Natural Language Processing

Fundamentals of Artificial Intelligence

Instructor: Chenhui Chu

Email: chu@i.kyoto-u.ac.jp

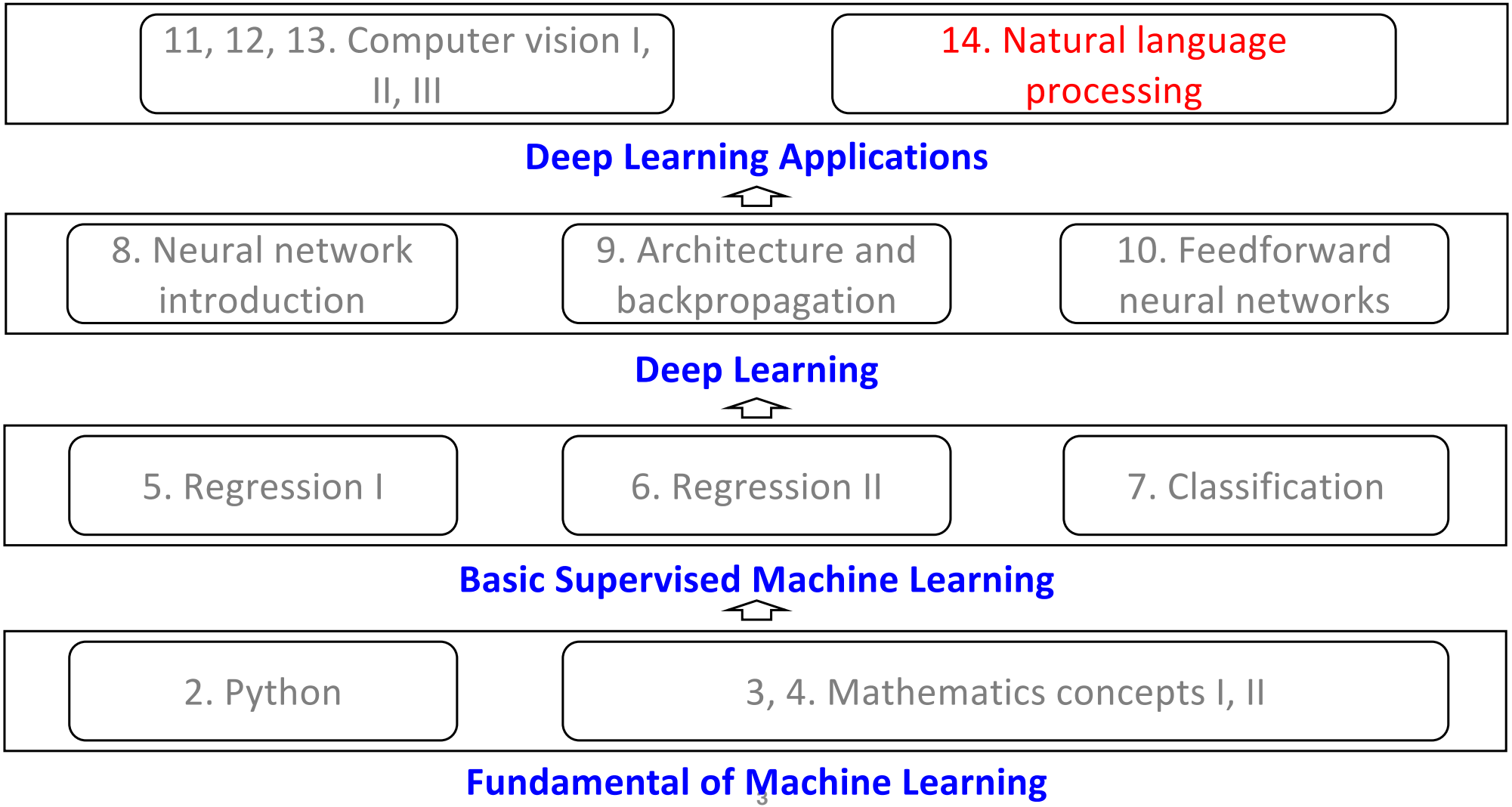
Teaching Assistant: Youyuan Lin

E-mail: youyuan@nlp.ist.i.kyoto-u.ac.jp

Schedule

- 1. Overview of AI and this Course (4/14)
- 2. Introduction to Python (4/21)
- 3, 4. Mathematics Concepts I, II (4/28, 5/12)
- 5, 6. Regression I, II (5/19, 5/26)
- 7. Classification (6/2)
- 8. Introduction to Neural Networks (6/9)
- 9. Neural Networks Architecture and Backpropagation (6/16)
- 10. Fully Connected Layers (6/23)
- 11, 12, 13. Computer Vision I, II, III (6/30, 7/7, 7/14)
- 14. Natural Language Processing (7/17)

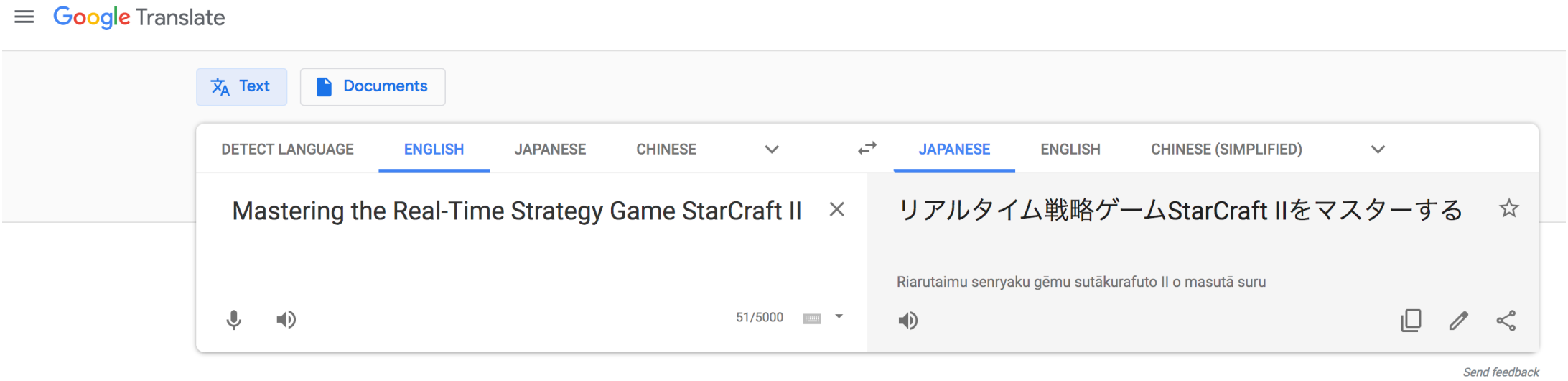
Overview of This Course



Natural Language Processing

- **Natural Language Processing** is a subfield of Computer Science
- It is about processing texts in **Human Languages** (English, Japanese, etc.)
- It covers many tasks:
  - Machine Translation

Machine Translation

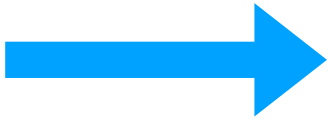


Natural Language Processing

- Natural Language Processing is a subfield of Computer Science
- It is about processing texts in Human Languages (English, Japanese, etc.)
- It covers many tasks:
  - Machine Translation
  - Automatic summarization

Automatic Summarization

Russian defense minister Ivanov called Sunday for the creation of a joint front for combating global terrorism



Russia calls for a joint front against terrorism

Natural Language Processing

- Natural Language Processing is a subfield of Computer Science
- It is about processing texts in Human Languages (English, Japanese, etc.)
- It covers many tasks:
  - Machine Translation
  - Automatic summarization
  - Human-Machine Dialog (chatbots, etc.)

Question Answering and Dialog

Computer, who was the Prime Minister of Japan in 2002?

It was Jun'Ichiro Koizumi

How long was he Prime Minister?

About 5 years

Is he still alive?

Yes

Natural Language Processing

- Natural Language Processing is a subfield of Computer Science
- It is about processing texts in Human Languages (English, Japanese, etc.)
- It covers many tasks:
  - Machine Translation
  - Automatic summarization
  - Human-Machine Dialog (chatbots, etc.)
  - Text understanding (Sentiment Analysis, etc.)

Sentiment Analysis

Imdb comment

Since Disney is incompetent of coming up with new ideas, and must resort to using older stories they did years ago, they certainly better live up to what they're doing. This does not happen with Mary Poppins Returns.

This user thinks the movie is bad or good?

Bad

amazon.com comment

I just got these today and they were exactly what I was looking for. They are lightweight without being too thin. Almost my entire house is hardwood or tile, and I use these on my swiffer mops in place of the expensive wet cloths

This user liked the product or not?

Liked

Natural Language Processing

- Natural Language Processing is a subfield of Computer Science
- It is about processing texts in Human Languages (English, Japanese, etc.)
- It covers many tasks:
  - Machine Translation
  - Automatic summarization
  - Human-Machine Dialog (chatbots, etc.)
  - Text understanding (Sentiment Analysis, etc.)

Natural Language Processing

- **Natural Language Processing** is a subfield of Computer Science
- It is about processing texts in **Human Languages** (English, Japanese, etc.)
- Like Computer Vision, it is now dominated by approaches based on **Neural Networks**
  - Slightly more recent trend:
    - Neural Networks beat other methods for Image recognition since 2012
    - Neural Networks beat other methods for Machine Translation since 2015~2016

Sequence Processing

- A **text** is a **sequence of words**
- The Neural Network architectures we are going to see can also be applied to **any sequence of symbols**
- For example:
  - Music sheets (sequences of notes)
  - DNA sequences (sequence of genes)
  - Etc.

Difficulties in Text Process (1/2)

- Processing text with Neural Networks is actually more complex than processing images
- Mainly for 2 reasons:

Difficulties in Text Process (2/2)

- Processing text with Neural Networks is actually more complex than processing images
- Mainly for 2 reasons:
  - Text is not “naturally” represented by numbers
  - Text can be of unlimited length
    - And it cannot be “naturally” reduced to a fixed size

How to Present Text as Numbers

- How to see this sentence as a set of numbers?

Let us go to the beach!

- First, we look at it as a sequence of symbols. This is called “Tokenization”

Let | us | go | to | the | beach | !

Tokenization

For a language like English, tokenization is easy: we just “cut” along the spaces:

Let us go to the beach!

Let | us | go | to | the | beach | !

For a language like Japanese, not so simple. Two options:

ビーチへ行こう！

Use a specialized software (called “segmenter”) that can recognize limits of Japanese words:

ビーチ | へ | 行こう | ！

Cut out each character:

ビ | ー | チ | へ | 行 | こ | う | ！

How to Present Text as Numbers

- How to see this sentence as a set of numbers?

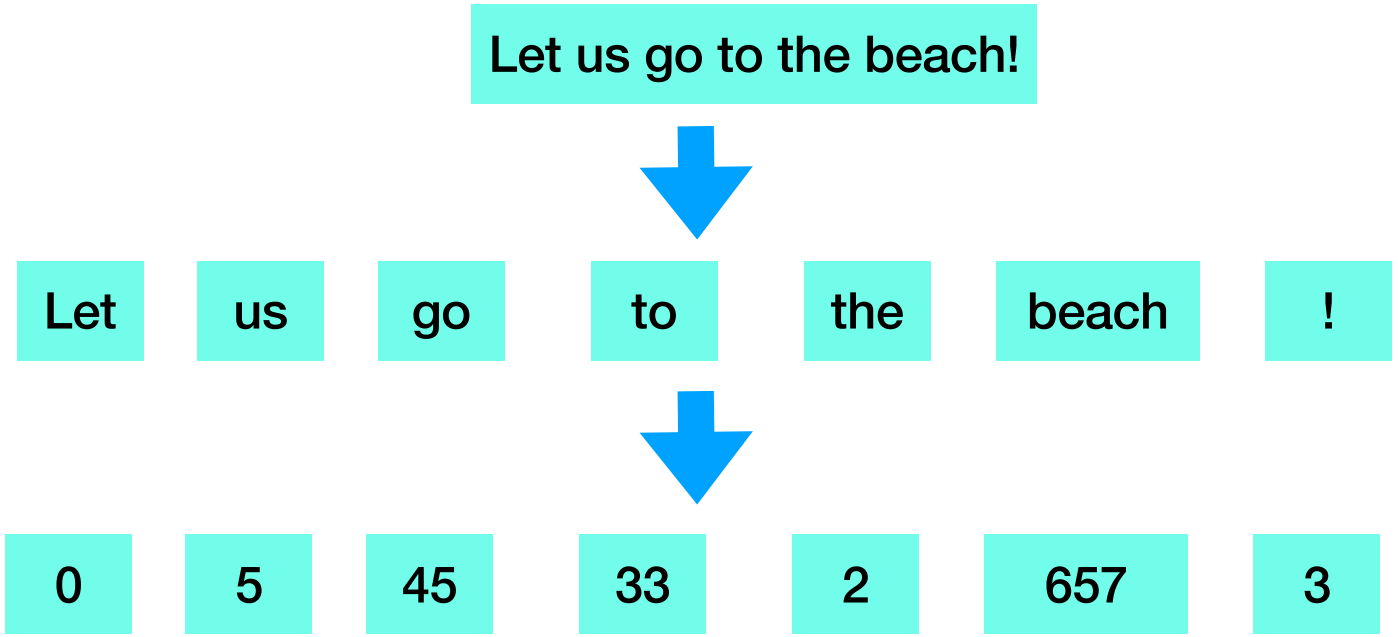
Let us go to the beach!

- First, we look at it as a sequence of symbols. This is called “Tokenization”

Let | us | go | to | the | beach | !

A Number for Each Word? (1/2)

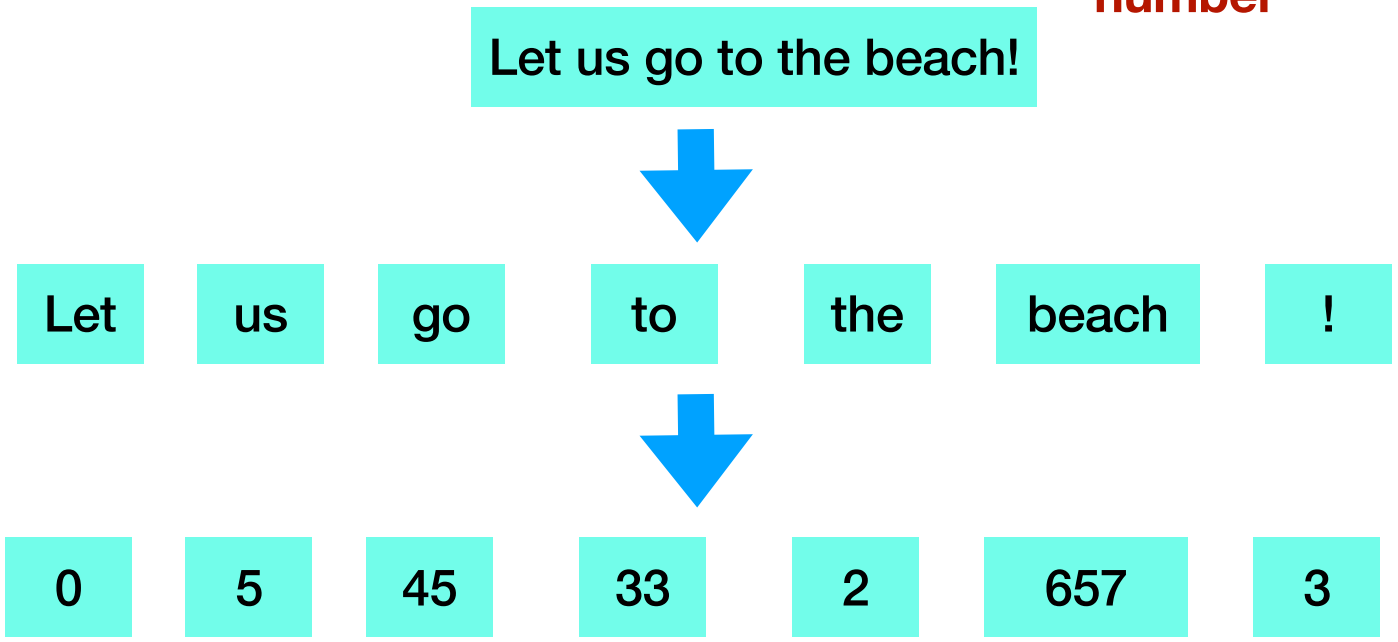
- Then maybe we can just assign a number to each word?
  - “Let” = 0
  - “Us” = 5
  - “go” = 45
  - …



A Number for Each Word? (2/2)

- Then maybe we can just assign a number to each word?
  - “Let” = 0
  - “Us” = 5
  - “go” = 45
  - …

Will not work!
The relations between words cannot be represented by a simple number



Words are not Independent! (1/2)

- Intuitively, we have some notion of **distance** between words

Close in meaning:

Big

Large

Gigantic

Not so close in meaning:

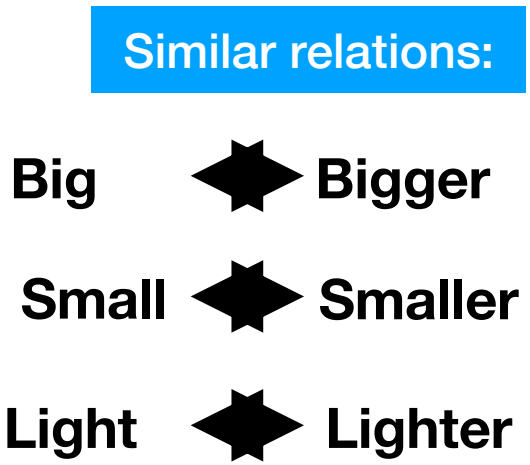
Big

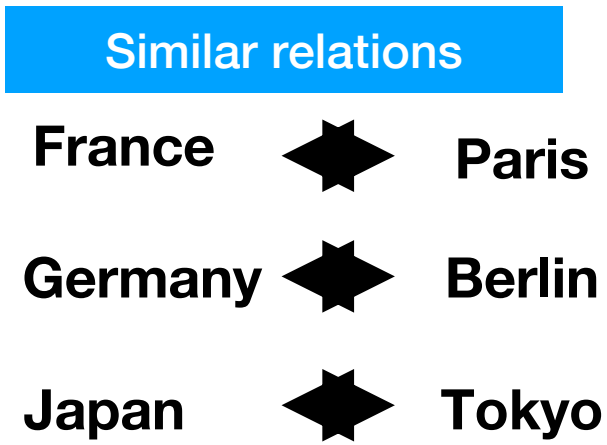
Green

Lawyer

Words are not Independent! (2/2)

- Intuitively, we have some notion of **relation** between words





More Than One Number?

- Words are more **expressive** than single numbers
- We will not get good results by using a single number for each word
- What about using more than one number?

Vectors

Addition:

0.2 3.6 2.1 5.3 -2.2 + 1.1 -2.0 1.1 -5.3 -1.0 = 1.3 1.6 3.2 0.0 -3.2

Norm:

$\| 1 \quad -1 \quad 2 \quad 5 \quad -2 \| = \sqrt{1^2 + -1^2 + 2^2 + 5^2 + (-2)^2} = \sqrt{32} \approx 5.65$

Note: Norm is always positive

Distance: $d(\vec{x}, \vec{y}) = ||\vec{x} - \vec{y}||$

$d($ 1 1 2 5 -2 , 1 0 2 4 2 $) = \sqrt{(1-1)^2 + (0-1)^2 + (2-2)^2 + (5-4)^2 + (-2-2)^2} \approx 4.24$

More Than One Number?

- Words are more expressive than single numbers
- We will not get good results by using a single number for each word
- What about using more than one number?
- We have a **notion of distance on vectors**

Word Embeddings (1/2)

- A word embedding associates a **fixed-size vector** to *each* **word** in a **vocabulary**

100 000 words in our Vocabulary		***100 000 vectors of dimension 200***
Madrid		$[2,4,1.2,\ldots,-1.3,4]$
Big		$[5,3,1.2,\ldots,5,-4]$
Japan	Embedding	$[1,2.1,-1,\ldots,2,2.1]$
Large		$[6,2.9,-2.2,\ldots,2,2.7]$
Small		$[-2.2,2.3,-1,\ldots,2,5.1]$
Spain		$[1.5,41,-1,\ldots,2,2.3]$

Word Embeddings (2/2)

- ***Word embeddings*** are vectors associated to each word in a vocabulary
- The vectors are expected to be “meaningful”:
  - Vectors for words “big” and “large” should be close together

$$d(\vec{big}, \vec{large}) < d(\vec{big}, \vec{lawyer})$$

  - Relation between vectors should reflect semantic relations between words:

$$\vec{King} - \vec{Man} + \vec{Woman} \approx \vec{Queen}$$

$$\vec{Paris} - \vec{France} + \vec{Japan} \approx \vec{Tokyo}$$

How to Obtain Meaningful Word Embeddings?

- We can obtain these Word Embeddings by training a Simple Neural Network on text data
- In practice, we will typically use vectors of dimension 200 to 400

Skip-Gram and CBOW (word2vec)

- (Mikolov+, 2013) proposed two very efficient simple models for learning word embeddings
- These models are sometimes referred to as ***word2vec*** models (word2vec being the name of the software that was released for producing these vectors)
- In the paper, the models are actually called ***CBOW*** and ***Skip-Gram***
- Let us see how “Skip-Gram” works (we will see a slightly modified version)

Context Word

- For each word w in a corpus C, we consider its ***context*** as the ***c*** words that come before and the c words that come after

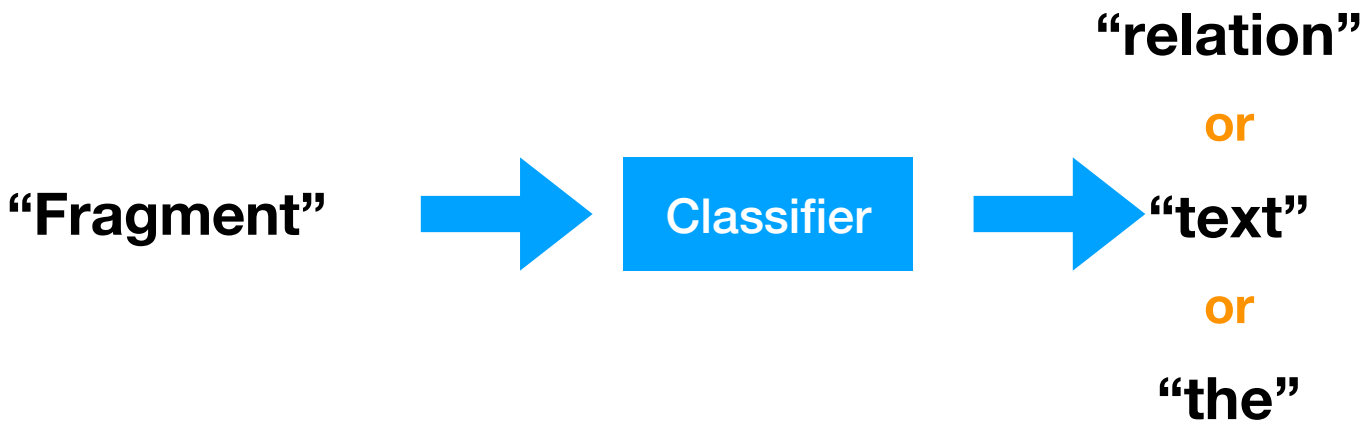
Textual entailment (TE) in natural language processing is a directional relation between text fragments. The relation holds whenever the truth of one text fragment follows from another text. In the TE framework, the entailing and entailed texts are termed text (t) and hypothesis (h), respectively. Textual entailment is not the same as pure logical entailment — it has a more relaxed definition

  - *Context* of this instance of fragments (if *c* = 3):

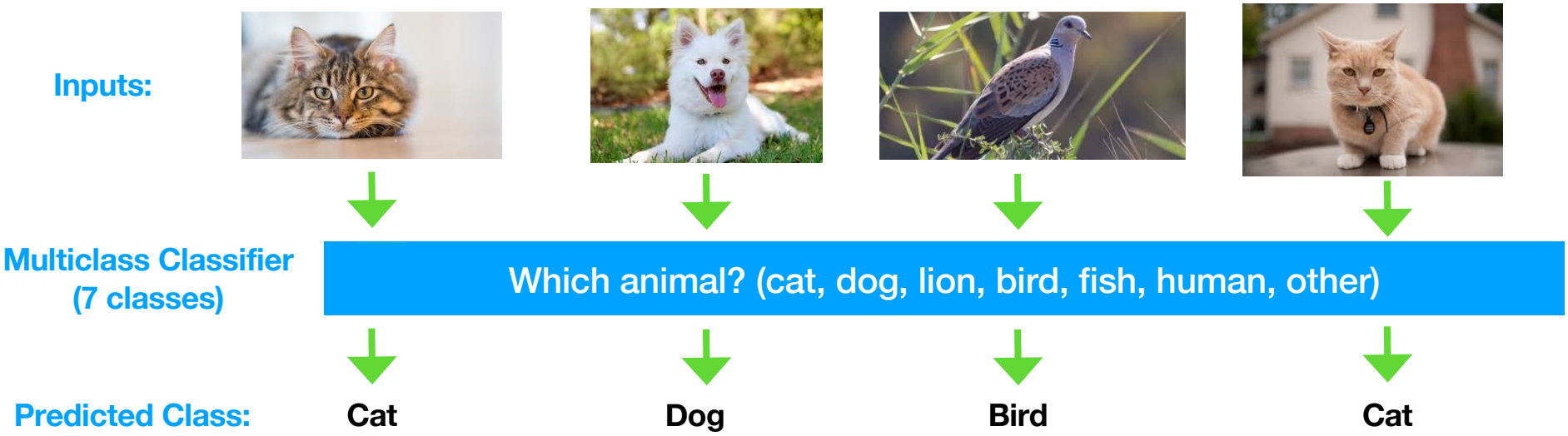
    - relation, between, text, the, relation, holds

Context Word Prediction

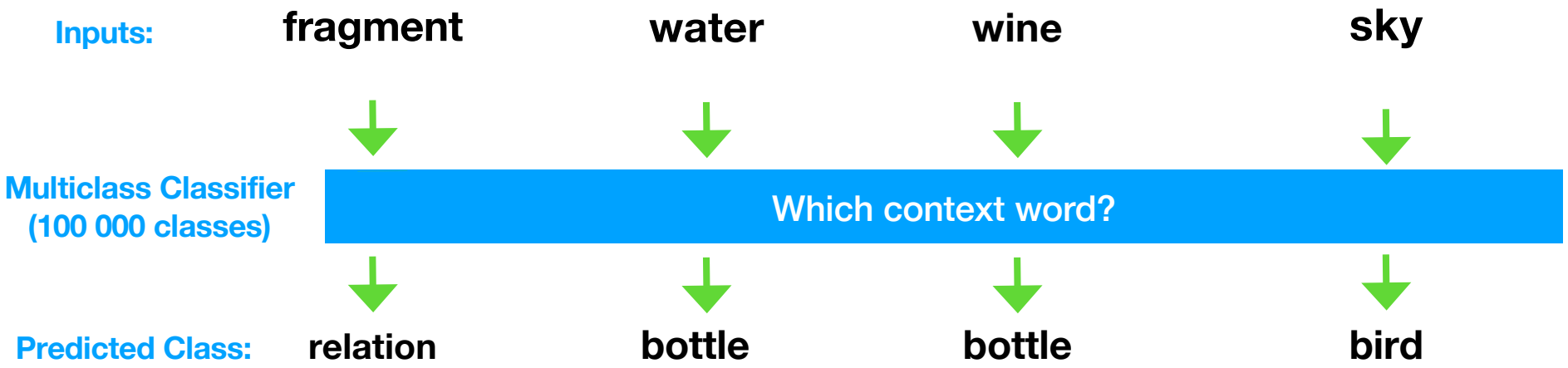
- We are going to train a simple Neural Network classifier that tries to predict context words given an input word



Multiclass Classification for Images

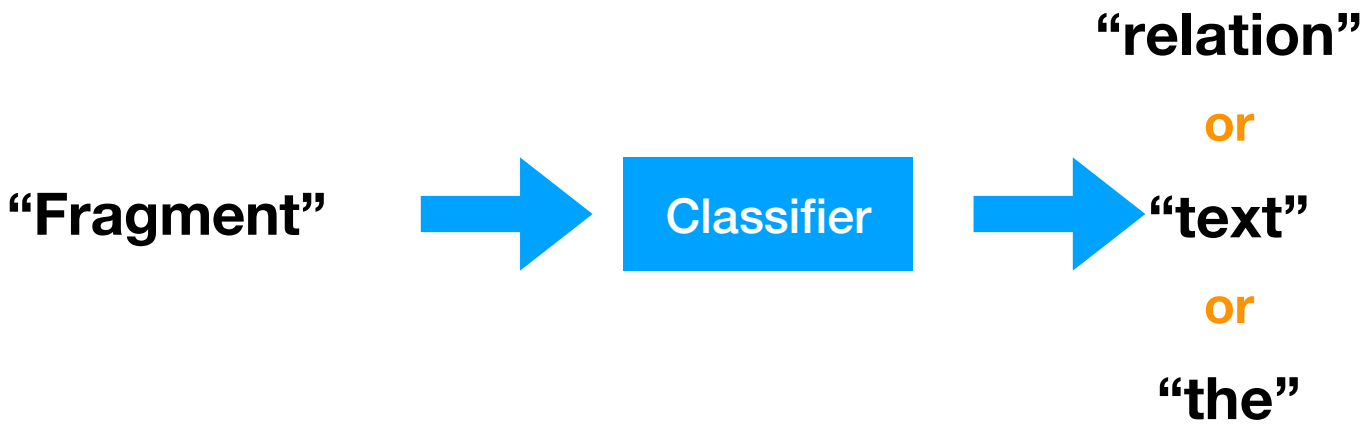


Multiclass Classification for Text



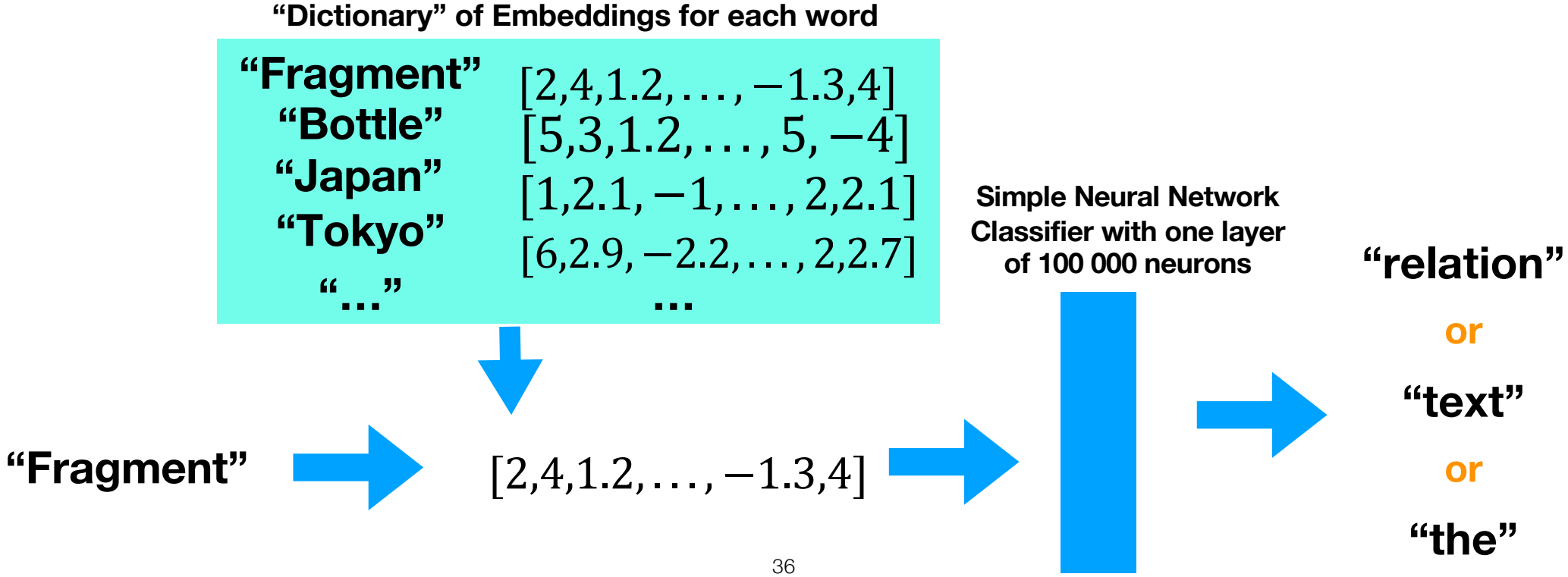
Context Word Prediction (1/3)

- We are going to train a simple Neural Network classifier that tries to predict context words given an input word



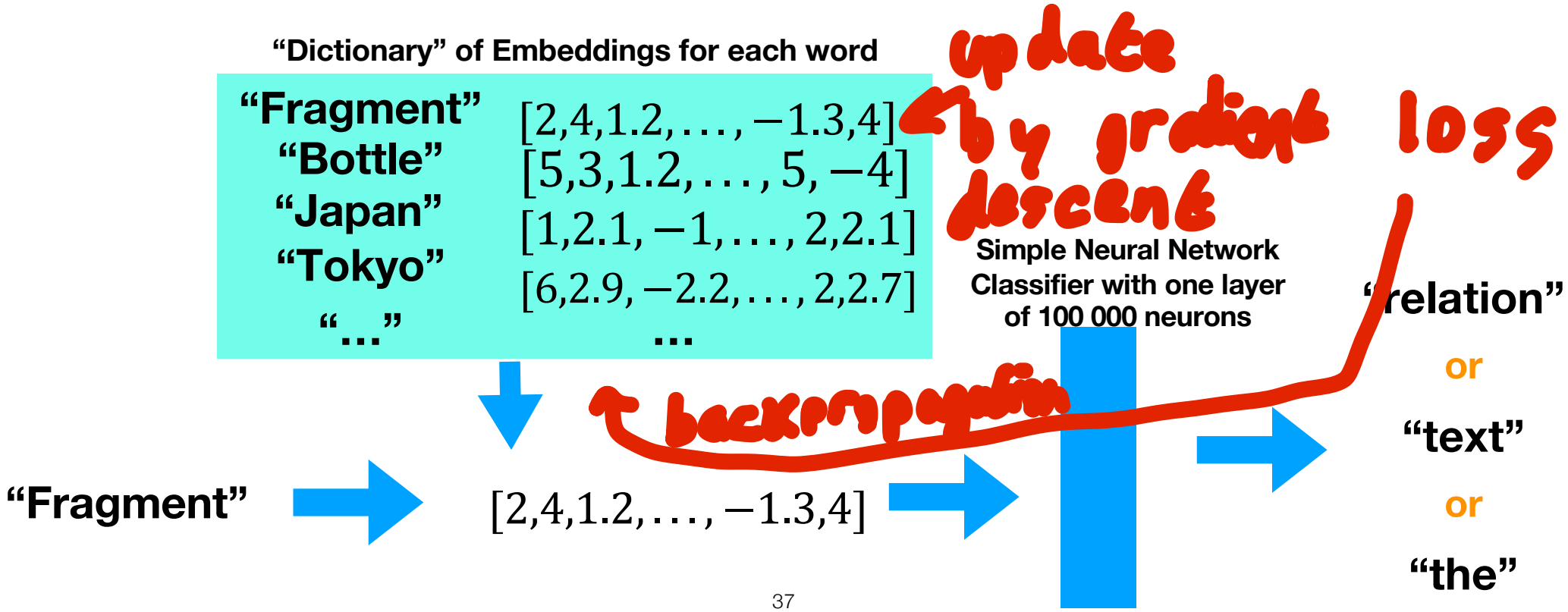
Context Word Prediction (2/3)

- We are going to train a simple Neural Network classifier that tries to predict context words given an input word



Context Word Prediction (3/3)

- During training, we use back propagation and gradient descent to update the embeddings in the dictionnary



Training Data

- Training data is easy to obtain. Just download Wikipedia, for example. Then extracts all pairs of (input word, context word)

Textual entailment (TE) in natural language processing is a directional relation between text fragments. The relation holds whenever the truth of one text fragment follows from another text. In the TE framework, the entailing and entailed texts are termed text (t) and hypothesis (h), respectively. Textual entailment is not the same as pure logical entailment — it has a more relaxed definition

- *Context* of this instance of fragments (if $c = 3$):
  - relation, between, text, the, relation, holds

Skip-gram Embeddings (1/2)

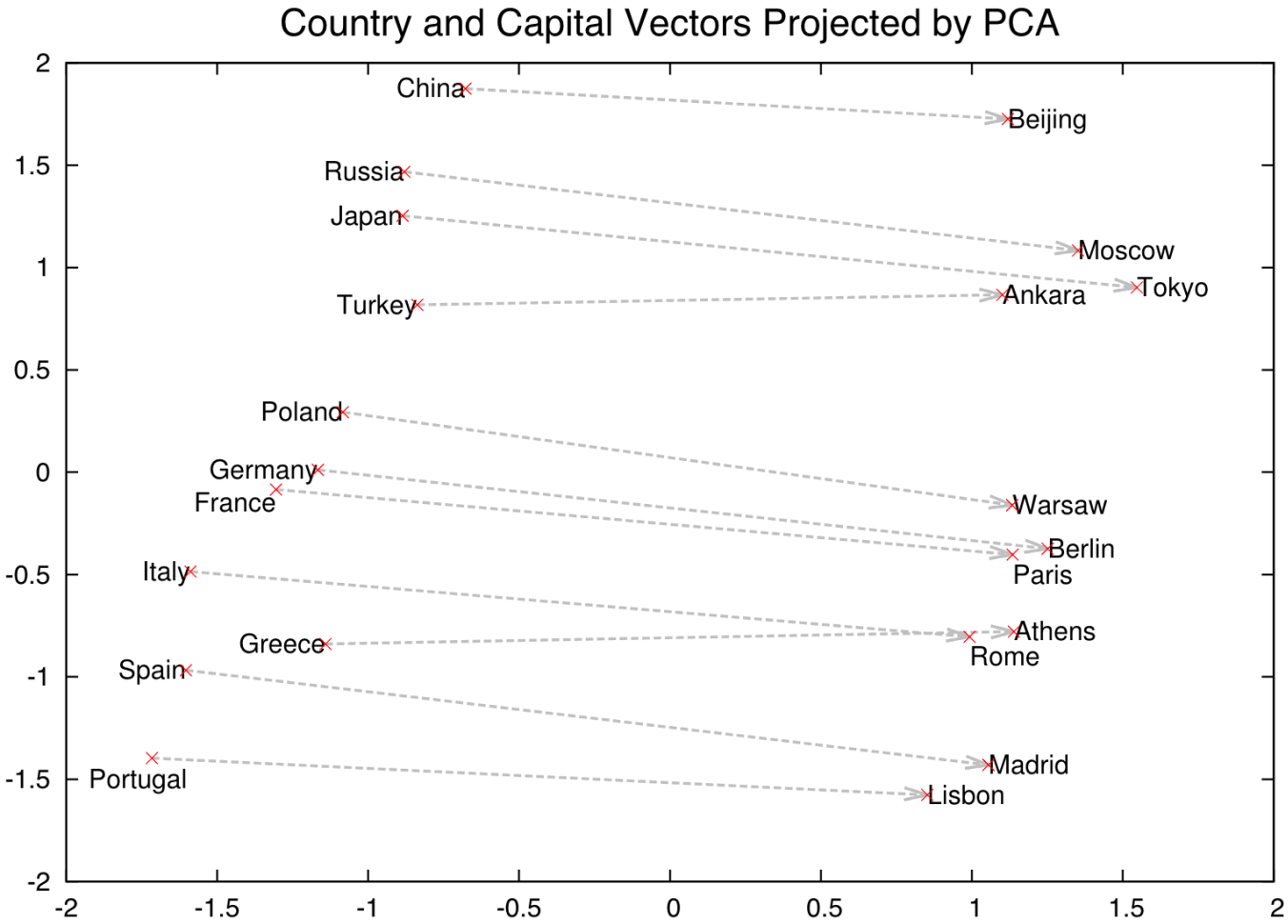
- The embeddings obtained in this way work very well to express relationship between words

$$d(\vec{big}, \vec{large}) < d(\vec{big}, \vec{lawyer})$$

$$\vec{King} - \vec{Man} + \vec{Woman} \approx \vec{Queen}$$

$$\vec{Paris} - \vec{France} + \vec{Japan} \approx \vec{Tokyo}$$

Skip-gram Embeddings (2/2)

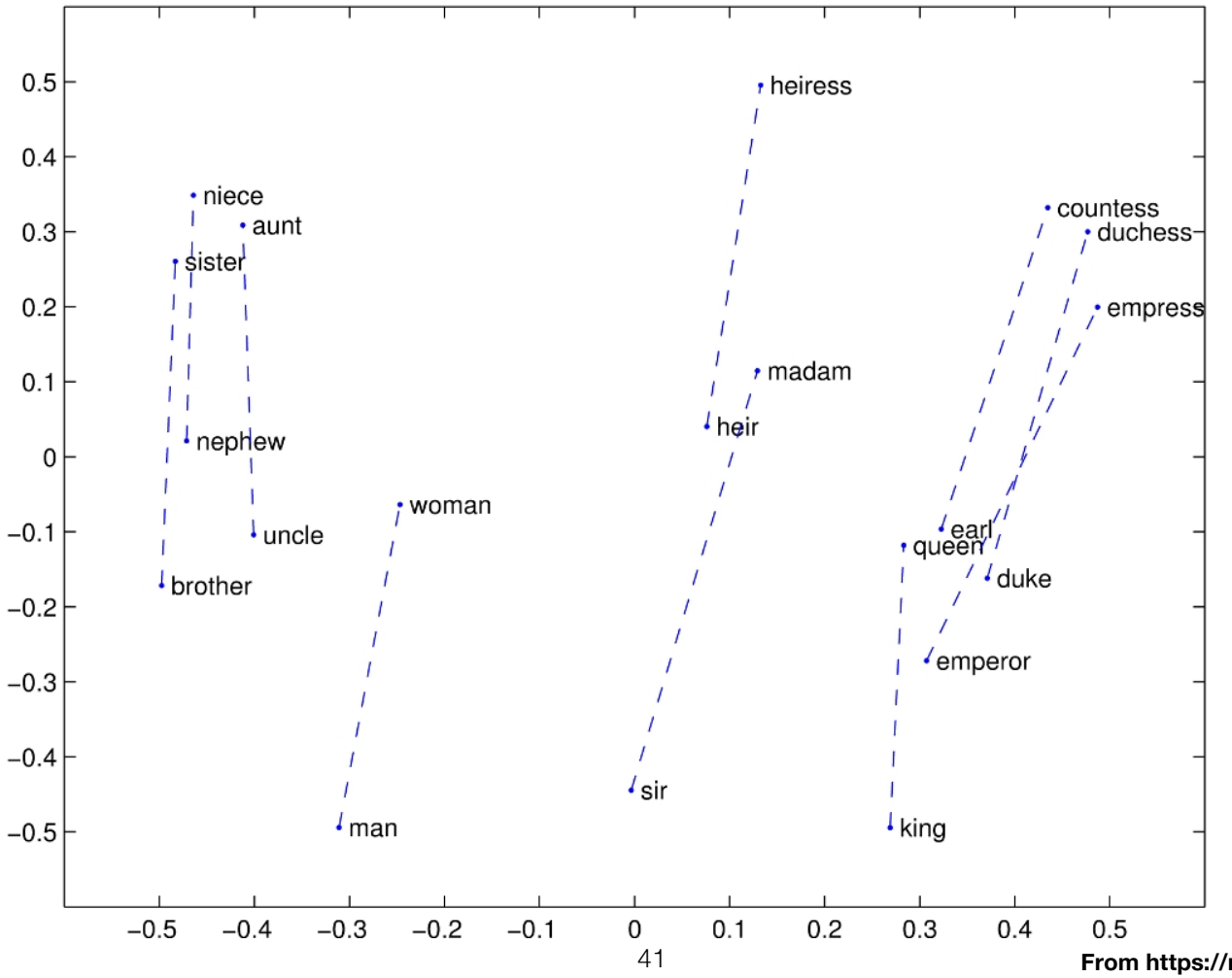


From (Mikolov+, 2013)

- If we map the embeddings into 2 dimension, we can visualize the relationships

- PCA = Principal Component Analysis
  - Reduces dimension of vectors to 2

Other Examples (1/2)

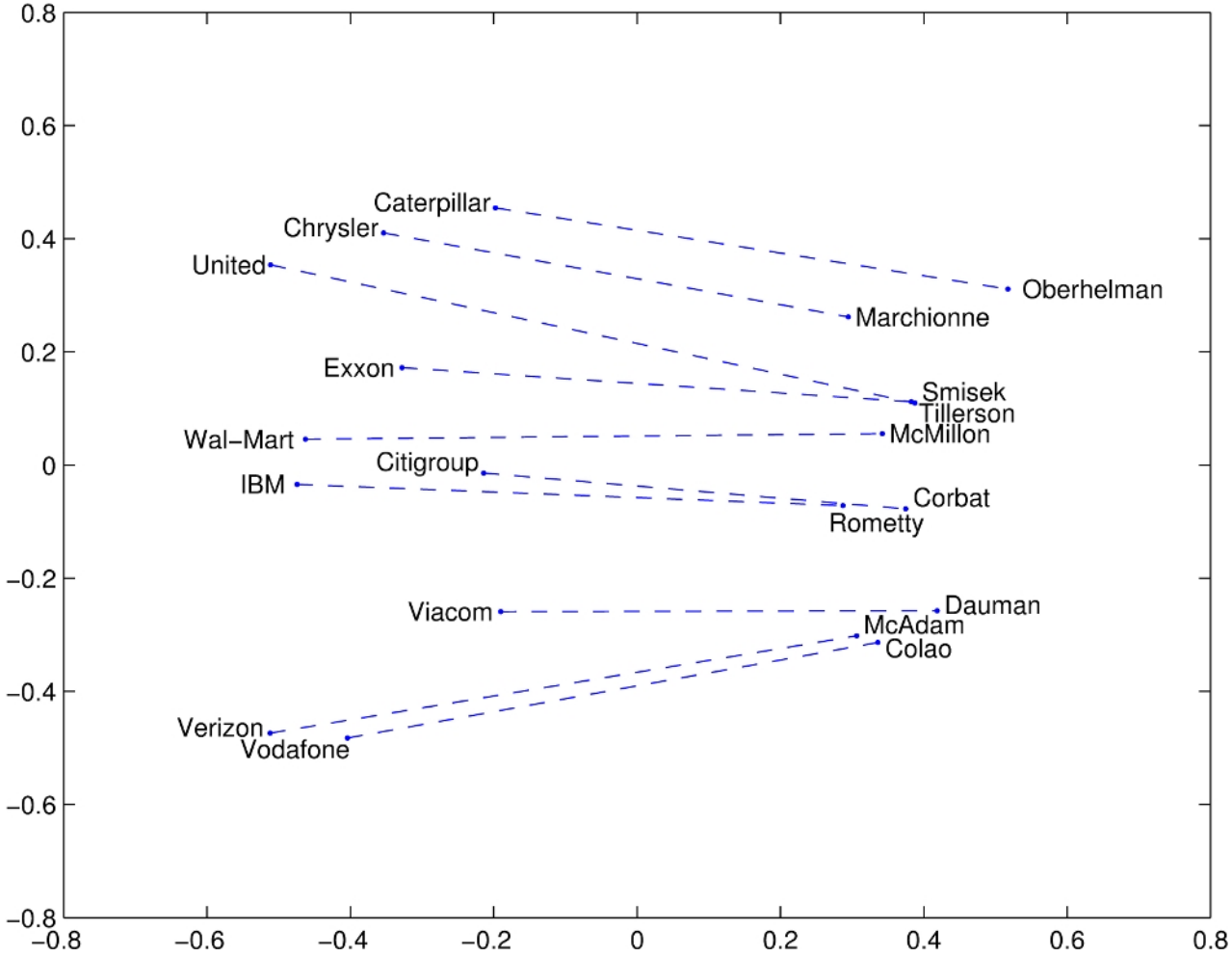


Men-Women

(Not Skip-Gram, but similar model)

From https://nlp.stanford.edu/projects/glove/

Other Examples (2/2)



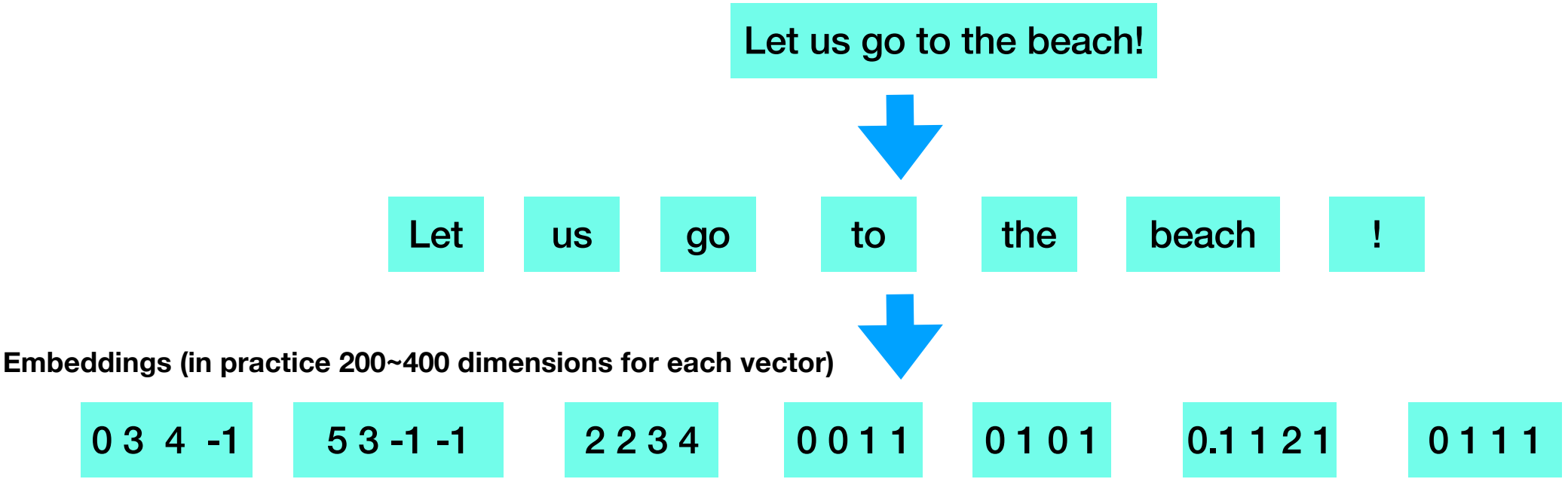
Company-CEO

(Not Skip-Gram, but similar model)

From https://nlp.stanford.edu/projects/glove/

Text as Numbers

- We solved our first issue: now we know how to see words as numbers



Difficulties in Text Process (1/2)

- Processing text with Neural Networks is actually more complex than processing images
- Mainly for 2 reasons:
  - Text is not “naturally” represented by numbers
  - Text can be of unlimited length
    - And it cannot be “naturally” reduced to a fixed size

Difficulties in Text Process (2/2)

- Processing text with Neural Networks is actually more complex than processing images
- Mainly for 2 reasons:
  - Text is not “naturally” represented by numbers
  - Text can be of unlimited length
    - And it cannot be “naturally” reduced to a fixed size

Actually, Word Embeddings works well for that

Can We Resize Text? (1/4)

- We can now transform our text into a sequence of vectors.
- But this sequence could have any length
- For images, having input images of different sizes is not really a big problem, because it is easy to “resize” the image

Can We Resize Text? (2/4)

- We can now transform our text into a sequence of vectors.
- But this sequence could have any length
- For images, having input images of different sizes is not really a big problem, because we it is easy to “resize” the image

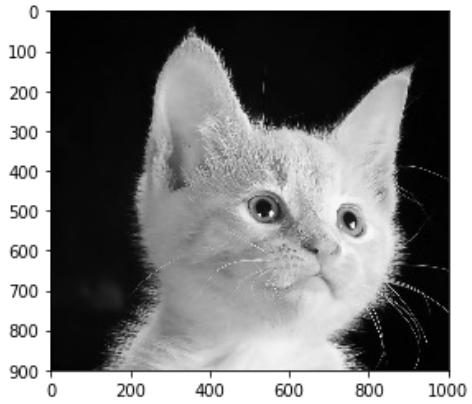
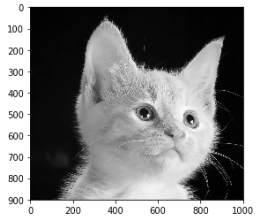


Image of a cat

For each 4 pixels, only keep one



Smaller Image of a cat

Can We Resize Text? (3/4)

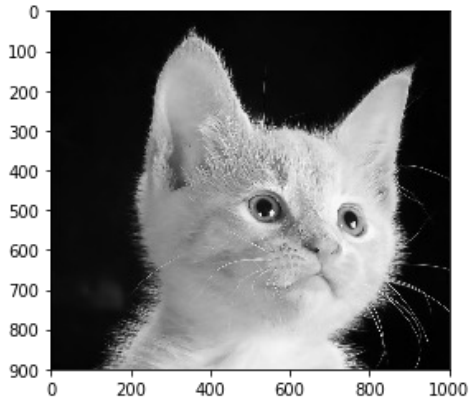
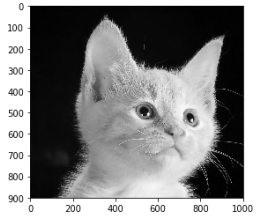


Image of a cat

For each 4 pixels, only keep one



Smaller Image of a cat

Meaningful text

Since Disney is incompetent of coming up with new ideas, and must resort to using older stories they did years ago, they certainly better live up to what they're doing. This does not happen with Mary Poppins Returns.

For each 4 words, only keep one

Meaningless text

Since of new resort stories ago, live they're not Poppins

Can We Resize Text? (4/4)

- In an image, individual pixels do not have strong meaning by themselves
  - Changing one pixel in an image will not change the meaning of the image
- In a text, each word influence the whole meaning

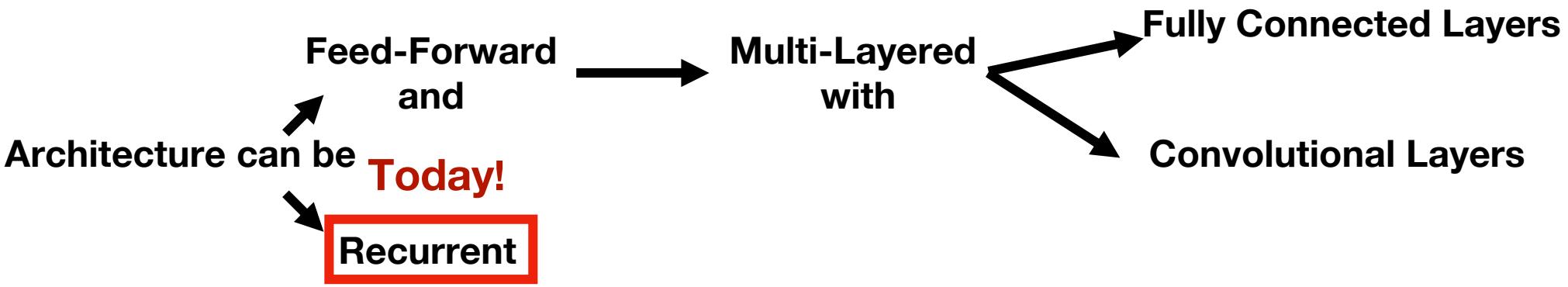
I do like you

I do not like you

- Because of that, we cannot easily manipulate the size of our input text like we can manipulate the size of input images
- Therefore, it is the network that has to be flexible

Neural Network Architectures

- In short:

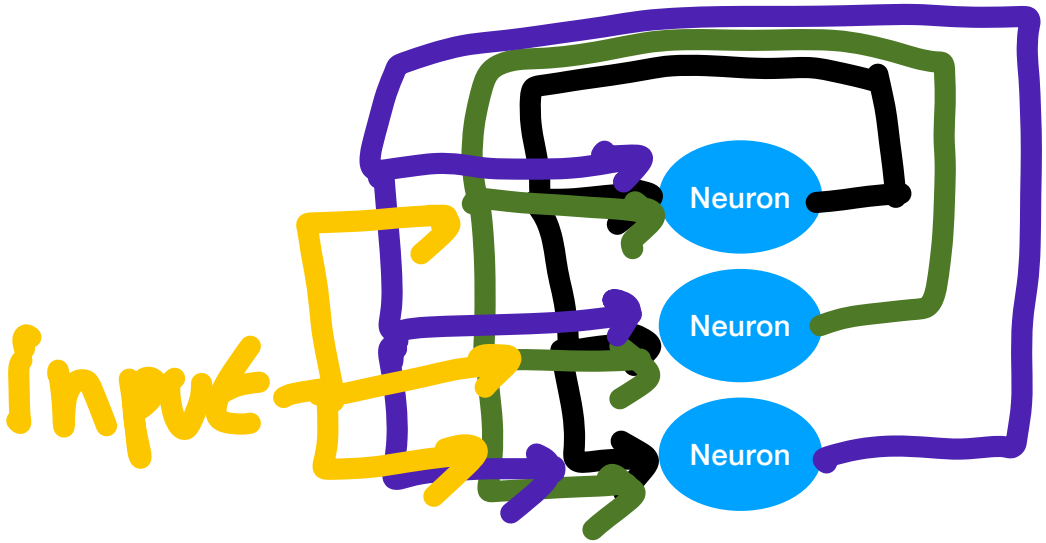


Recurrent Neural Network

- So far, we have only seen “Feed-Forward” Architectures
  - i.e., Information is processed from left to right

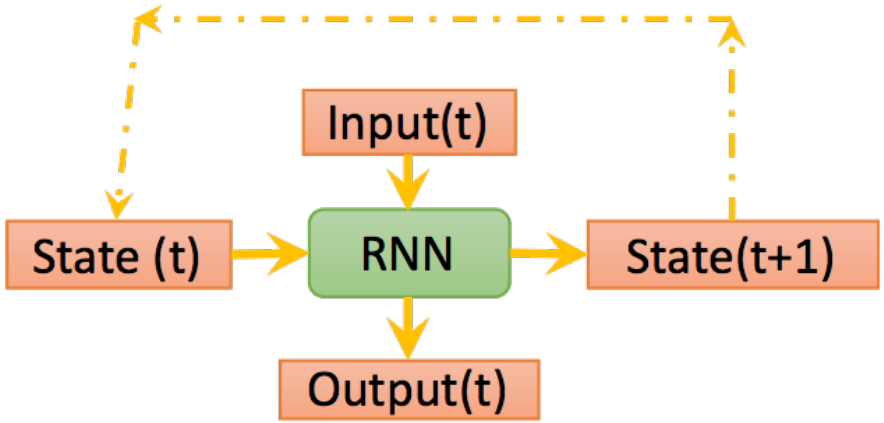
Recurrent Layer

- The idea of a recurrent Neural Network Layer is that, on top of its normal input, neurons in the layer will receive their **output** as **additional input**



Recurrent Layers

- In practice, we use more complicated architecture
- But the general idea is that part of the output of a Recurrent Neural Network (RNN) layer is used as additional input
- The part of the output that is redirected to the input will be seen as a “state” of the RNN

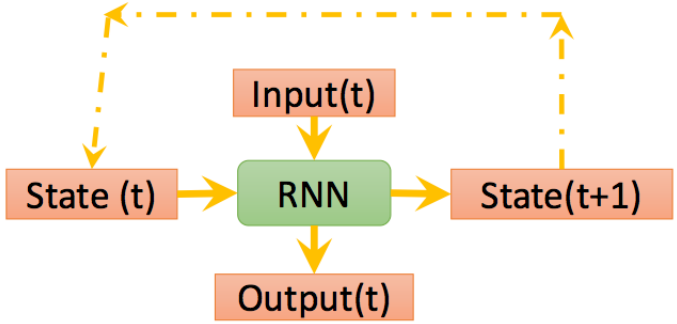


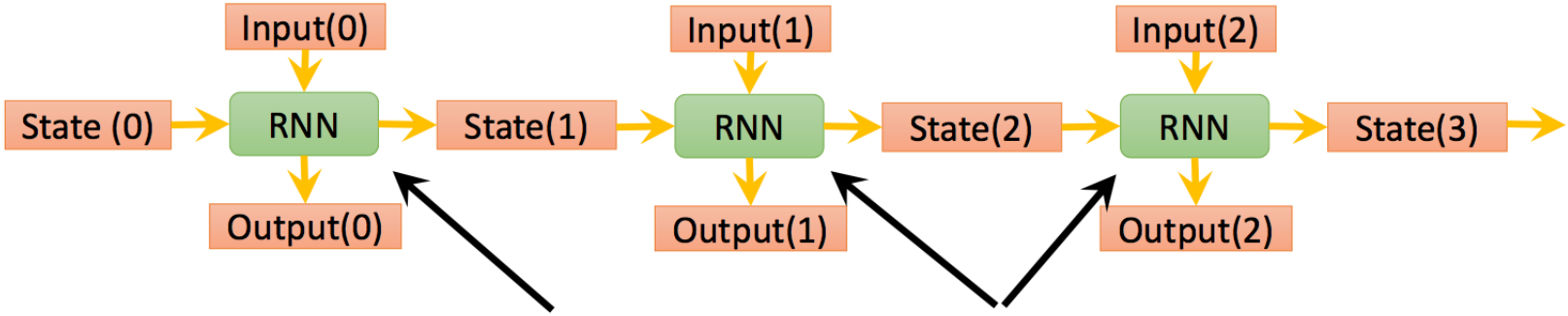
Recurrent Neural Networks

- These recurrent architectures are very useful for processing sequences
- Each element of the sequence is successively given as input to the RNN
- Because of the “state” output going back to the input, the RNN does not “forget” the previous input
- This way, after being given the whole input, the final state of the RNN contains information about the whole sequence

Recurrent Network

We can represent the Recurrent Neural Network unrolled in time:

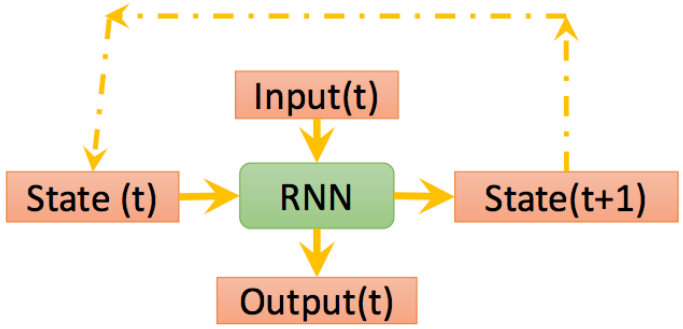


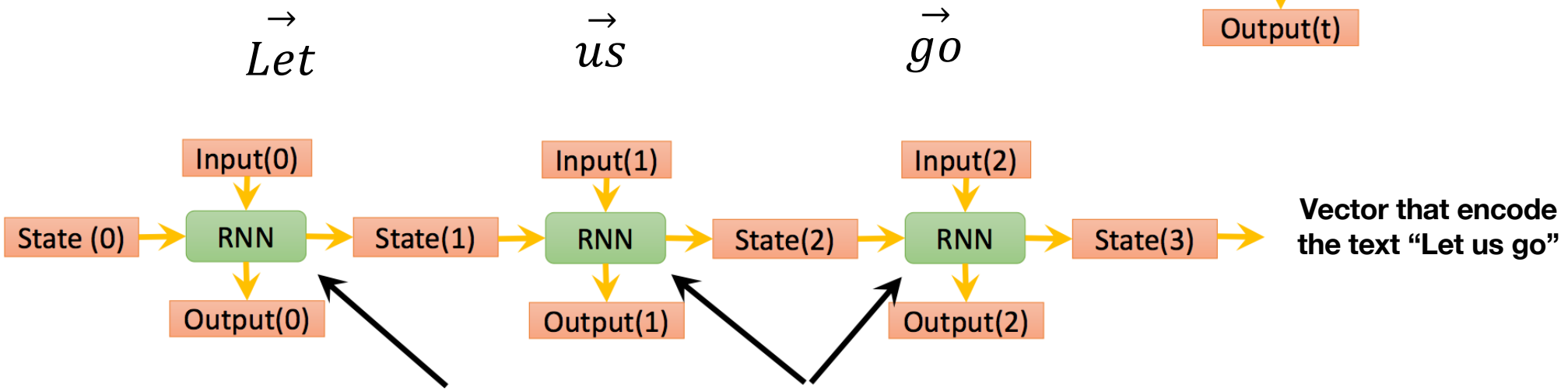


Represented as different cells, but actually using the same parameters

Recurrent Network

We can represent the Recurrent Neural Network unrolled in time:





Vector that encode the text "Let us go"

Represented as different cells, but actually using the same parameters

Difficulties in Text Process (1/2)

- Processing text with Neural Networks is actually more complex than processing images
- Mainly for 2 reasons:
  - Text is not “naturally” represented by numbers
  - Text can be of unlimited length
    - And it cannot be “naturally” reduced to a fixed size

Actually, Word Embeddings works well for that

Difficulties in Text Process (2/2)

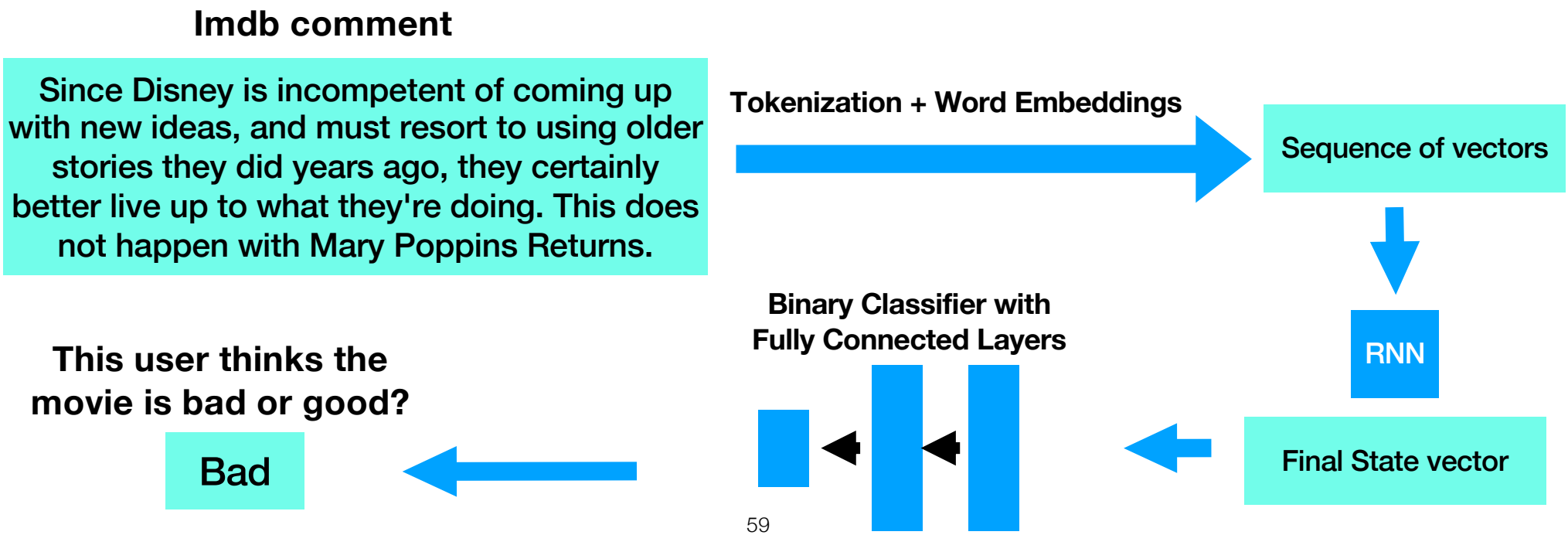
- Processing text with Neural Networks is actually more complex than processing images
- Mainly for 2 reasons:
  - Text is not “naturally” represented by numbers
  - Text can be of unlimited length
    - And it cannot be “naturally” reduced to a fixed size

Actually, Word Embeddings works well for that

Recurrent Neural Networks can process sequences of any size

Text Classifiers

- We now know how to process text as input, and we could easily build a “Sentiment classifier”, for example



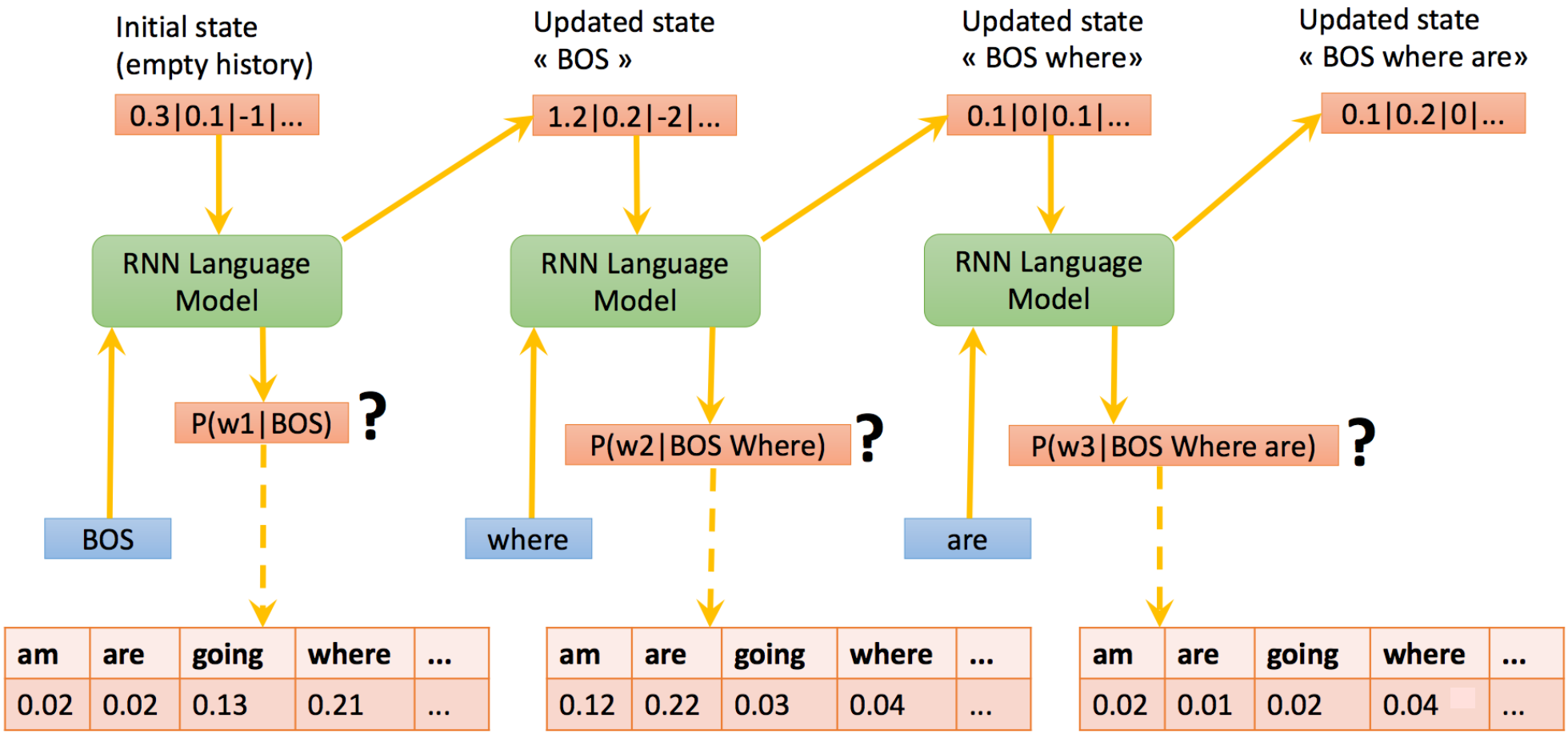
Generating Text

- We have seen how to use text as input
- What about text as output?
- One thing that is a bit easier with text than image is generation

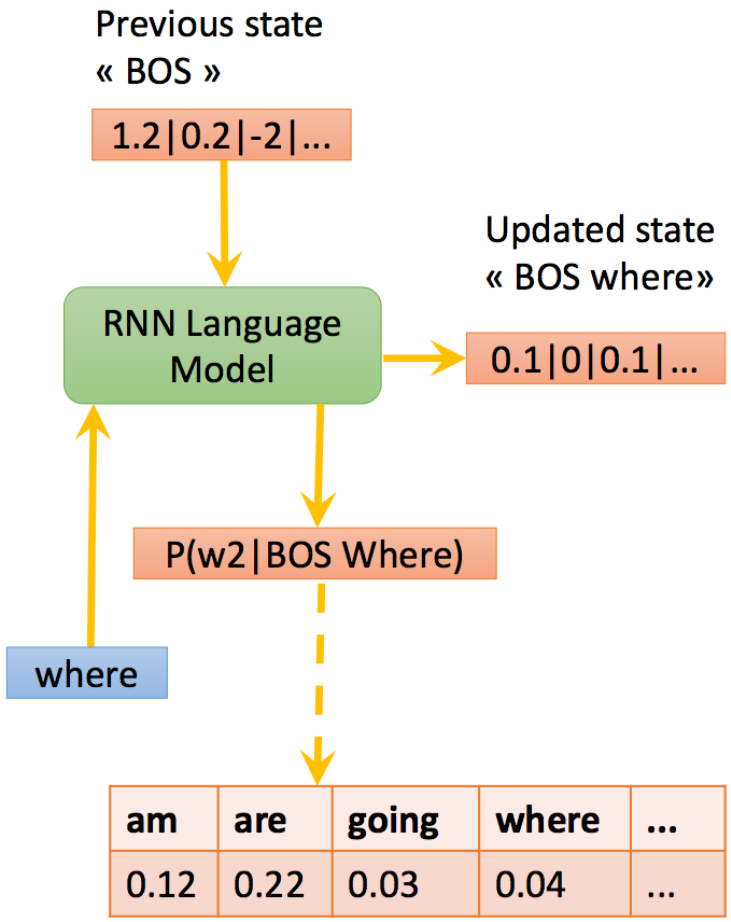
RNN Language Models

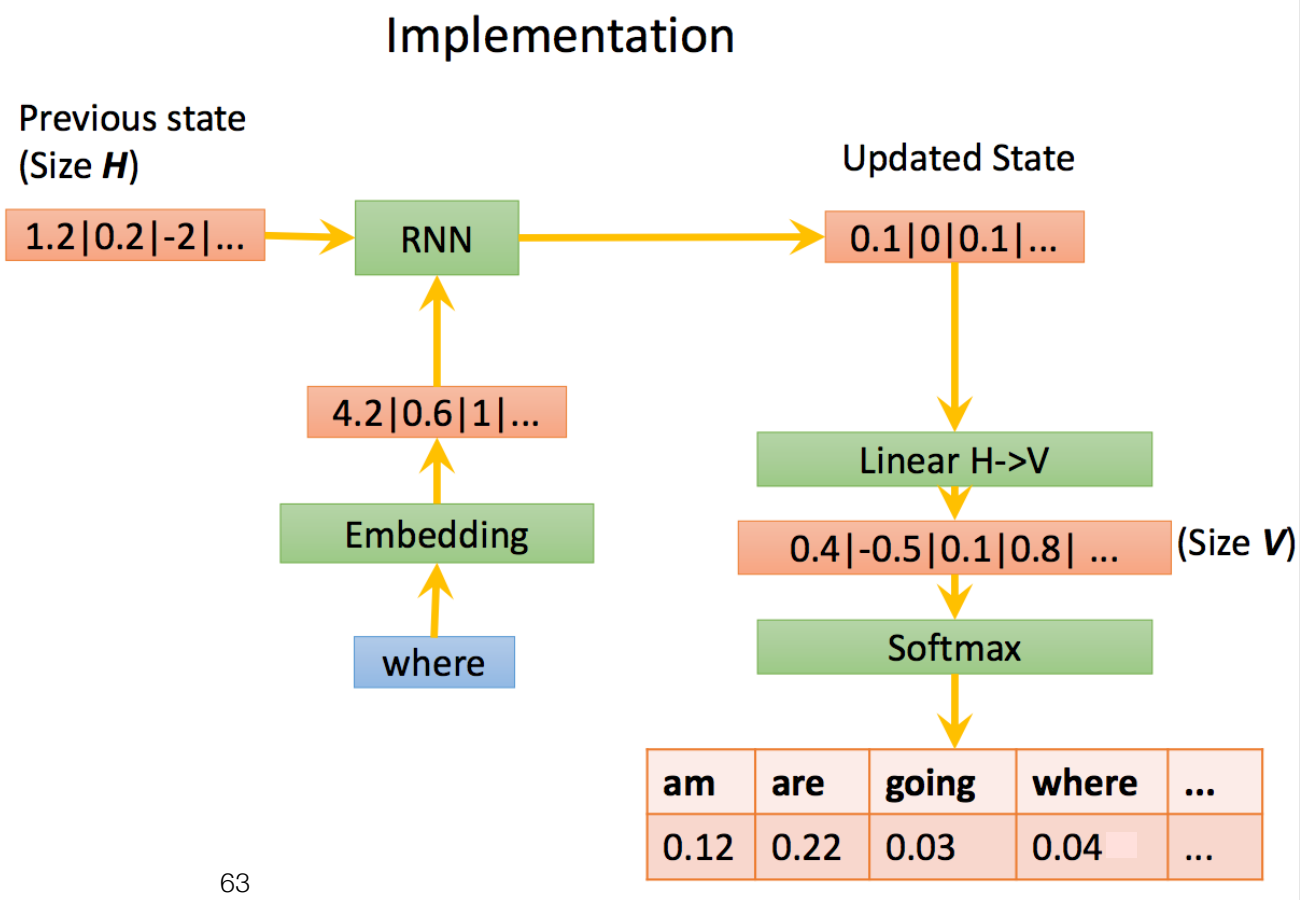
- A Language Model is a Model that can predict the probability of the next word in a sequence given the previous words
  - « Where is the ... » ? P(cat)=0.3, P(sky)= 0.02, ...
- How we do that with Neural Networks:
  - A state representing the *history* is maintained
    - here, history means the words of the translation that have been already generated
    - This state is simply a vector
  - The Language Model compute:
    - the probably of the next word given the history: P(w3 | w0 w1 w2)
    - the updated history state as we add more words to the translation

RNN Language Model

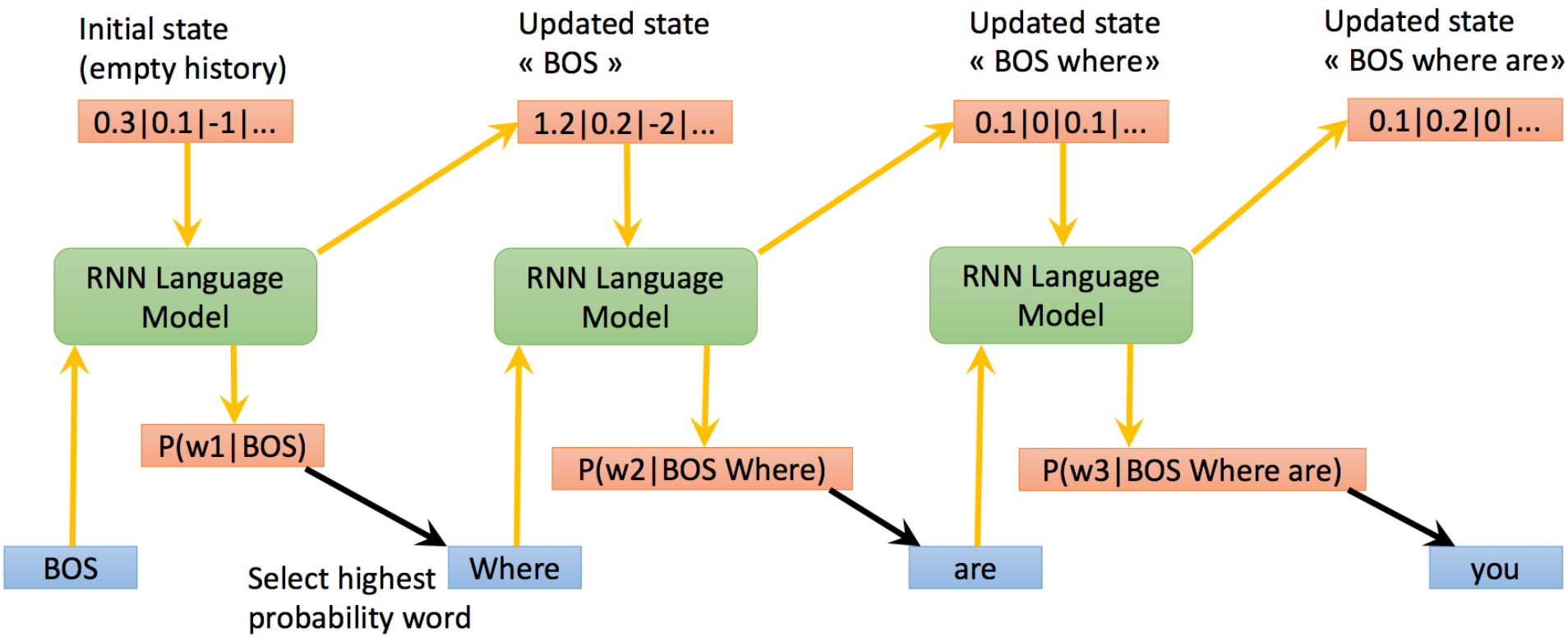


Implementation of the RNN LM

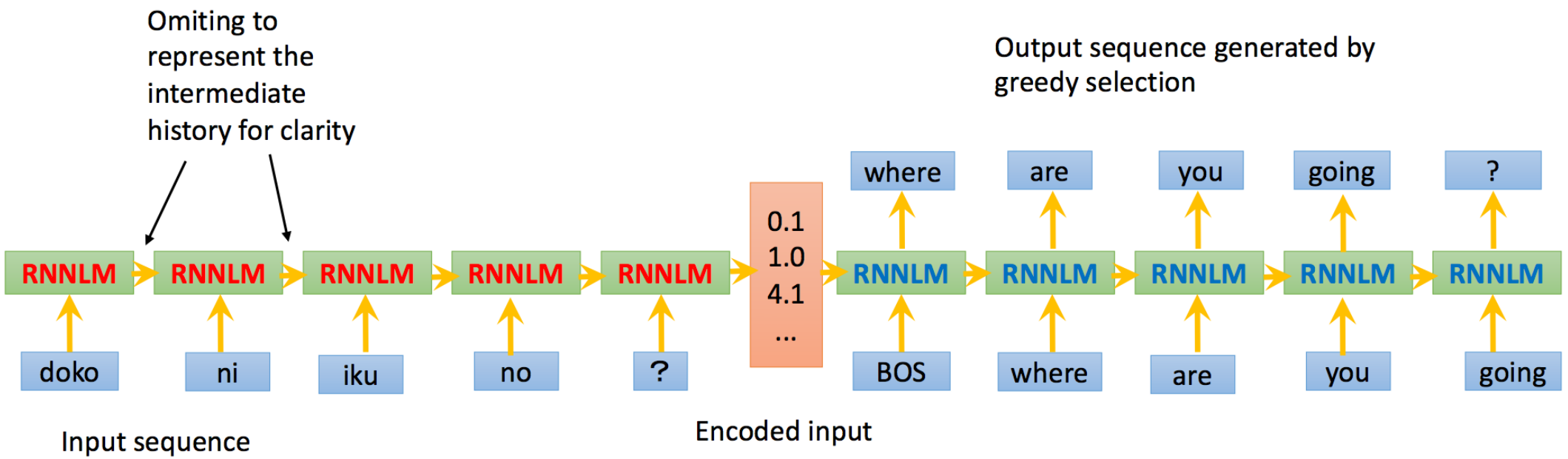




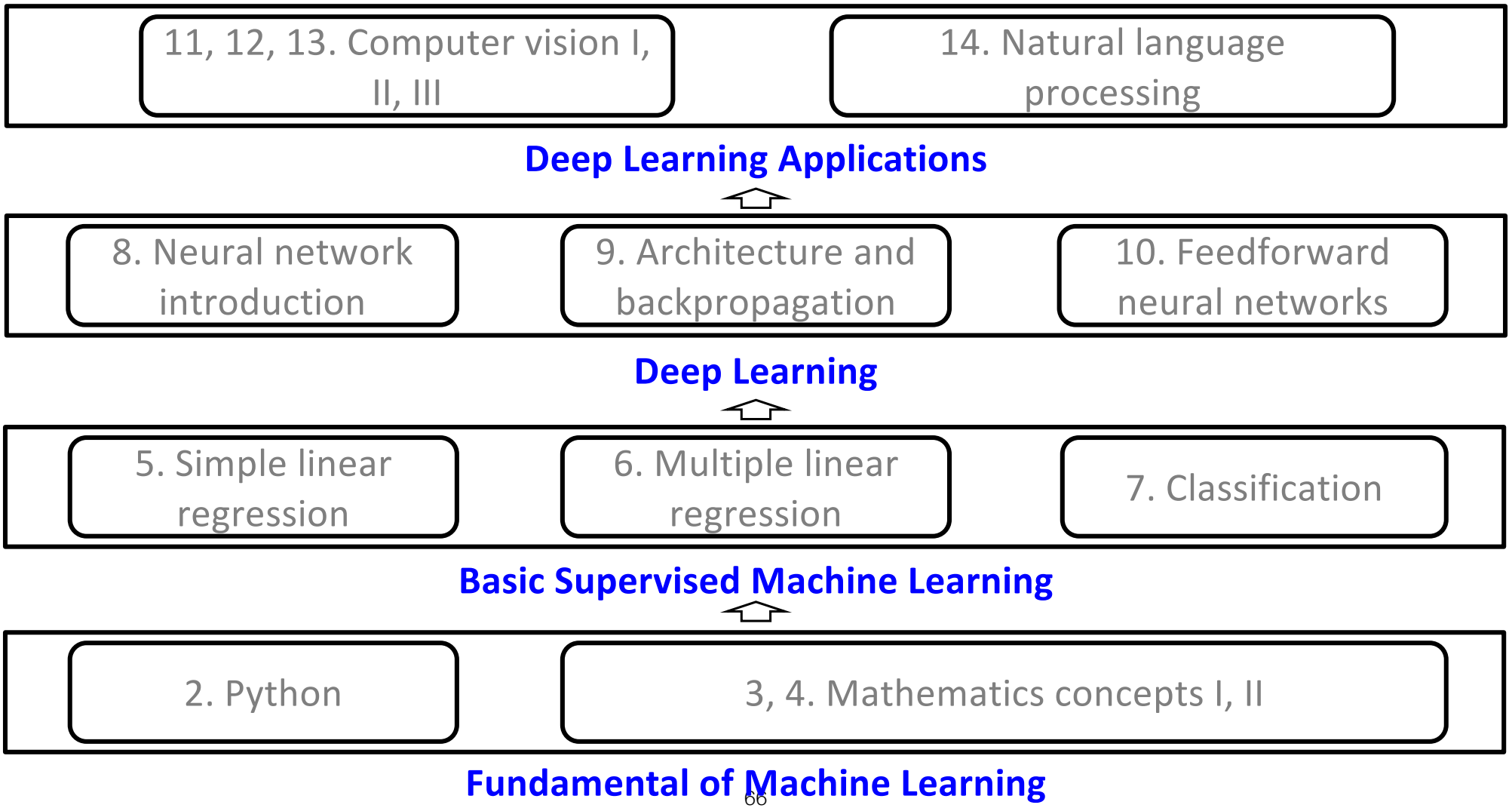
Generating Sequences: Greedy Decoding



Neural MT as a Neural Language Model



Review of This Course



Final Report

- Content: Describe a real problem in your study that can be solved by the models (either basic supervised machine learning or deep learning) introduced in this course together with the model structure and the reason for using the model. It would be better if you implemented the model and conducted experiments. The report should have no less than 2,000 words in English. Figures & diagrams are allowed to assist your illustration.

- Submit your final report in pdf named as **[student id_name]** via PandA by August 1st, 2025.