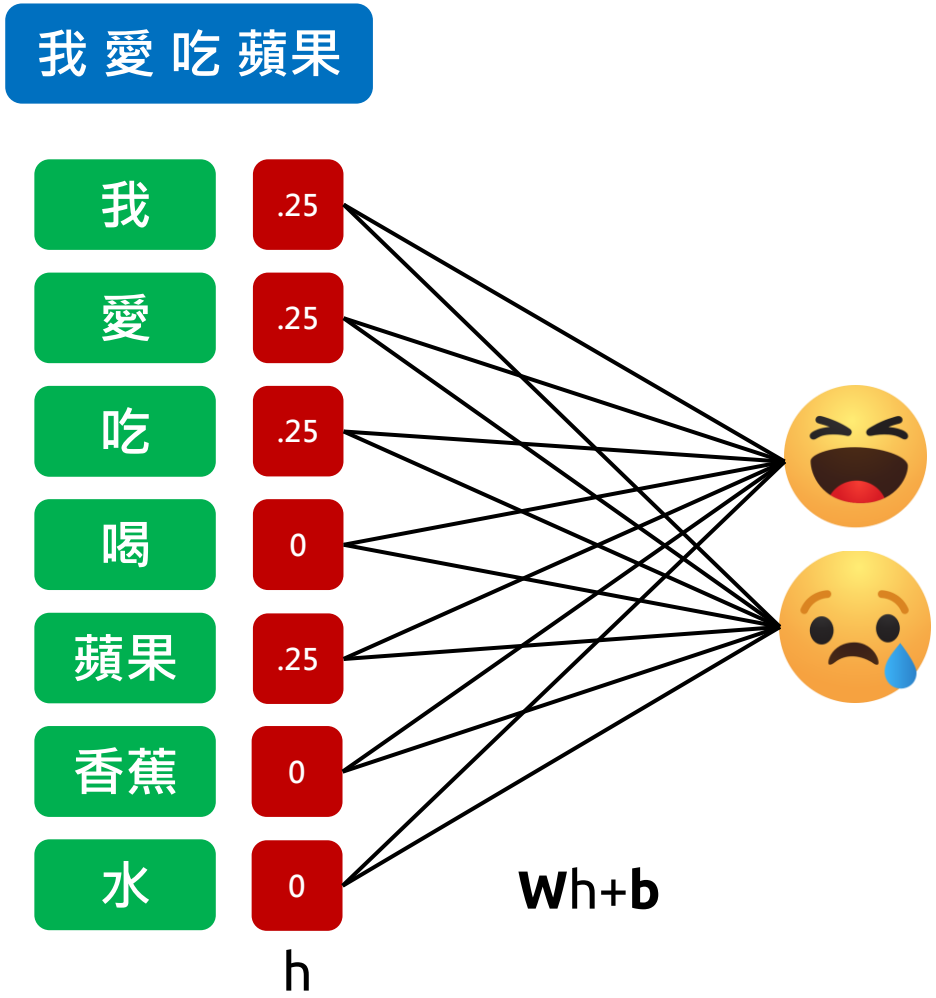
  - w/ term-**frequency**

我 愛 喝 水 → .25 .25 0 .25 0 0 .25

我 愛 喝 水 水 → .2 .2 0 .2 0 0 **.4**

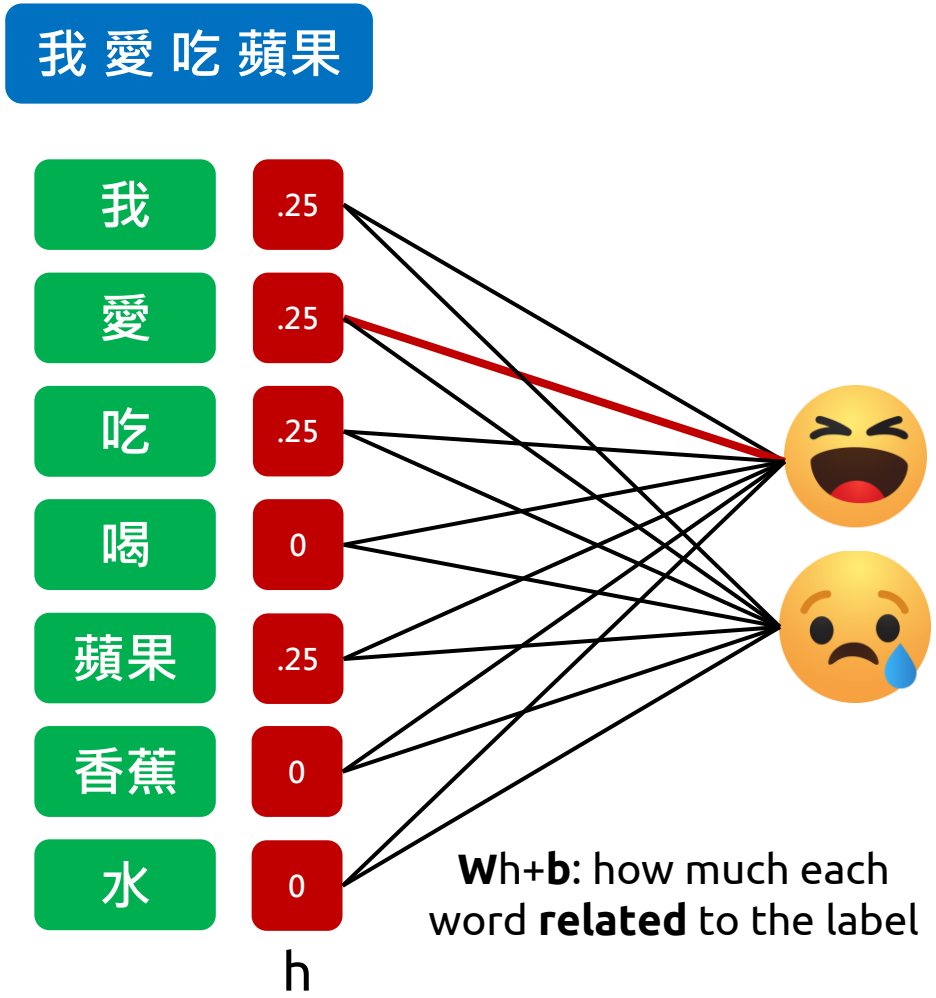
Classification by Text Feature

- Classifier (**W**h+**b**)



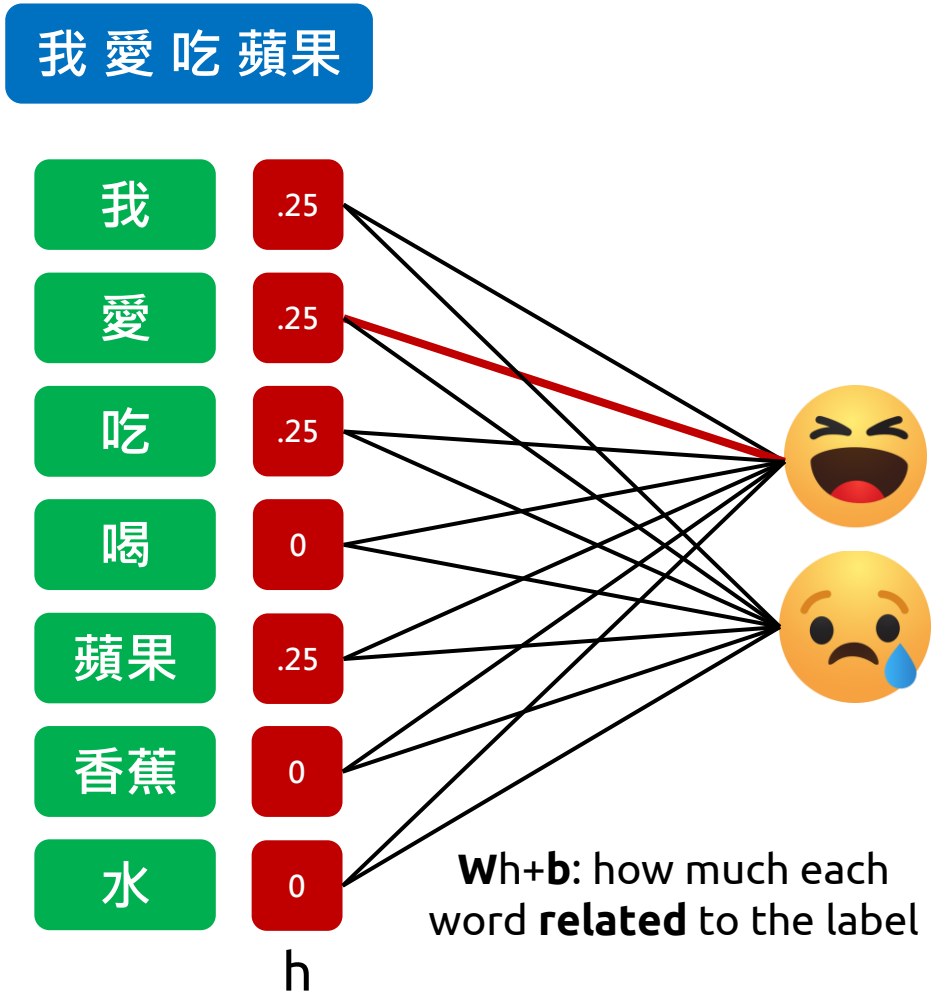
Classification by Text Feature

- Classifier (**W**h+**b**)



Classification by Text Feature

- Classifier (**W**h+**b**)



Problem of Traditional Text Feature

- **Fixed** vocabulary space
  - **inflexible** when adding new word

$W \in 2x3$

$W \in 2x4$

- **Too Large** text feature size
  - **>2M** vocabulary in ENG

- Neglect the **word order**
  - “i love u” vs “u love i”

Outline

- Natural Language Processing (NLP)

- Deep Learning for Language
  - Word Vector (word2vec)
  - Recurrent Neural Network (RNN)
  - Long Short-term Memory (LSTM)
  - Sequence-to-Sequence (seq2seq)

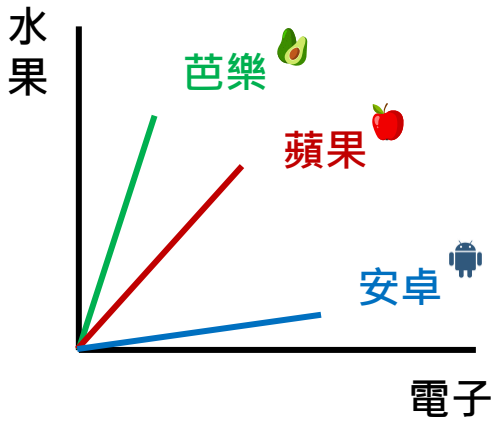
- Research / Application on NLP

Outline

- Natural Language Processing (NLP)
- Deep Learning for Language
  - Word Vector (word2vec)
  - Recurrent Neural Network (RNN)
  - Long Short-term Memory (LSTM)
  - Sequence-to-Sequence (seq2seq)
- Research / Application on NLP

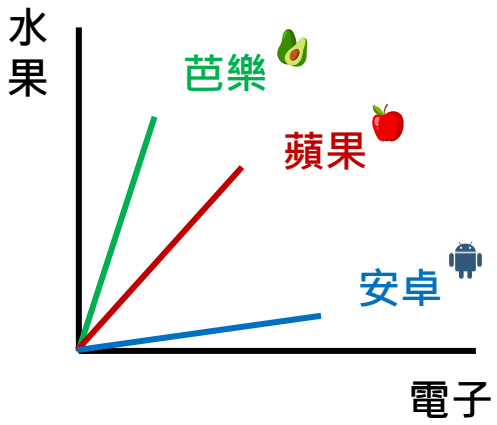
Word Vector (word2vec)

- word → **vector**
  - each dimension contains **hidden semantic**
  - consider **relative** between words on vector space



Word Vector (word2vec)

- word → **vector**
  - each dimension contains **hidden semantic**
  - consider **relative** between words on vector space



	one-hot	word2vec
vector size	unfixed (vocabulary)	**fixed** (ex: 300d)
consider relative	✘	✔

Word Vector (word2vec)

- How to get word vector?
  - word **shares similar context** has similar vector

i bought apple in the market

i bought guava in the market

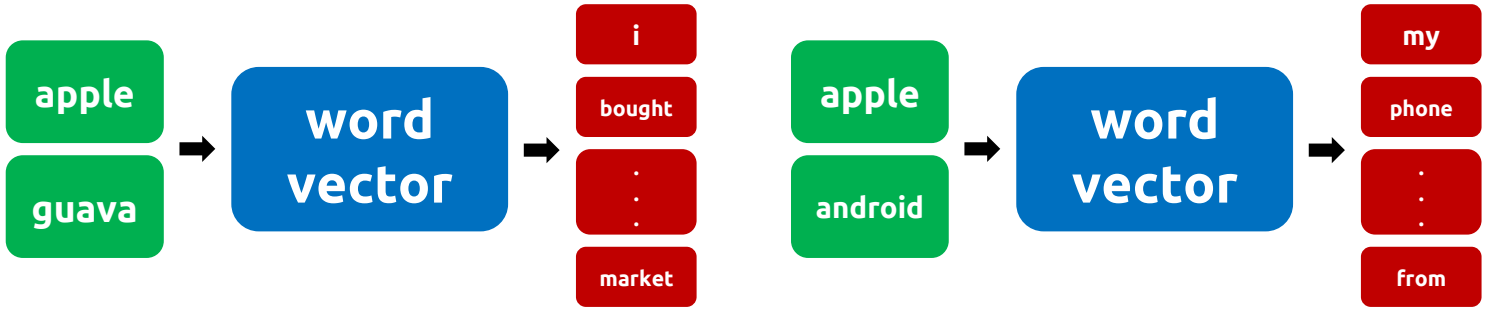
Word Vector (word2vec)

- How to get word vector?
  - word **shares similar context** has similar vector

Word Vector (word2vec)

- How to get word vector?
  - word **shares similar context** has similar vector

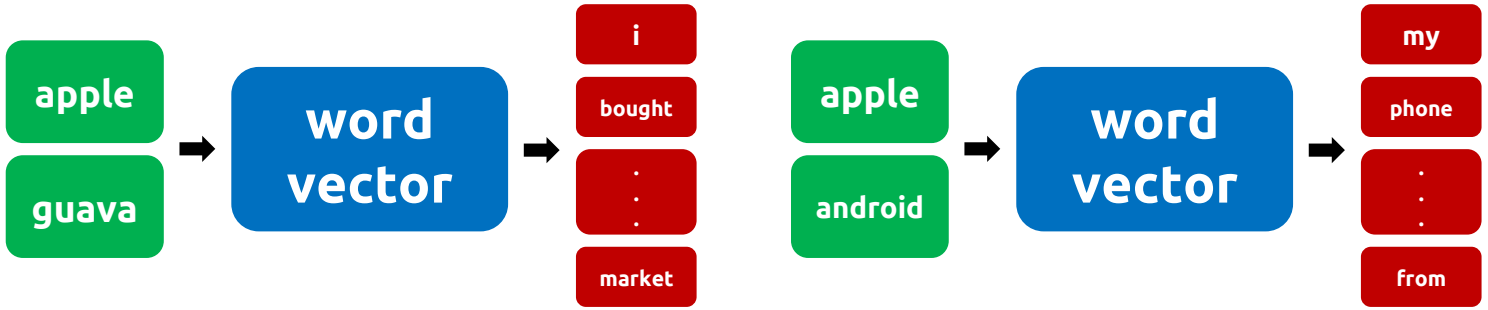
  - **skip-gram** (predict **near** word)



Word Vector (word2vec)

- How to get word vector?
  - word **shares similar context** has similar vector

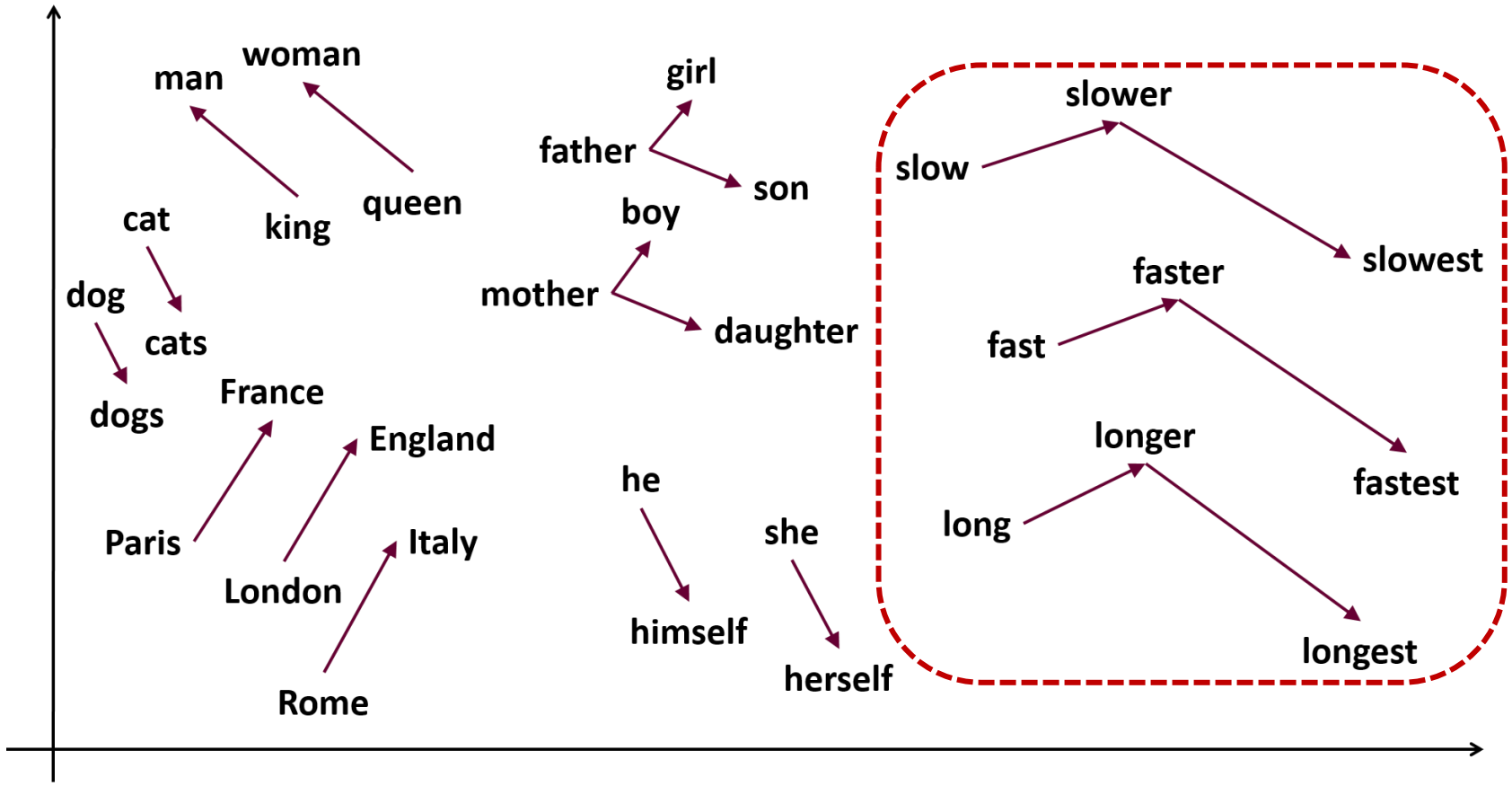
  - **skip-gram** (predict **near** word)



  - need lots of text to train (Wiki, common crawl, …)

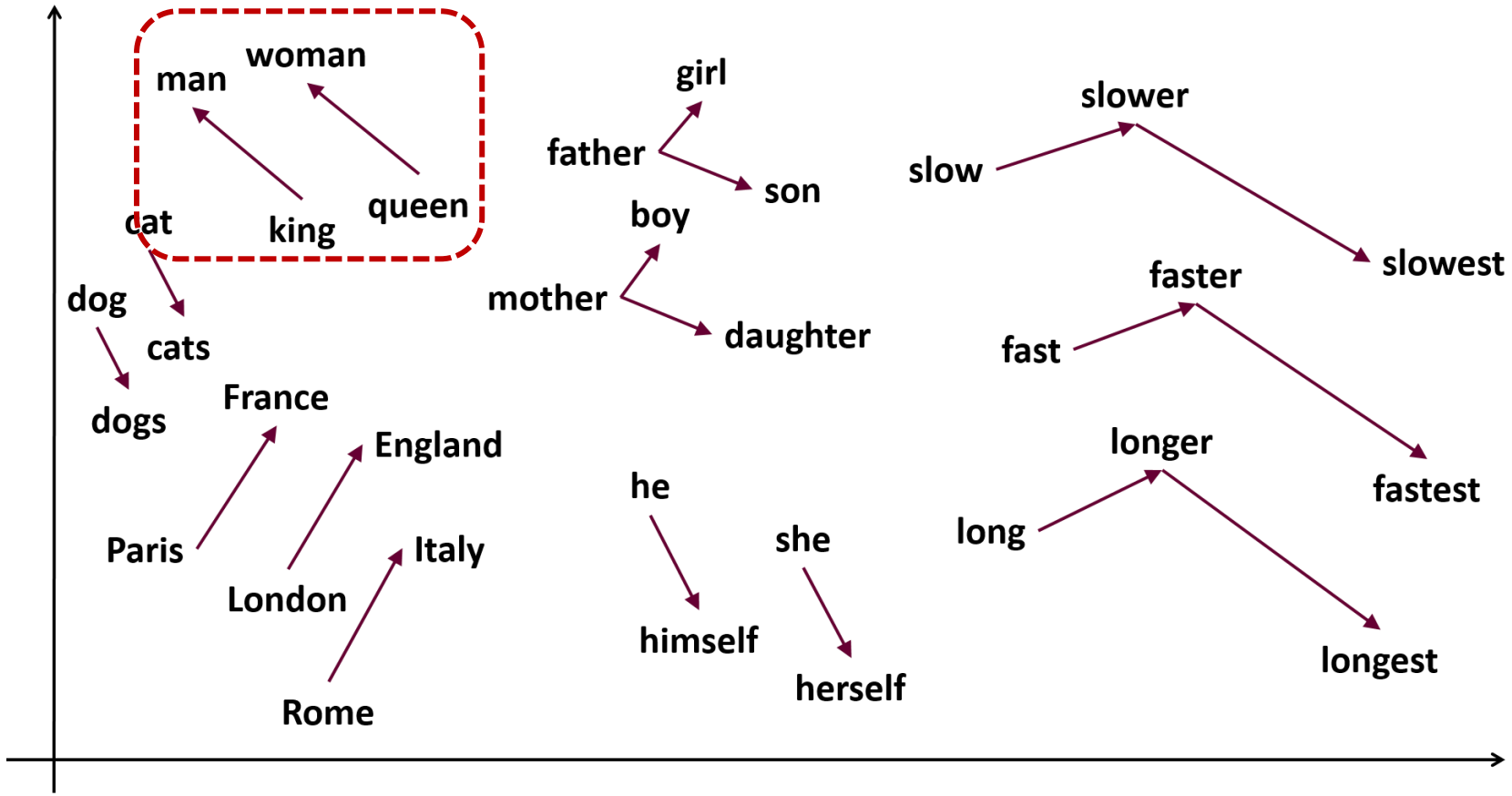
Word Vector (word2vec)

- reflect **semantic** of words on vector space



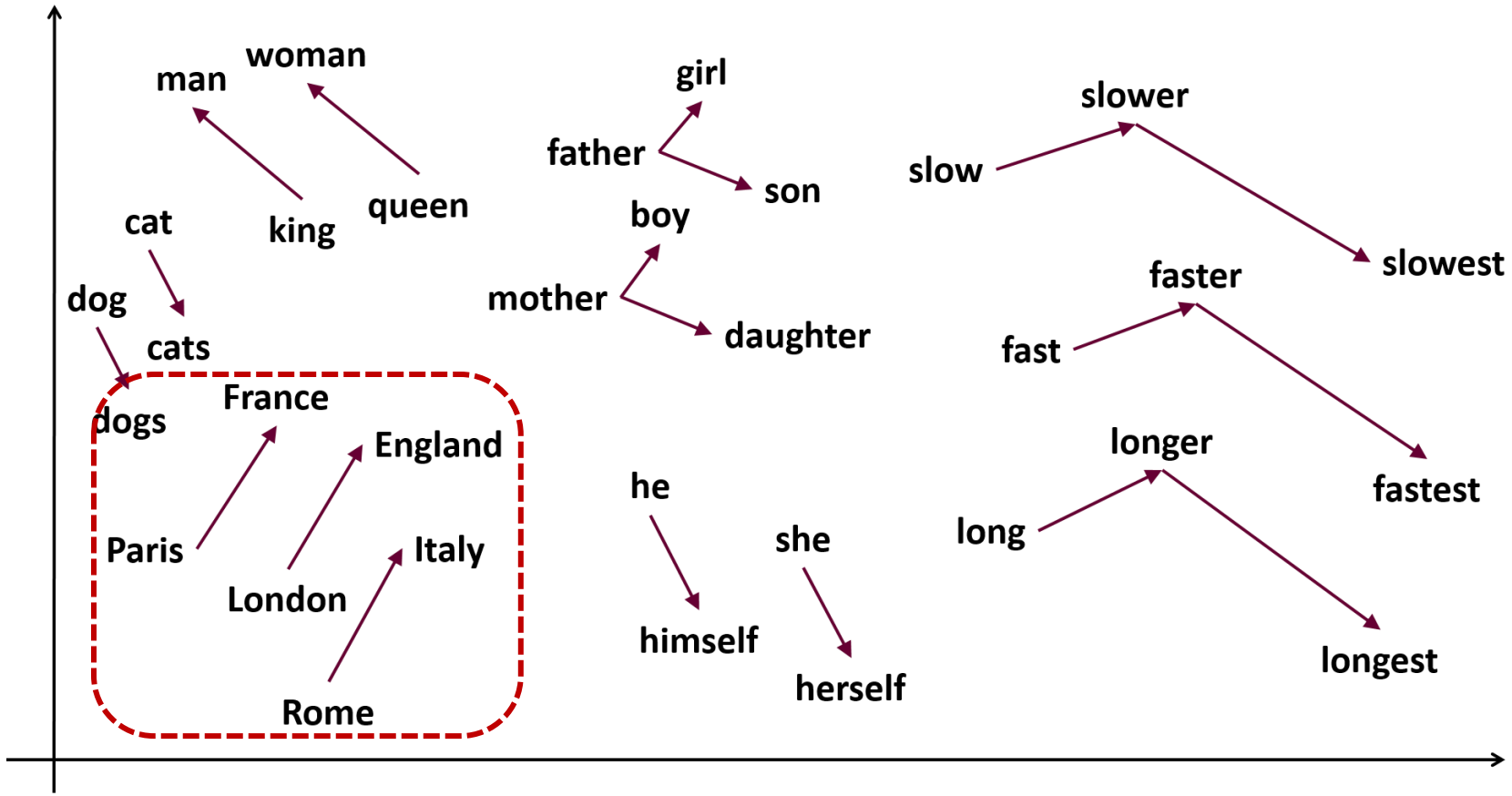
Word Vector (word2vec)

- reflect **semantic** of words on vector space



Word Vector (word2vec)

- reflect **semantic** of words on vector space



Word Vector (word2vec)

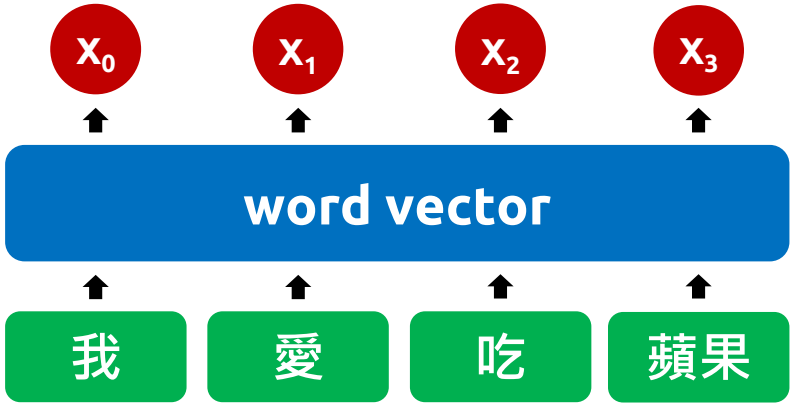
- Now, we have vector to represent each word
- But how about **the whole text**?

Outline

- Natural Language Processing (NLP)
- Deep Learning for Language
  - Word Vector (word2vec)
  - Recurrent Neural Network (RNN)
  - Long Short-term Memory (LSTM)
  - Sequence-to-Sequence (seq2seq)
- Research / Application on NLP

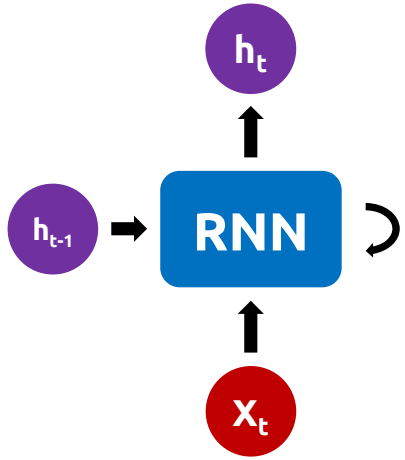
Recurrent Neural Network (RNN)

- With word vector, how to represent the whole text?



Recurrent Neural Network (RNN)

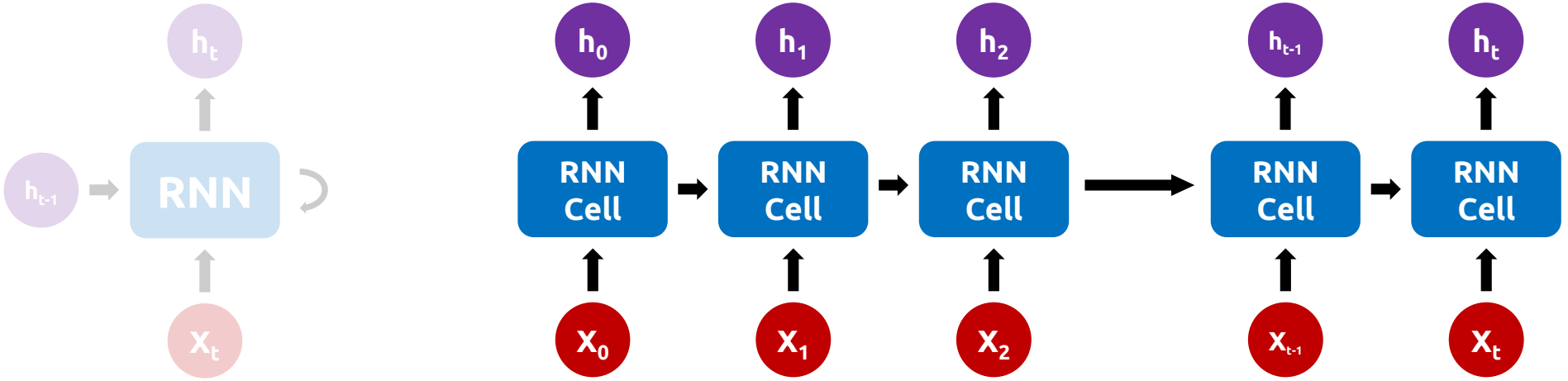
- Consider “order” to model **sequential** data



$$h_t = \sigma(\boldsymbol{W}x_t + \boldsymbol{U}h_{t-1} + \boldsymbol{b})$$

Recurrent Neural Network (RNN)

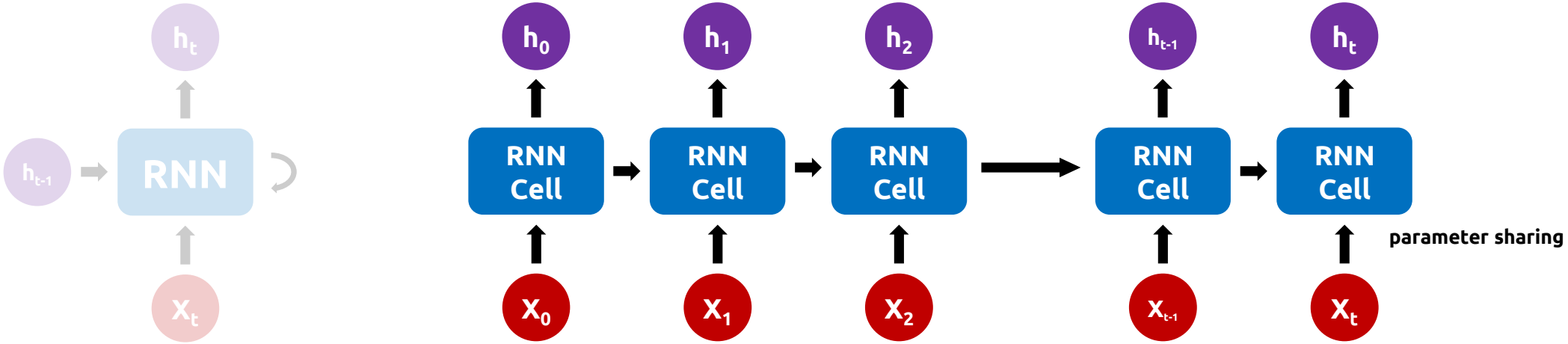
- Consider “order” to model **sequential** data



$$h_t = \sigma(\boldsymbol{W}x_t + \boldsymbol{U}h_{t-1} + \boldsymbol{b})$$

Recurrent Neural Network (RNN)

- Consider "order" to model **sequential** data

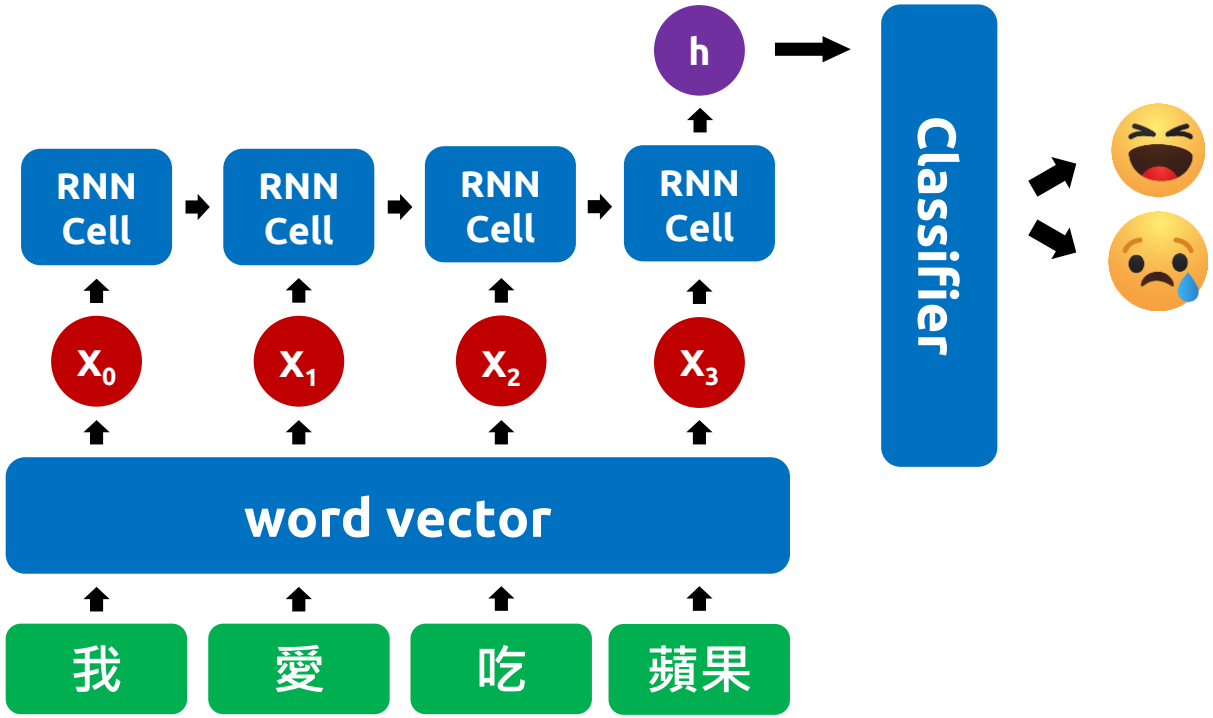


$$h_t = \sigma(\boldsymbol{W}x_t + \boldsymbol{U}h_{t-1} + \boldsymbol{b})$$

- h_t represents $\mathbf{x_{0..t}}$
- RNN Cell shares same parameter (ex: *W, U, b*)

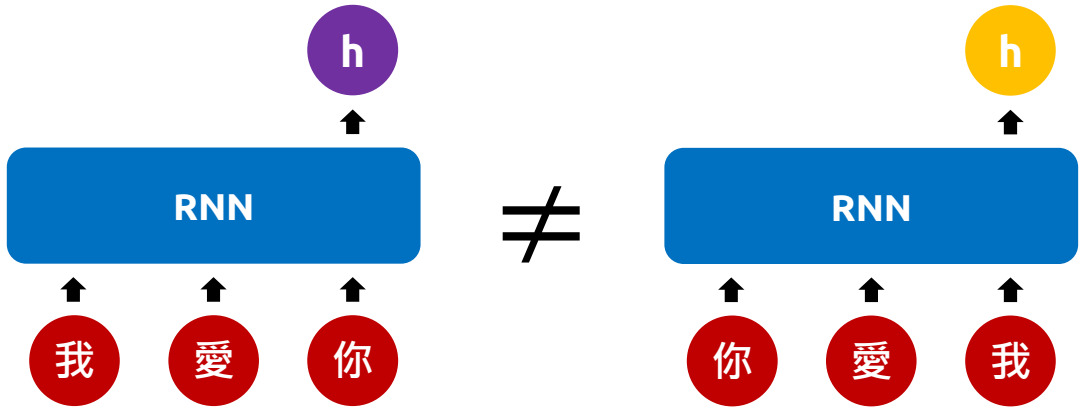
Recurrent Neural Network (RNN)

- Consider “order” to model **text**
  - apply W2V and RNN into sentiment classifier



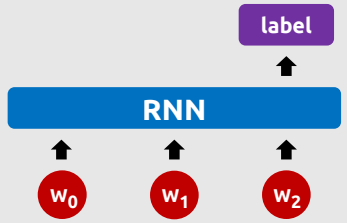
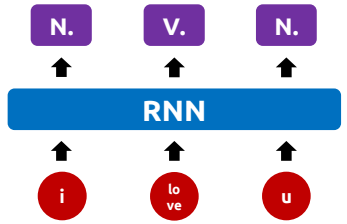
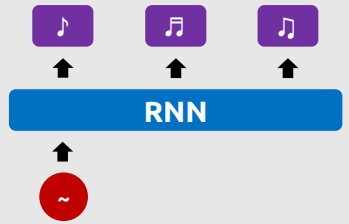
Recurrent Neural Network (RNN)

- Consider “order” to model **text**
  - different orders come out different results



Recurrent Neural Network (RNN)

- Different types of RNN

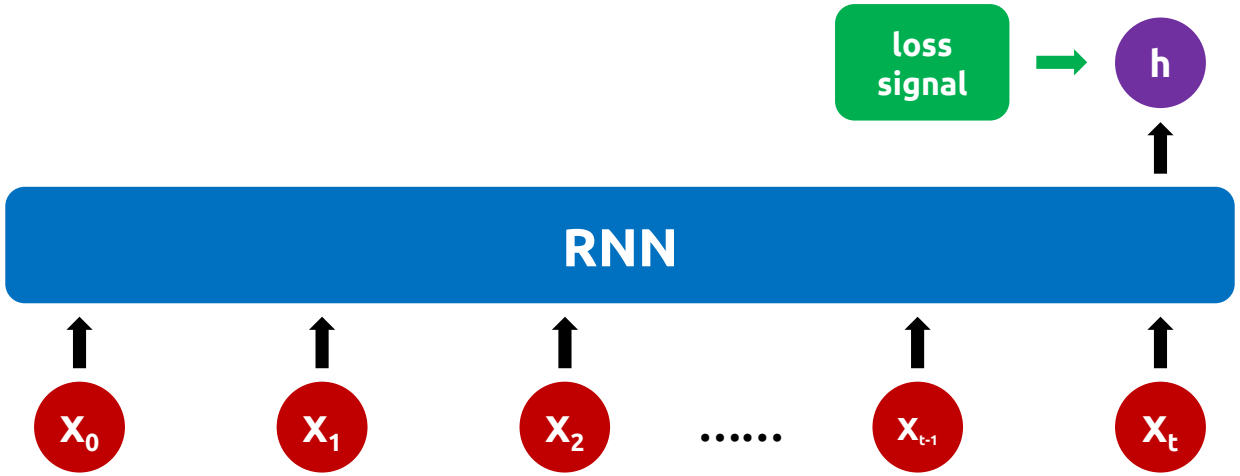
Type	Model	Example
many-to-one		text classification
many-to-many		POS tagging
one-to-many		music generation

Problem of Vanilla RNN

- Gradient vanishing
  - ideally, RNN can model infinite long sequence 🤔

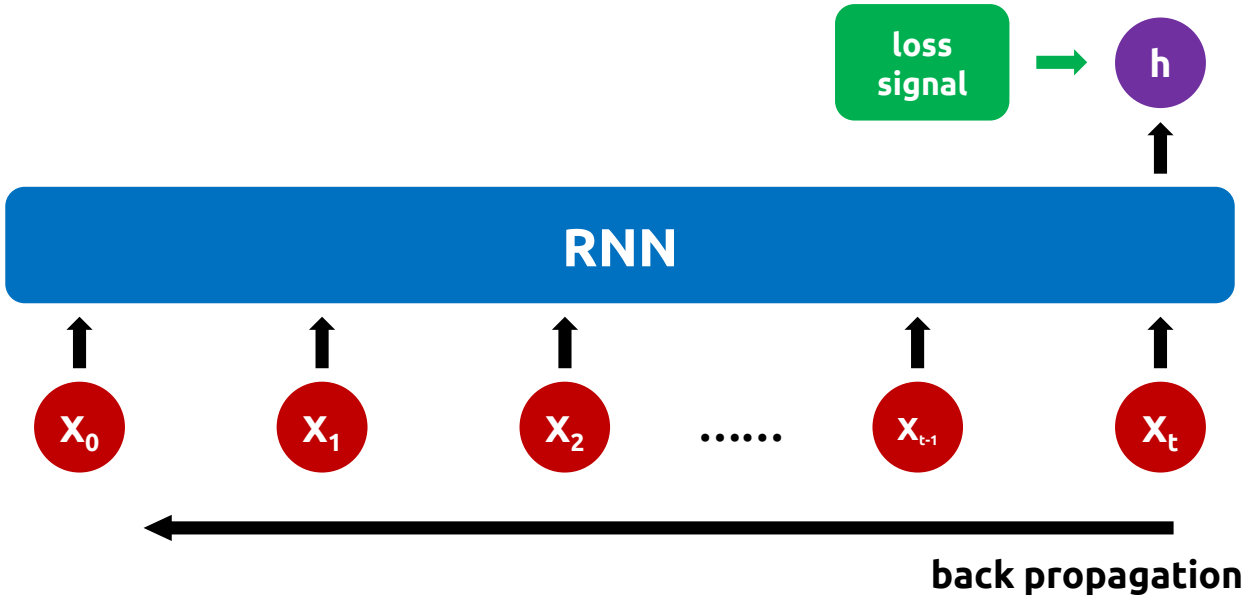
Problem of Vanilla RNN

- Gradient vanishing
  - ideally, RNN can model infinite long sequence 🤔
  - however, gradient will be limited if **too long** 😰



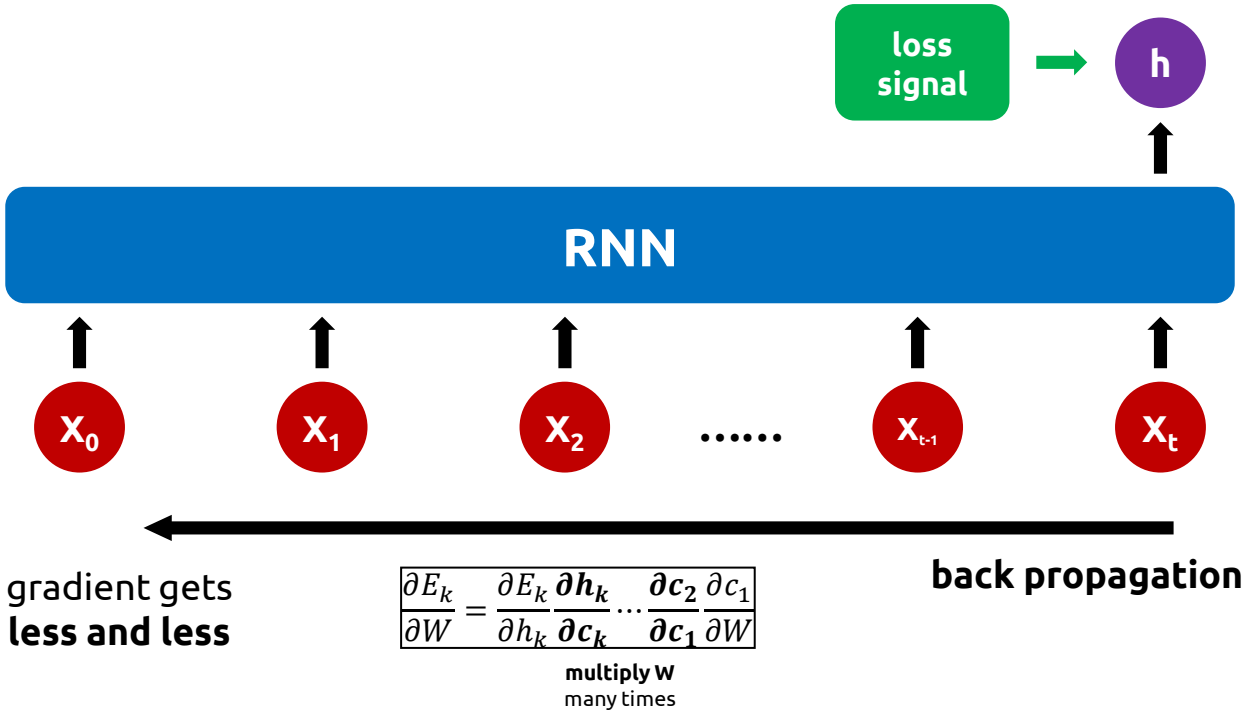
Problem of Vanilla RNN

- Gradient vanishing
  - ideally, RNN can model infinite long sequence 🤔
  - however, gradient will be limited if **too long** 😰



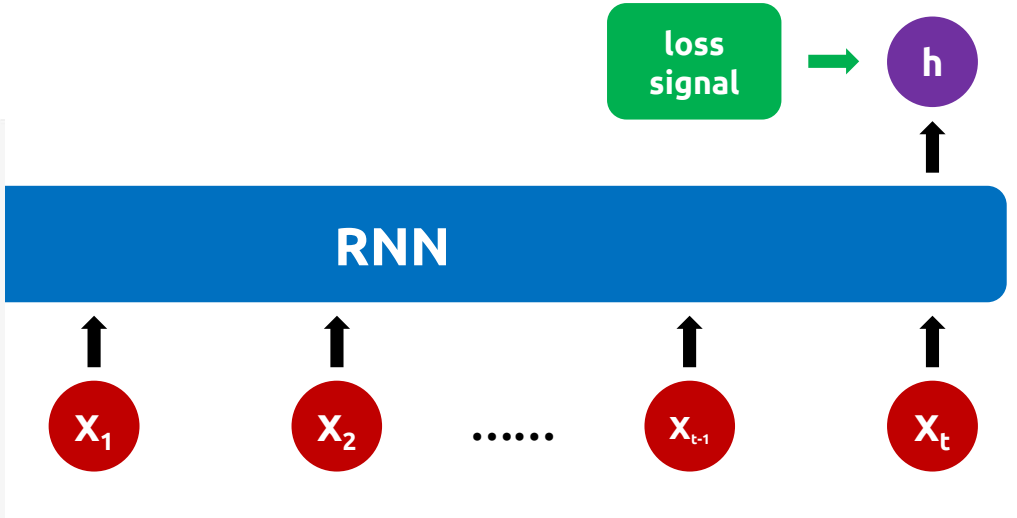
Problem of Vanilla RNN

- Gradient vanishing
  - ideally, RNN can model infinite long sequence 🤔
  - however, gradient will be limited if **too long** 😰



Problem of Vanilla RNN

- Gradient vanishing
  - ideally, RNN can model infinite long sequence 🤔
  - however, gradient will be limited if **too long** 😰



$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial h_k}\frac{\partial \boldsymbol{h_k}}{\partial \boldsymbol{c_k}}\cdots\frac{\partial \boldsymbol{c_2}}{\partial \boldsymbol{c_1}}\frac{\partial c_1}{\partial W}$$

multiply W
many times


Break 10 min

Outline

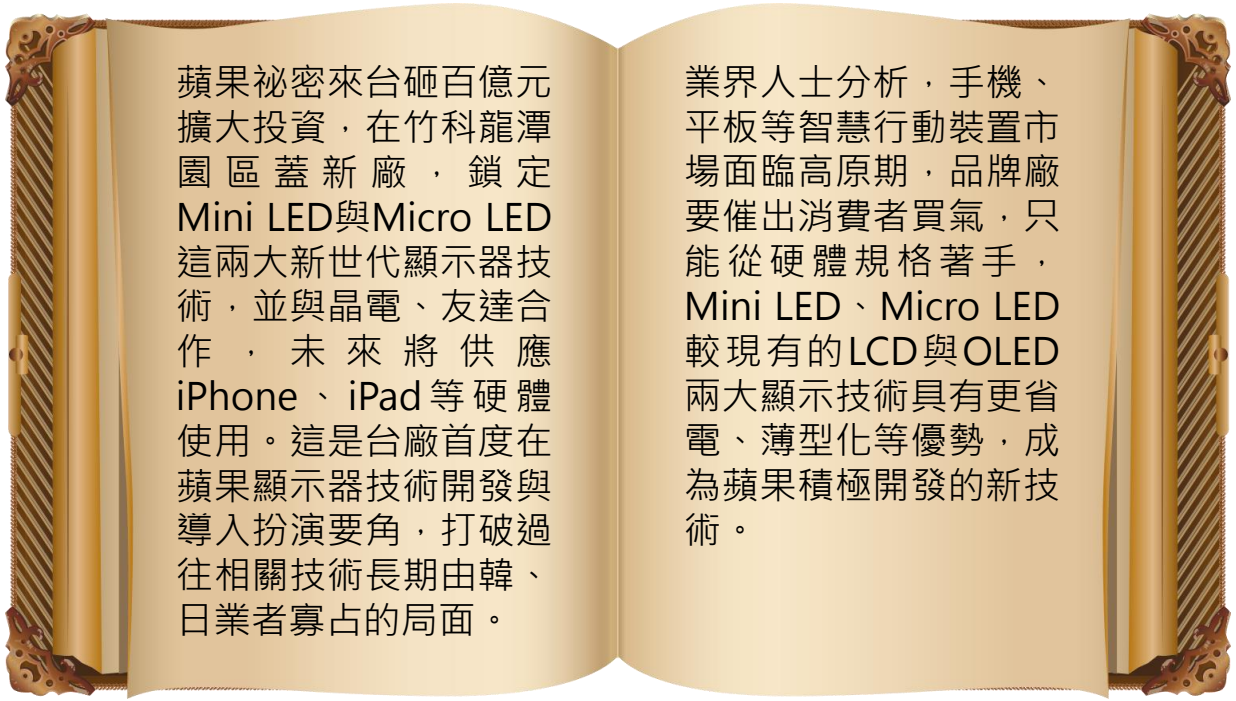
- Natural Language Processing (NLP)
- Deep Learning for Language
  - Word Vector (word2vec)
  - Recurrent Neural Network (RNN)
  - Long Short-term Memory (LSTM)
  - Sequence-to-Sequence (seq2seq)
- Research / Application on NLP

Long Short-Term Memory (LSTM)

- Vanilla RNN suffers from **gradient vanishment**
- Let’s rethink about how we read?

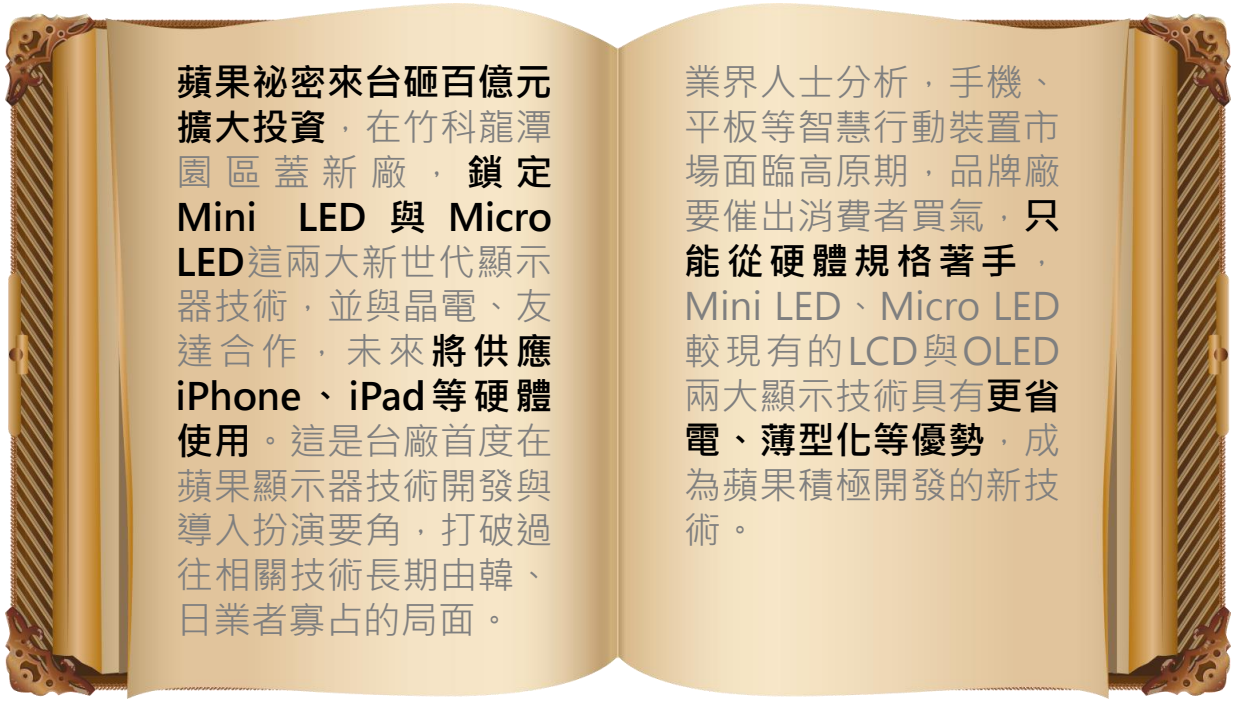
Long Short-Term Memory (LSTM)

- Vanilla RNN suffers from **gradient vanishment**
- Let’s rethink about how we read?



Long Short-Term Memory (LSTM)

- Vanilla RNN suffers from **gradient vanishment**
- Let's rethink about how we read?



Long Short-Term Memory (LSTM)

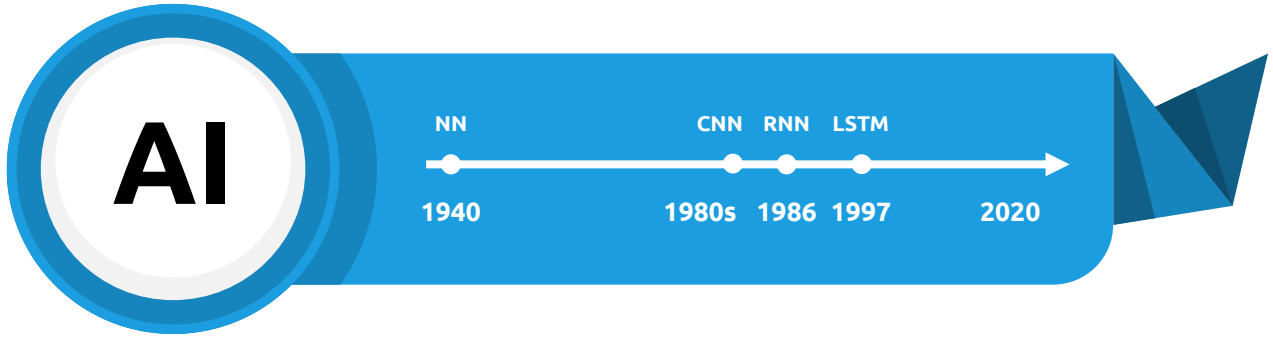
- Let’s rethink about how we read?
  - **forget** unimportant part
  - but still maintain an understanding

- Long Short-Term Memory (1997)
  - integrate “forgetness” into RNN

Long Short-Term Memory (LSTM)

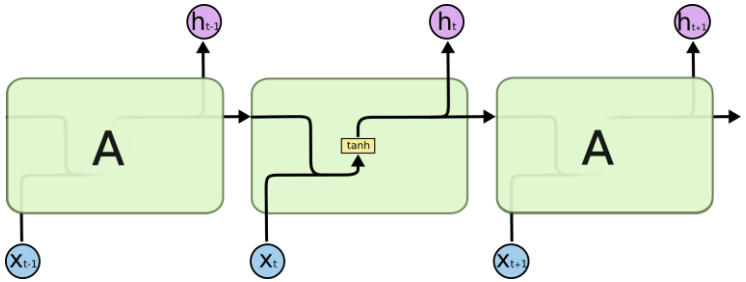
- Let’s rethink about how we read?
  - **forget** unimportant part
  - but still maintain an understanding

- Long Short-Term Memory (1997)
  - integrate “forgetness” into RNN

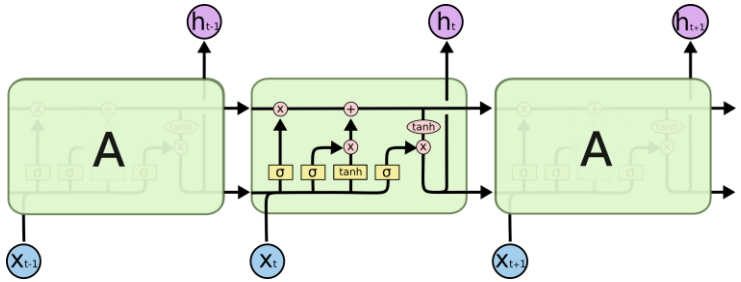


Long Short-Term Memory (LSTM)

- Long Short-Term Memory (1997)
  - integrate “forgetness” into RNN



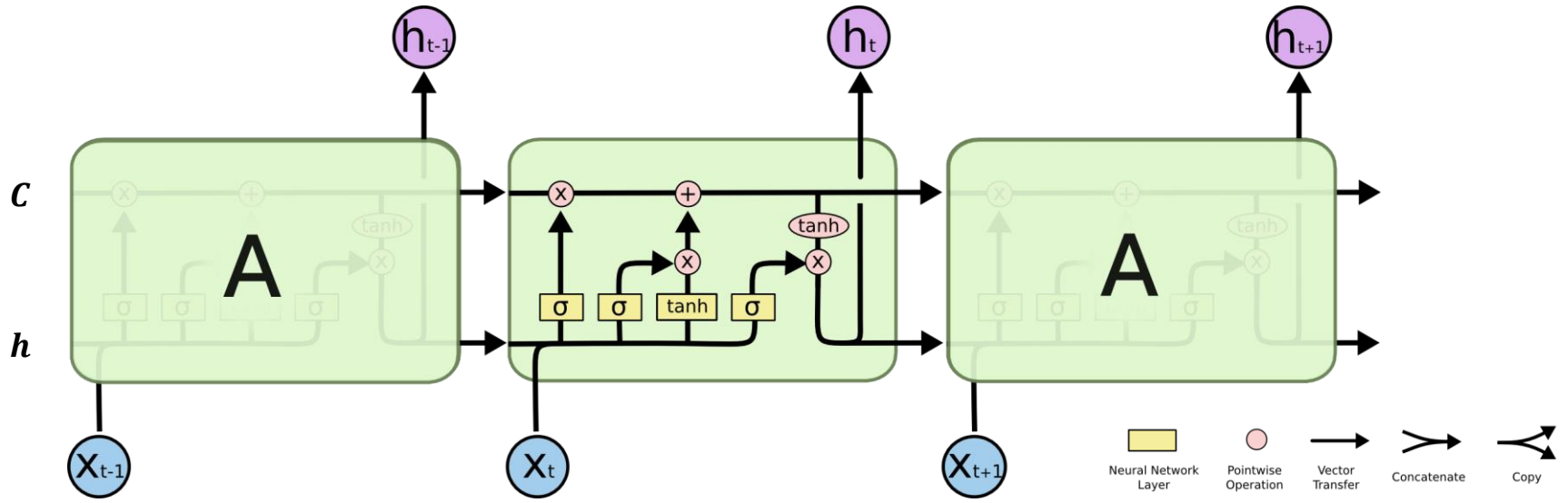
Vanilla RNN



LSTM

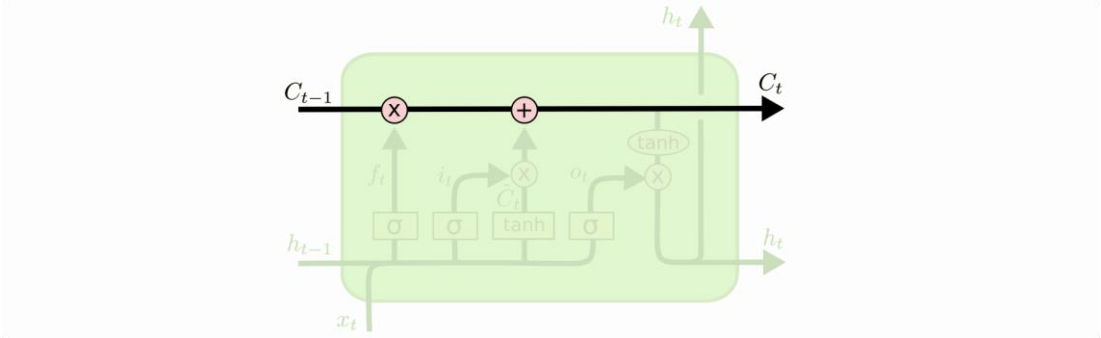
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

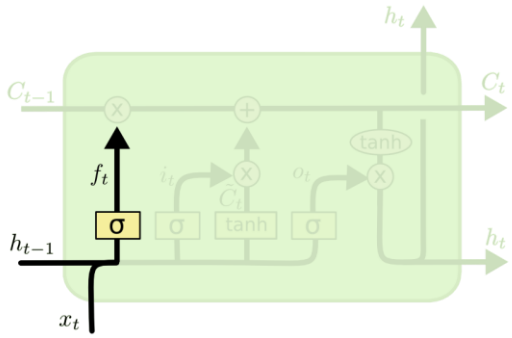


- $\boldsymbol{h}$: hidden state as vanilla RNN
  - changes a lot with time
  - **short**-term

- $\boldsymbol{C}$: cell state
  - changes a little (only some linear operation)
  - **long**-term

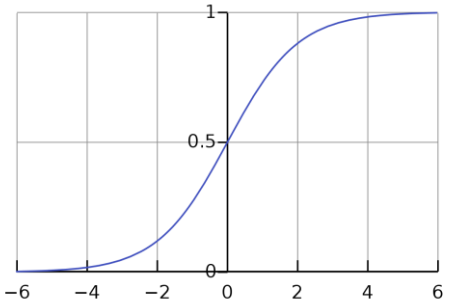
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)



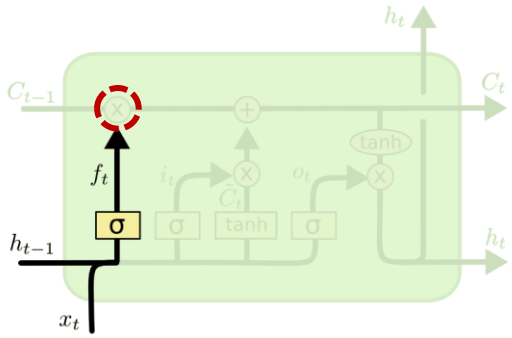
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

- Step 1: **forget**
  - f_t: scalar after **sigmoid** function (**[0~1]**)

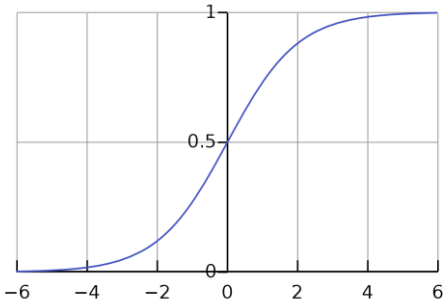


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)



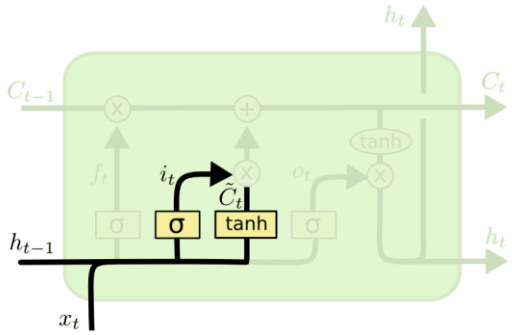
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$



- Step 1: **forget**
  - f_t: value after sigmoid function ([0~1])
  - **multiply with C**: how much to forget

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

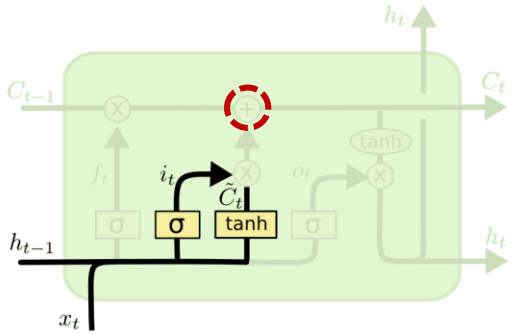


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

- Step 2: **memorize**
  - $\sim \boldsymbol{C_t}$: what to memorize
  - $\boldsymbol{i_t}$: how much to memorize

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

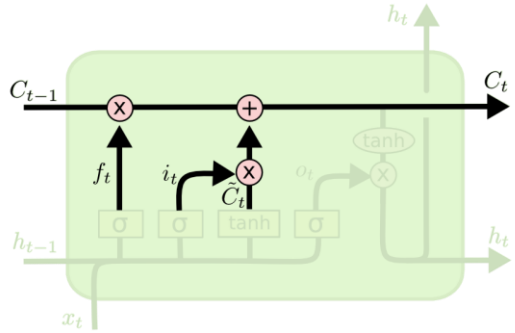


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

- Step 2: **memorize**
  - $\sim C_t$: what to memorize
  - i_t: how much to memorize
  - **add with C**: to memorize new information

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

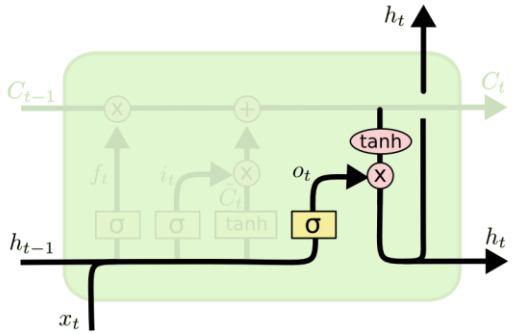


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- $\boldsymbol{C_t}$:
  - forget by $\boldsymbol{f_t}$
  - memorize by $\boldsymbol{i_t}$ and $\sim\boldsymbol{C_t}$

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

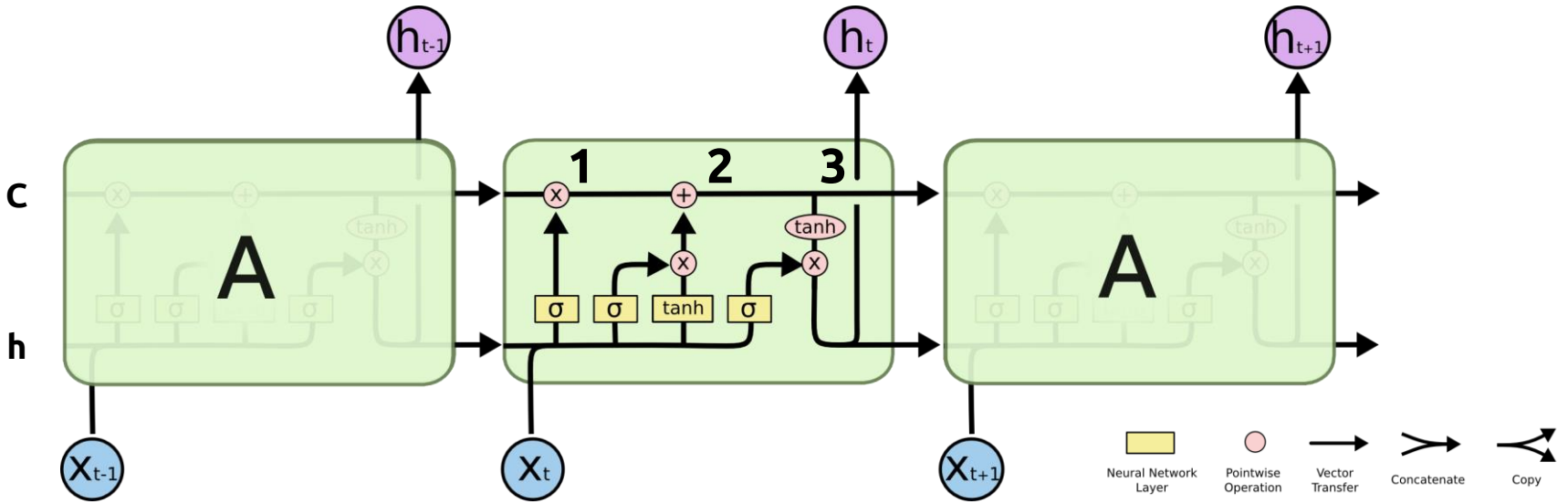


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

- $\boldsymbol{h_t}$:
  - similar as vanilla RNN
  - also based on long-term $\boldsymbol{C}$

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

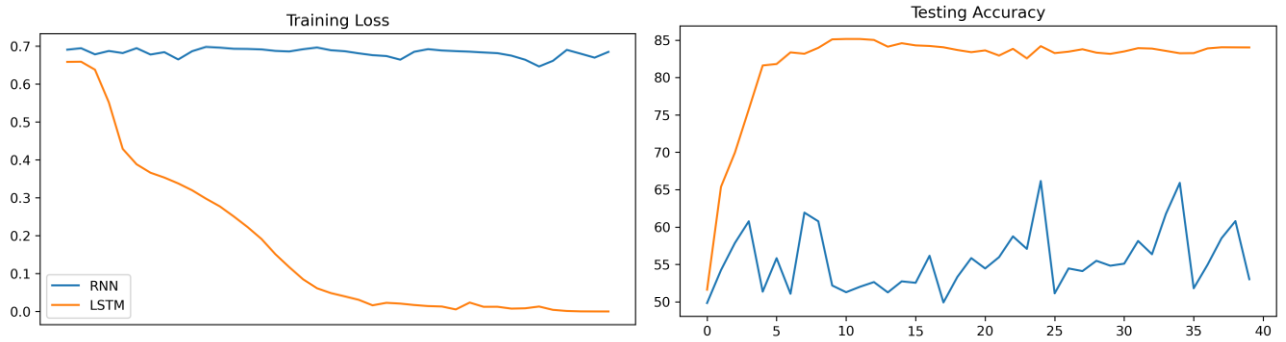


1. **Forget** long-term $\boldsymbol{C}$
2. **Update** long-term $\boldsymbol{C}$
3. **Decide** short-term $\boldsymbol{h}$

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM)

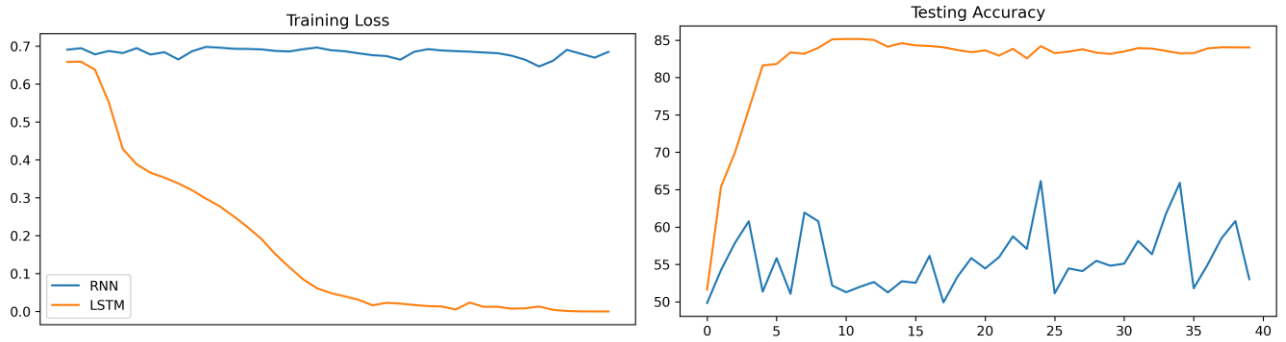
- Binary Sentiment Analysis



- Training Loss:
  - RNN suffers from gradient vanishment
  - LSTM keeps reducing loss
- Testing Accuracy:
  - LSTM achieves 85% after 10 epochs

Long Short-Term Memory (LSTM)

- Binary Sentiment Analysis



- Training Loss:
  - RNN suffers from gradient vanishment
  - LSTM keeps reducing loss
- Testing Accuracy:
  - LSTM achieves 85% after 10 epochs

https://medium.com/datadriveninvestor/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577

Outline

- Natural Language Processing (NLP)
- Deep Learning for Language
  - Word Vector (word2vec)
  - Recurrent Neural Network (RNN)
  - Long Short-term Memory (LSTM)
  - Sequence-to-Sequence (seq2seq)
- Research / Application on NLP

Sequence-to-Sequence (seq2seq)

- With RNN (or LSTM), we can model a sequence
- Now, we want (***seq*** → ***seq'***)

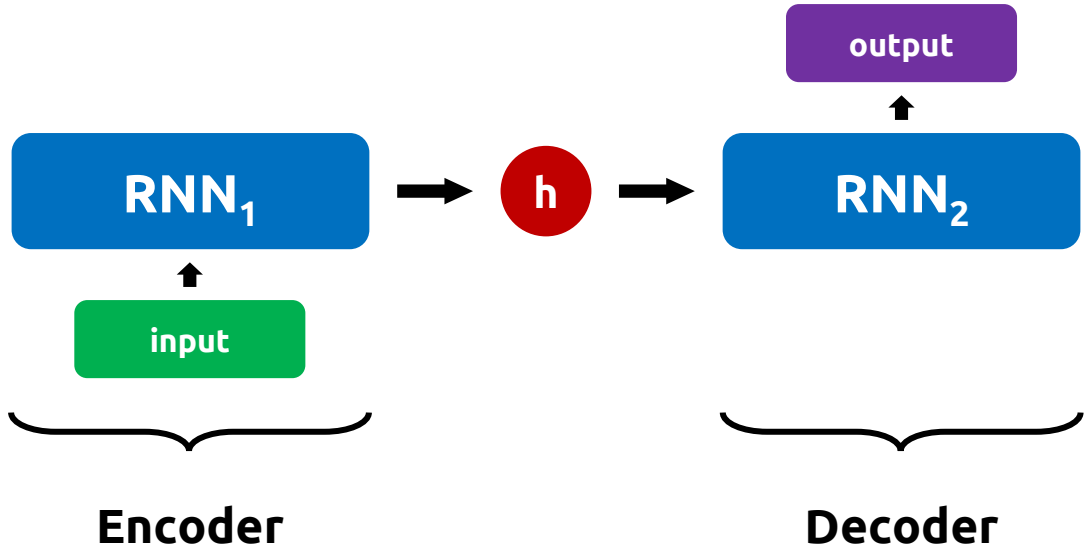
Sequence-to-Sequence (seq2seq)

- With RNN (or LSTM), we can model a sequence
- Now, we want (***seq*** → ***seq'***)

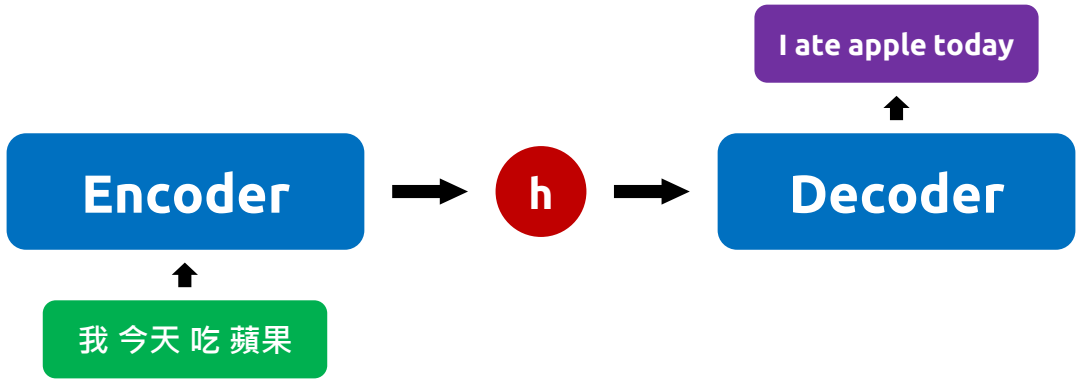
  - application:
    - machine translation: (sentence → sentence)
    - question answering: (paragraph → sentence)
    - text summarization: (article → paragraph)
    - …

Sequence-to-Sequence (seq2seq)

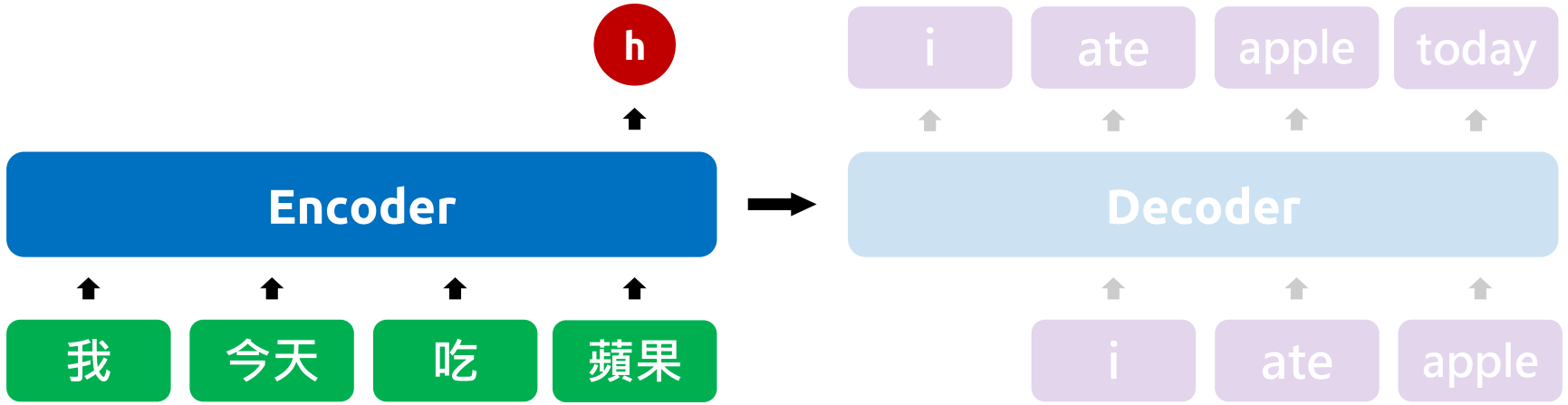
- Combine 2 RNNs
  - RNN_1: **encodes** input into h
  - RNN_2: **decodes** output based on h



Sequence-to-Sequence (seq2seq)

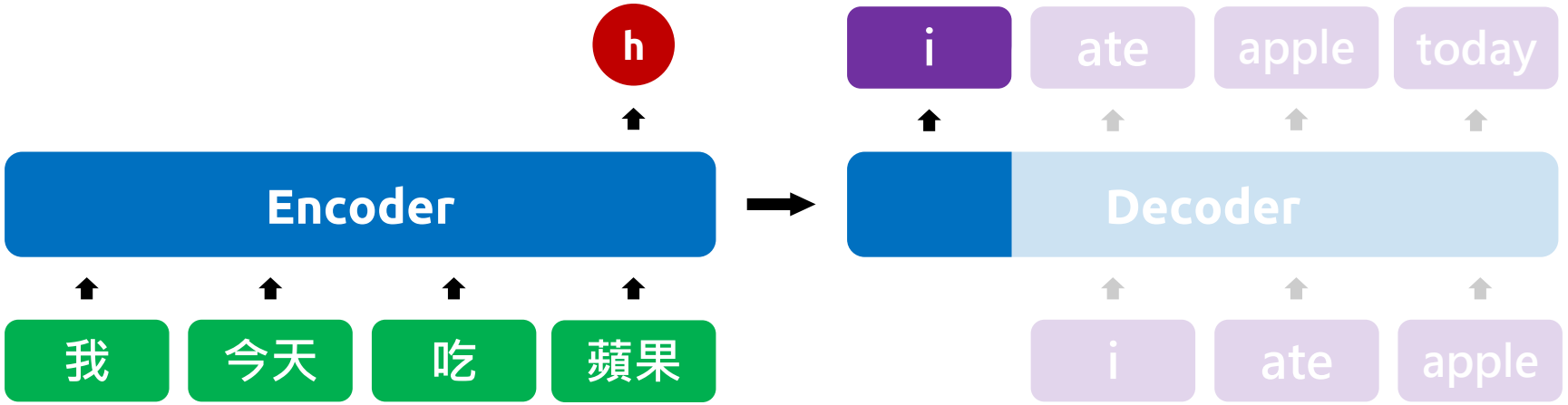


Sequence-to-Sequence (seq2seq)



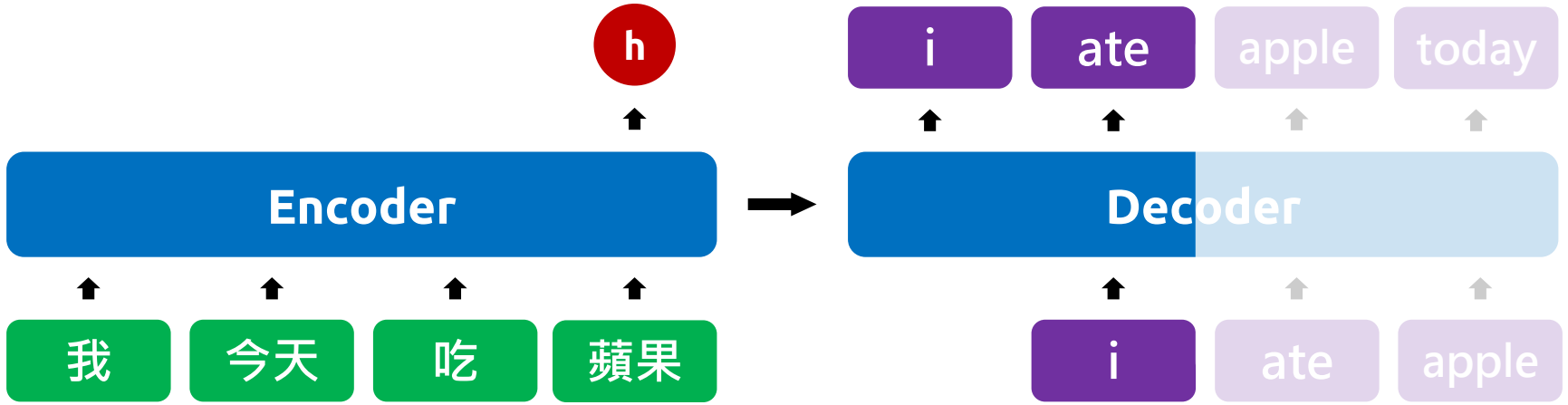
- **h** = Encoder(我 今天 吃 蘋果)

Sequence-to-Sequence (seq2seq)



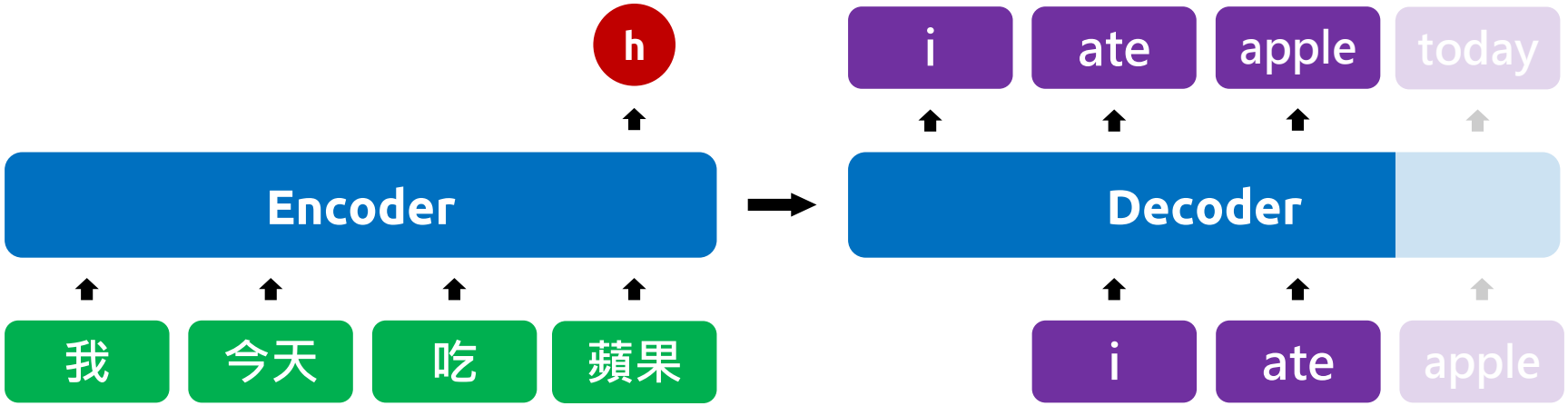
- **i** = Decoder(**| h**)

Sequence-to-Sequence (seq2seq)



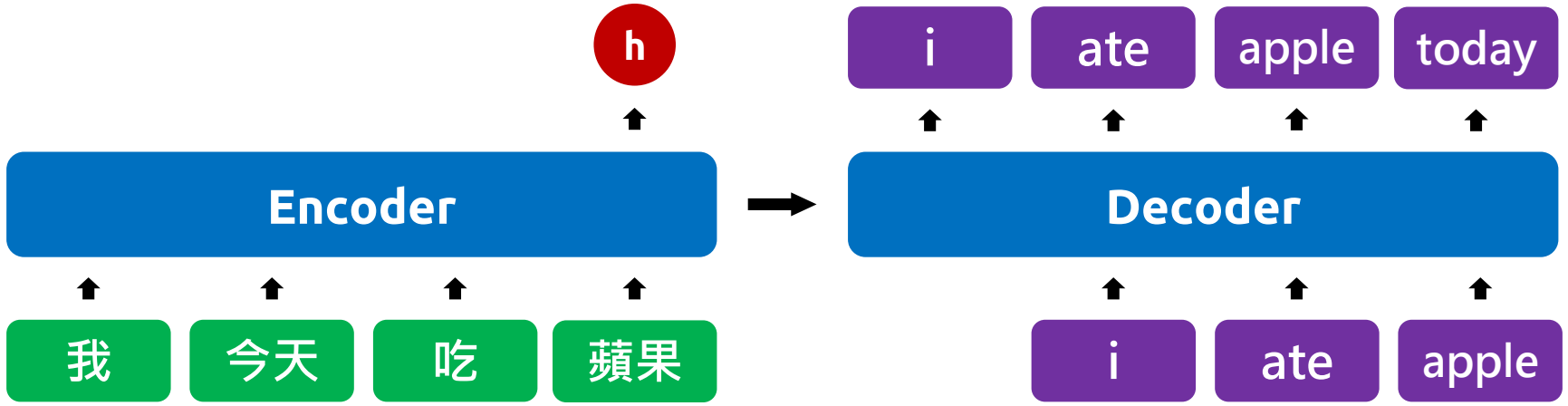
- **ate** = Decoder(i **| h**)

Sequence-to-Sequence (seq2seq)



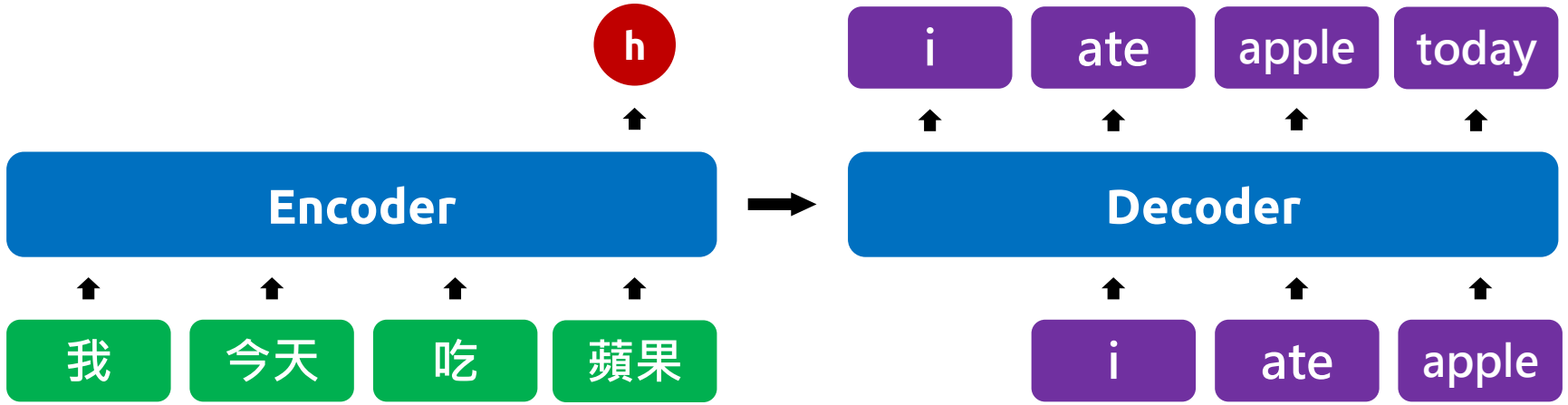
- **apple** = Decoder(i ate **| h**)

Sequence-to-Sequence (seq2seq)



- **today** = Decoder(i ate apple **| h**)

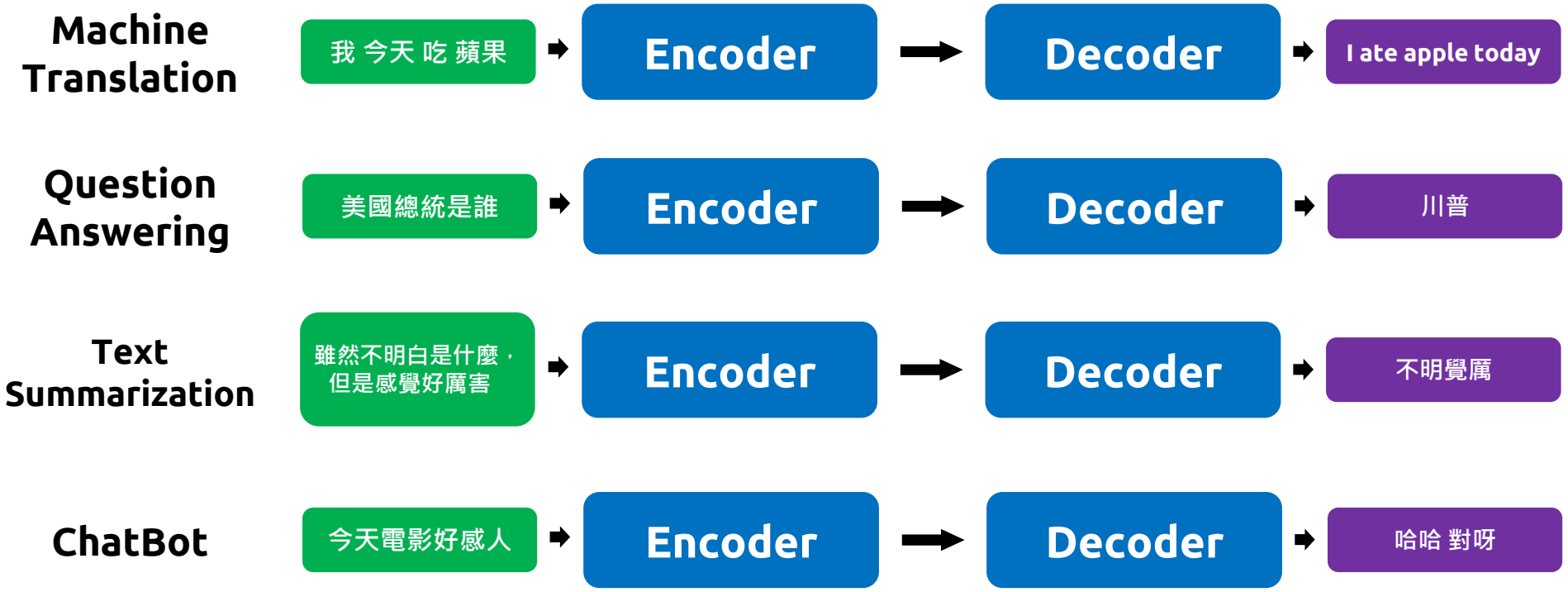
Sequence-to-Sequence (seq2seq)



- **h** = Encoder(我 今天 吃 蘋果)

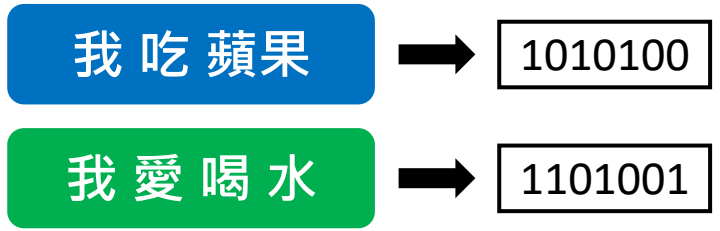
- **i** = Decoder(**| h**)
- **ate** = Decoder(i **| h**)
- **apple** = Decoder(i ate **| h**)
- **today** = Decoder(i ate apple **| h**)

Sequence-to-Sequence (seq2seq)



Review

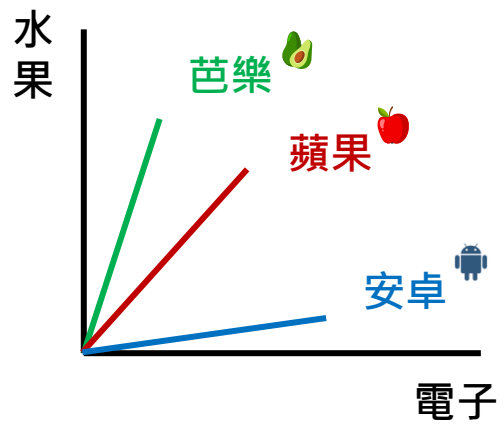
- Traditional
  - one-hot vector
    - inflexible (fixed by dictionary size)
    - lost “order”



Review

- Traditional
  - one-hot vector

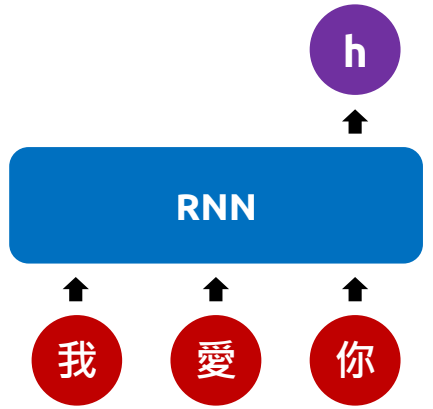
- Deep Learning
  - word vector (word2vec)
    - fixed dimension size
    - consider word relative



Review

- Traditional
  - one-hot vector

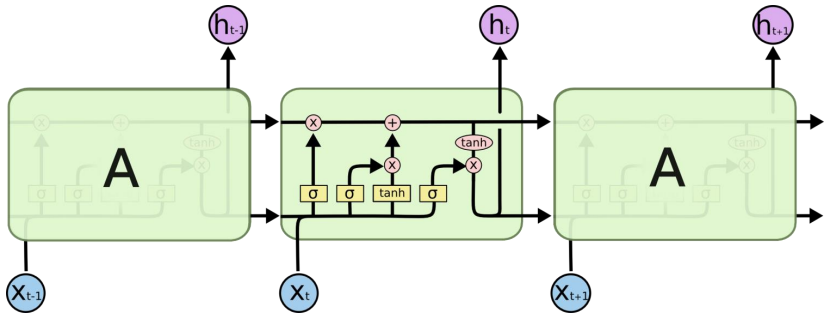
- Deep Learning
  - word vector (word2vec)
  - recurrent neural network (RNN)
    - consider “order” to model sequence
    - suffer from gradient vanishment



Review

- Traditional
  - one-hot vector

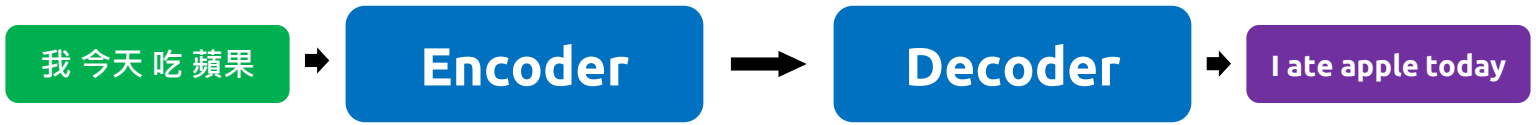
- Deep Learning
  - word vector (word2vec)
  - recurrent neural network (RNN)
  - long short-term memory (LSTM)
    - integrate “forgetness” into vanilla RNN



Review

- Traditional
  - one-hot vector

- Deep Learning
  - word vector (word2vec)
  - recurrent neural network (RNN)
  - long short-term memory (LSTM)
  - sequence-to-sequence (seq2seq)
    - 2 RNNs (encoder + decoder)
    - encodes input into h and decodes based on h



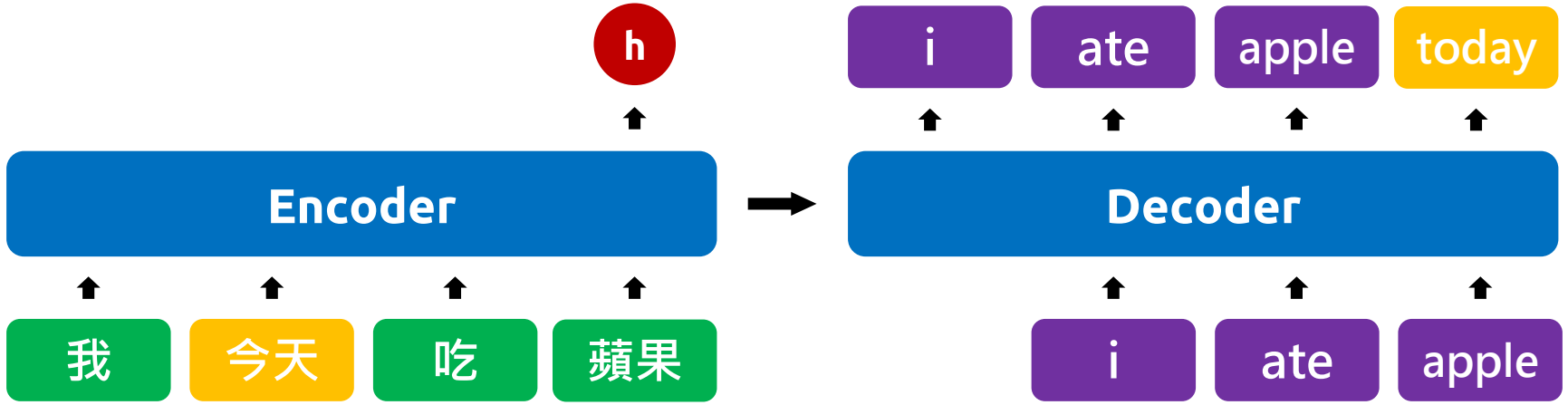
Review

- Traditional
  - one-hot vector

- Deep Learning
  - word vector (word2vec)
  - recurrent neural network (RNN)
  - long short-term memory (LSTM)
  - sequence-to-sequence (seq2seq)

More if you Like

- **attention** mechanism

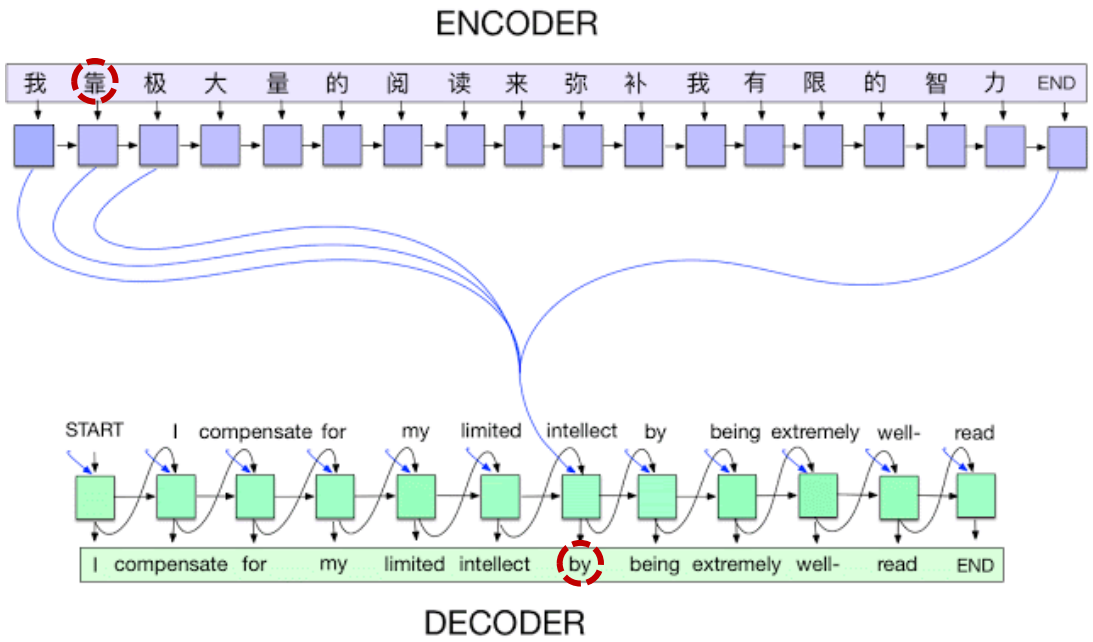


- during decoding, attends on different parts of input

https://towardsdatascience.com/day-1-2-attention-seq2seq-models-65df3f49e263

More if you Like

- **attention** mechanism

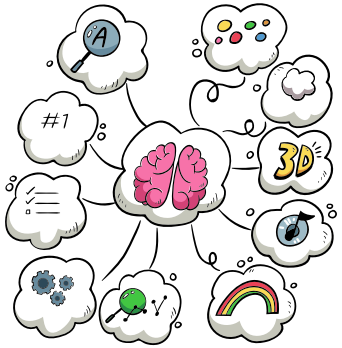


https://ai.googleblog.com/2017/04/introducing-tf-seq2seq-open-source.html

Outline

- Natural Language Processing (NLP)
- Deep Learning for Language
- Research / Application on NLP

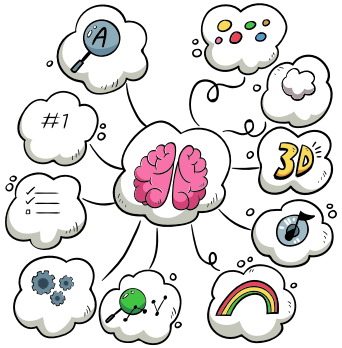
Relation Extraction



- Extract the **relation** between entities

Sentence	Relation
Jobs founded **Apple**	**Founder**
Jobs ate **apple**	**NA**
Gates built **Microsoft**	**Founder**

Relation Extraction



- Extract the **relation** between entities

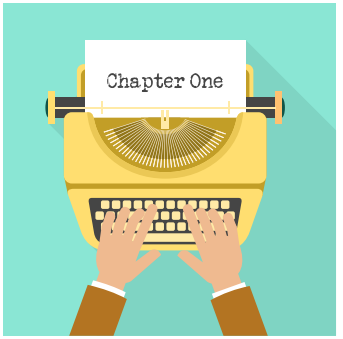
Sentence	Relation
Jobs founded **Apple**	**Founder**
Jobs ate **apple**	**NA**
Gates built **Microsoft**	**Founder**

> 張學友（英語：Jacky Cheung Hok-yau，1961年7月10日 - ），香港歌手、演員和音樂人，享譽國際，同時也是大中華區以至亞洲巨星，在華語地區享有「歌神」美譽。

➡

姓名	張學友
英文名	Jacky
暱稱	歌神
國籍	香港
出生	1961年7月10日
職業	歌手、演員

Data-to-Text Generation



Medal Table from Tournament

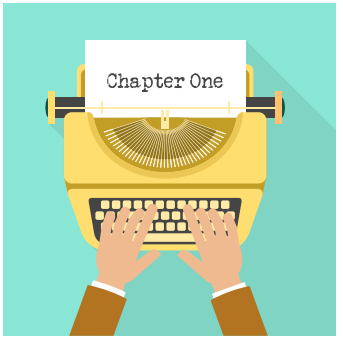
Nation	Gold Medal	Silver Medal	Bronze Medal	Sports
Canada	3	1	2	Ice Hockey
Mexico	2	3	1	Baseball
Colombia	1	3	0	Roller Skating

Surface-level Generation
Sentence: Canada has got 3 gold medals in the tournament. **Sentence:** Mexico got 3 silver medals and 1 bronze medal.

Logical Natural Language Generation
Sentence: Canada obtained 1 more gold medal than Mexico. **Sentence**: Canada obtained the most gold medals in the game.

https://arxiv.org/pdf/2004.10404.pdf

Data-to-Text Generation

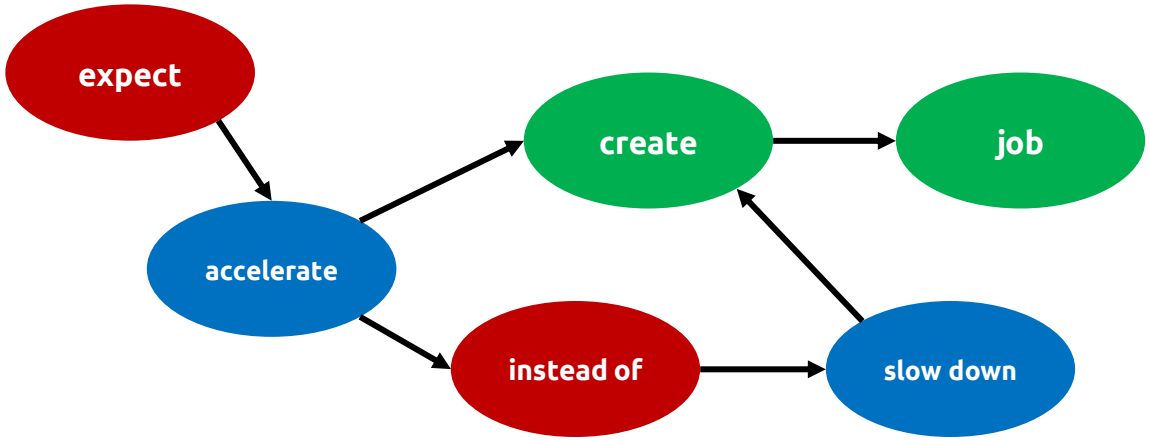


Medal Table from Tournament

Nation	Gold Medal	Silver Medal	Bronze Medal	Sports
Canada	3	1	2	Ice Hockey
Mexico	2	3	1	Baseball
Colombia	1	3	0	Roller Skating

Surface-level Generation
Sentence: Canada has got 3 gold medals in the tournament. **Sentence:** Mexico got 3 silver medals and 1 bronze medal.

Logical Natural Language Generation
Sentence: Canada obtained 1 more gold medal than Mexico. **Sentence**: Canada obtained the most gold medals in the game.



Job creation is not **expected** to **slow down** but will **instead accelerate**.

https://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00297

Visual-Story Telling

- **Describe** images by language

→

The **sun** is setting over the **ocean** and **mountains**.

Visual-Story Telling

- **Describe** images by language

The **sun** is setting over the **ocean** and **mountains**.

We went on a hike yesterday.

There were a lot of strange plants there.

I had a great time.

We drank a lot of water while we were hiking.

The view was amazing.

https://arxiv.org/pdf/1804.09160.pdf

Instruction-Manipulated Robotics

- **Manipulate** robots by language

Can you move the **tissue box** to the **left**?

https://arxiv.org/pdf/1710.06280.pdf

Instruction-Manipulated Robotics

- **Manipulate** robots by language

Can you move the **tissue box** to the **left**?

Can you move the **brown fluffy thing** to the **bottom**?

https://arxiv.org/pdf/1710.06280.pdf

Vision-and-Language Navigation

- **Reach the goal** based on the instruction in an environment

Leave the bedroom, and **enter the kitchen**. Walk forward and take a **left at the couch**. Stop **in front of the window**.

Exit the bedroom. Turn **left and exit** the room using the **door on the left**. **Wait** there.

https://arxiv.org/pdf/1711.07280.pdf

Instructed Image Editing

- **Edit image** based on instruction

https://arxiv.org/pdf/1906.07689.pdf

Instructed Image Editing

- **Edit image** based on instruction

Remove the people from the picture.

Change sky color to red

Model

https://arxiv.org/pdf/1906.07689.pdf

Gender Bias

- Caused by training on **biased** data

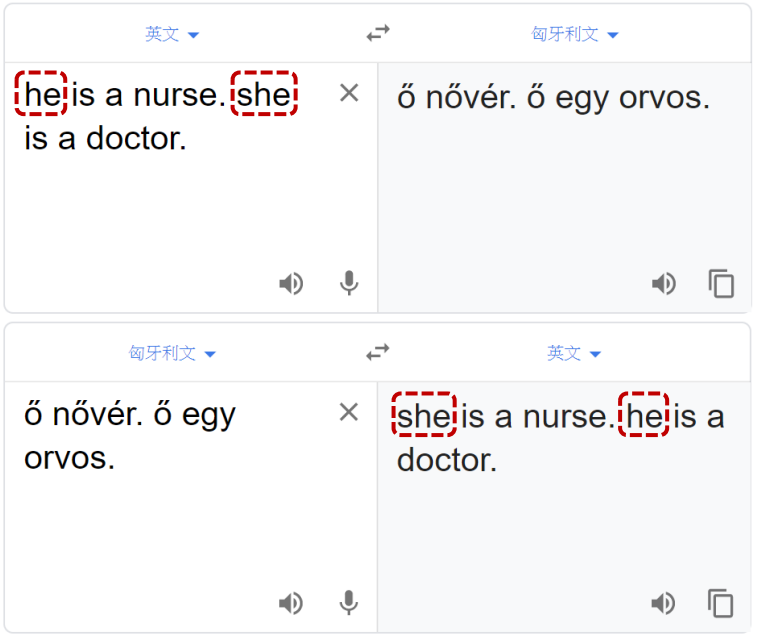
A **man** sitting at a desk
with a **laptop computer**

https://arxiv.org/pdf/1803.09797.pdf

Gender Bias

- Caused by training on **biased** data

A **man** sitting at a desk with a **laptop computer**



https://arxiv.org/pdf/1803.09797.pdf


If Someone in Your Family Has Cancer
Definition
Feelings
Treatment
Who?
What?
When?
Where?
Why?
How?
EUROPE
Q&A


Welcome to
UCSB