KISTI 한국과학기술정보연구원 Korea Institute of Science and Technology Information

2022

Deep Learning based Text Processing

Lec 11: Text generation with RNN

hsyi@kisti.re.kr

Hongsuk Yi (이홍석)

Overview of Course (III)

- Introduction to Recurrent Neural Network
  - ✓ Simple RNN, BPTT, Memory Cell
  - ✓ Code: Implementing an RNN with Keras
- Introduction to Long-Short Term Memroy
  - ✓ Cell state, LSTM, and GRU, and Applications
  - ✓ A Visual Guide to Recurrent Layers in Keras
  - ✓ Code: A simple LSTM layers
- Text generation with RNN
  - ✓ Tokenizer, Character-Level Language model
  - ✓ Code: Alice's Adventures in Wonderland
- Sequence to Sequence Learning model with RNN
  - ✓ Introduction to Seq2Seq and Attention model
  - ✓ Code: Character-Level Neural Machine Translation

Review the last class:

Long-Short Term Memory and RNN

Recurrent Neural Networks: Process Sequences

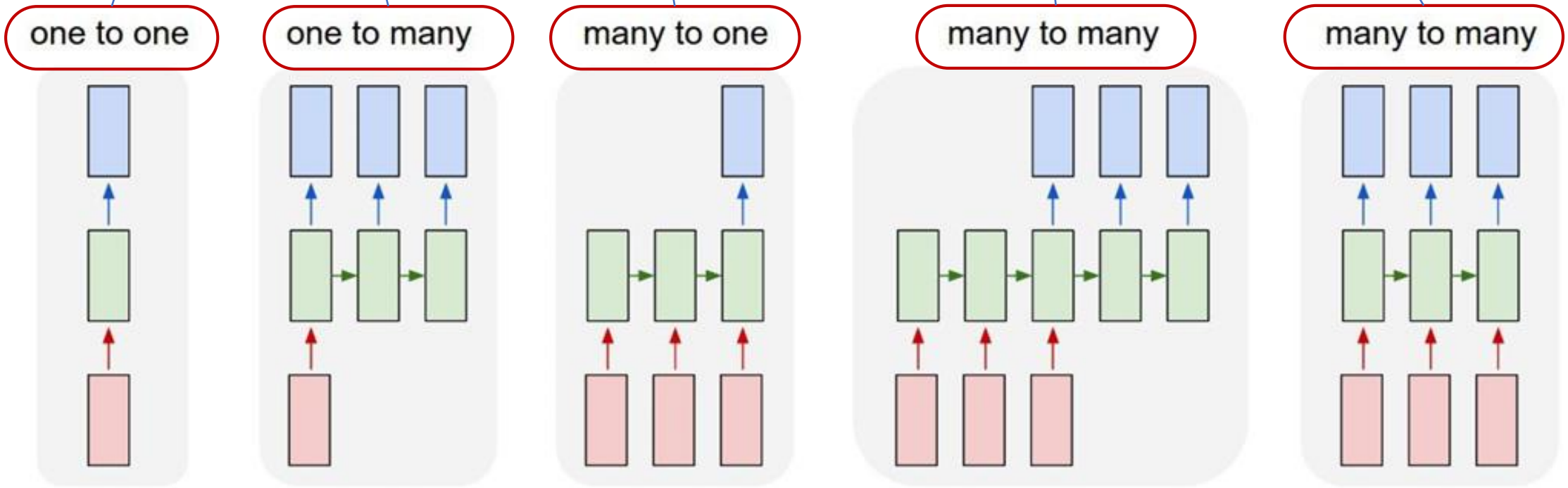
Vanilla
Neural Network

Image Captioning
(image→ seq of words)

Sentiment Classification
(Seq of words → sentiment)

Machine Translation
(Seq of words → seq of words)

Video classification
on frame level



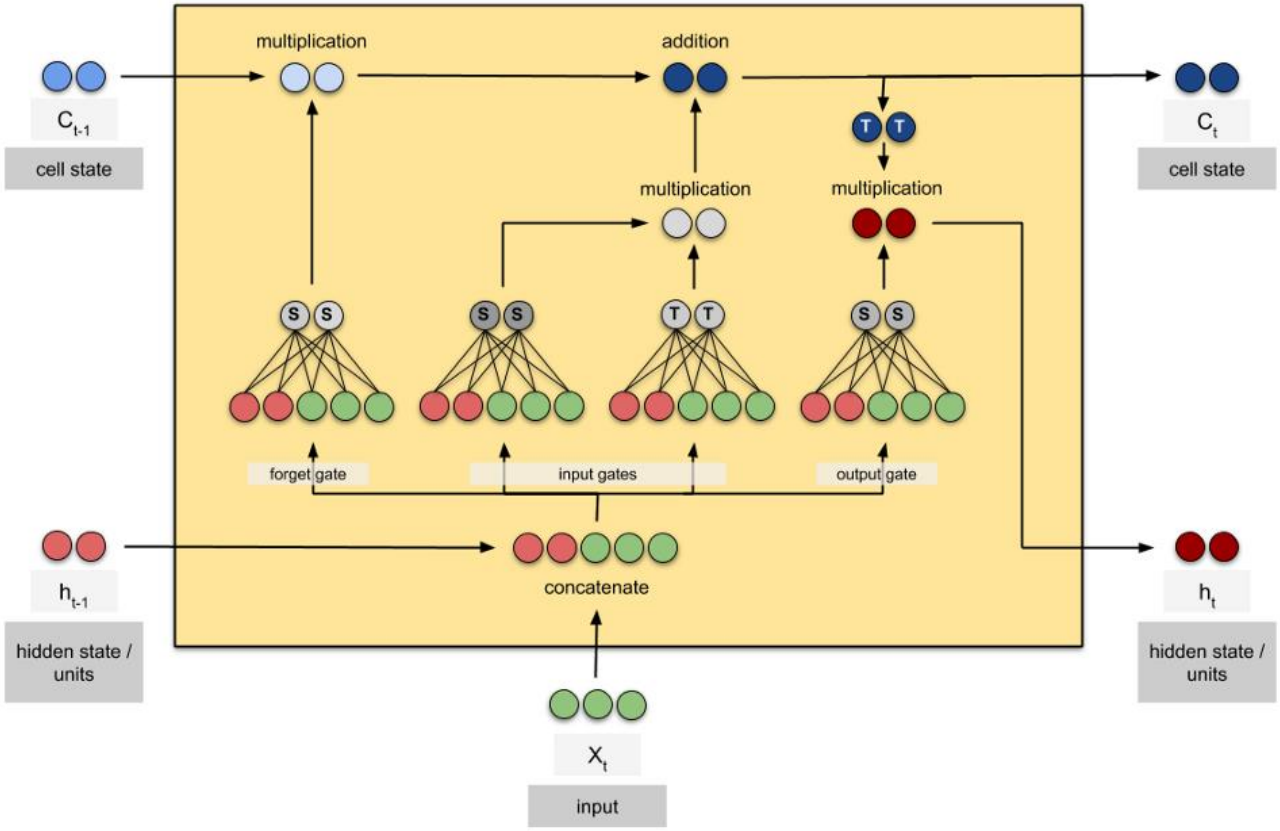
LSTM(Long-Short Term Memory)의 특징

❖ RNN의 기억 상태 (메모리, 혹은 Hidden State)를 제어하기 위해 '셀 상태 ' 를 도입

✓Cell State를 도입후 4개의 게이트로 RNN의 장기 메모리를 다룰 수 있는 장단기 메모리로 만듬

추가된
Cell State

RNN 메모리

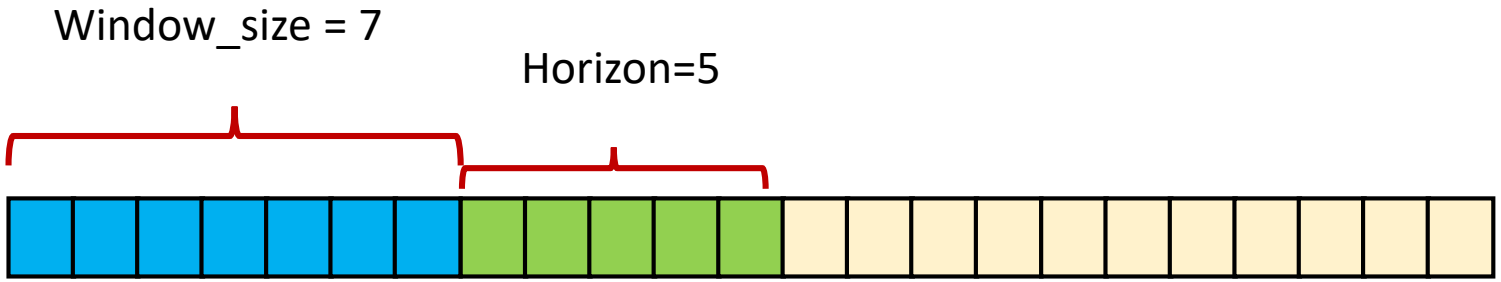


Modelling Experiments

- Two terms are really important in the type of forecasting model
  - ✓ **Window Size** :The number of timesteps we take to predict into the future
  - ✓ **Horizon** : The number of timesteps ahead into the future we predict.

```
model.add(SimpleRNN(5, input_shape=(7, 1), return_sequences=True))
```

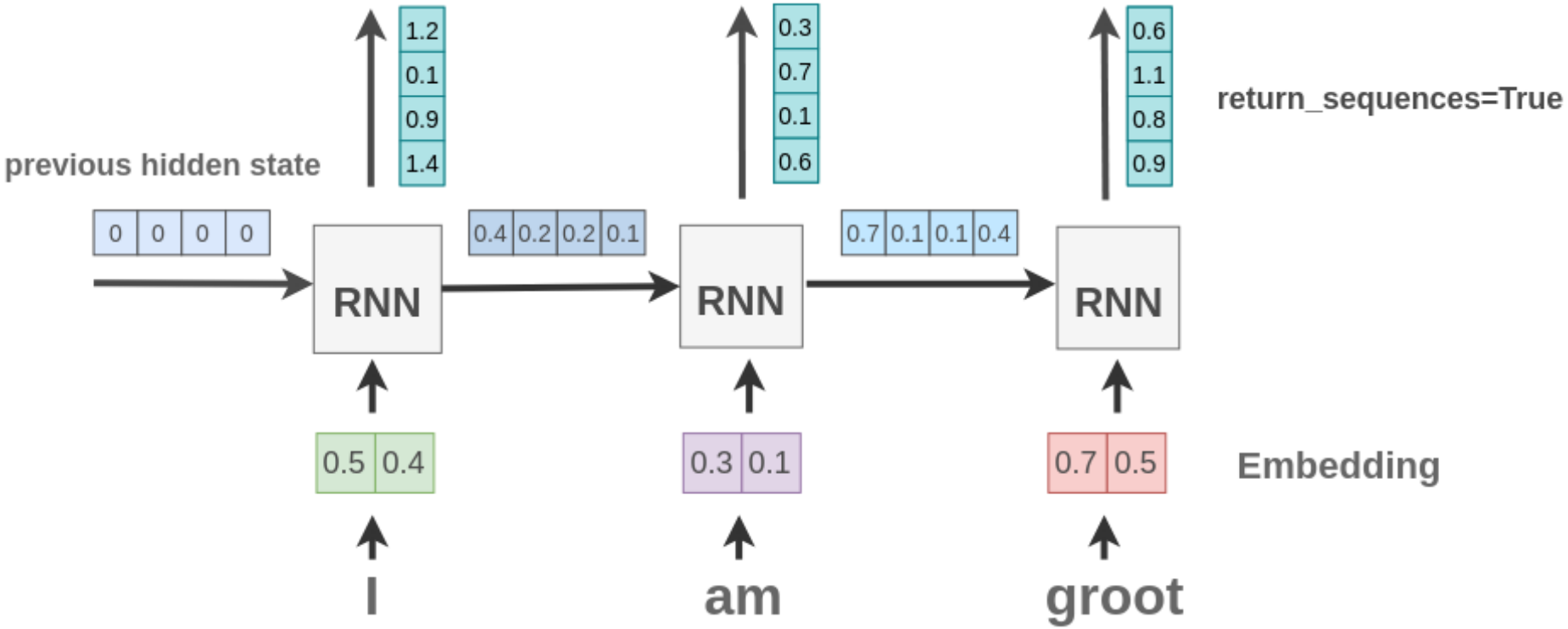
→ 자연어 처리에서는 임베딩 차원이 된다.



RNN with return_sequences =True

- The output from each unfolded RNN cell is returned instead of only the last cell.

```
model.add(SimpleRNN(4, input_shape=(3, 2), return_sequences=True))
```



Text Tockenizer

Tokenizing words

- **Token : Language elements that we can't share anymore**
  - ✓ Word tokenization divides sentences based on spacing as follows.

Text

"The cat sat on the mat."

↓

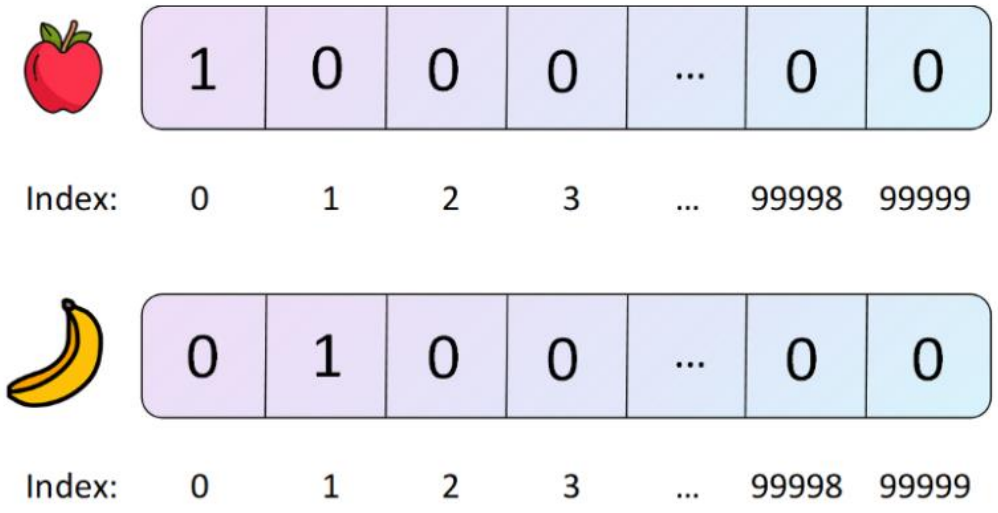
Tokens

"the", "cat", "sat", "on", "the", "mat", "."

- **Tokenizer : work to input text data into the neural network.**
  - ✓ The preprocessing process that converts it into an appropriate form through encoding

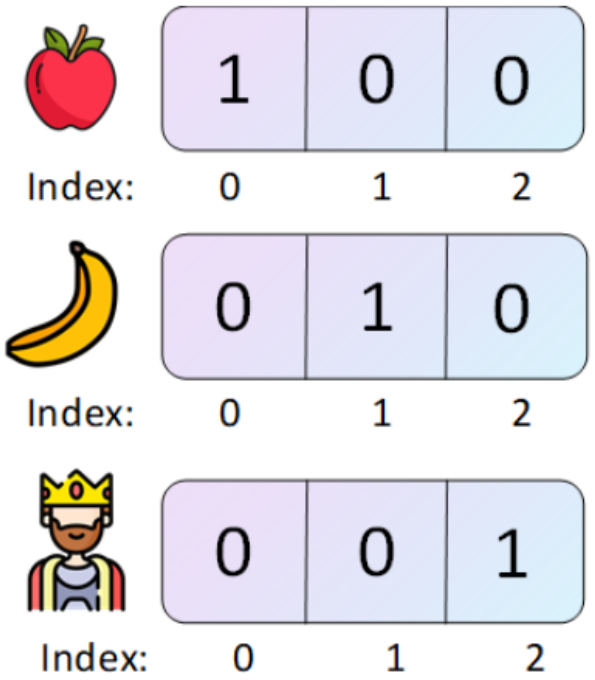
Why Word Embeddings are used?

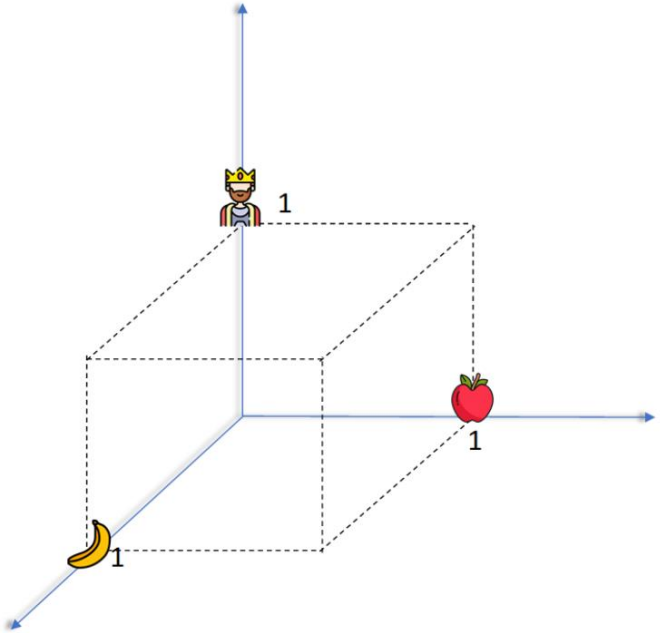
- we can use two more techniques
  - ✓ one-hot encoding
  - ✓ we can use unique numbers to represent words in a vocabulary.
    - In the case of text data, an embedding layer is basically used.
- A simple example of one-hot encoding



Word Embedding (1/2)

- Word embeddings are basically a form of word representation that bridges the human understanding of language to that of a machine.

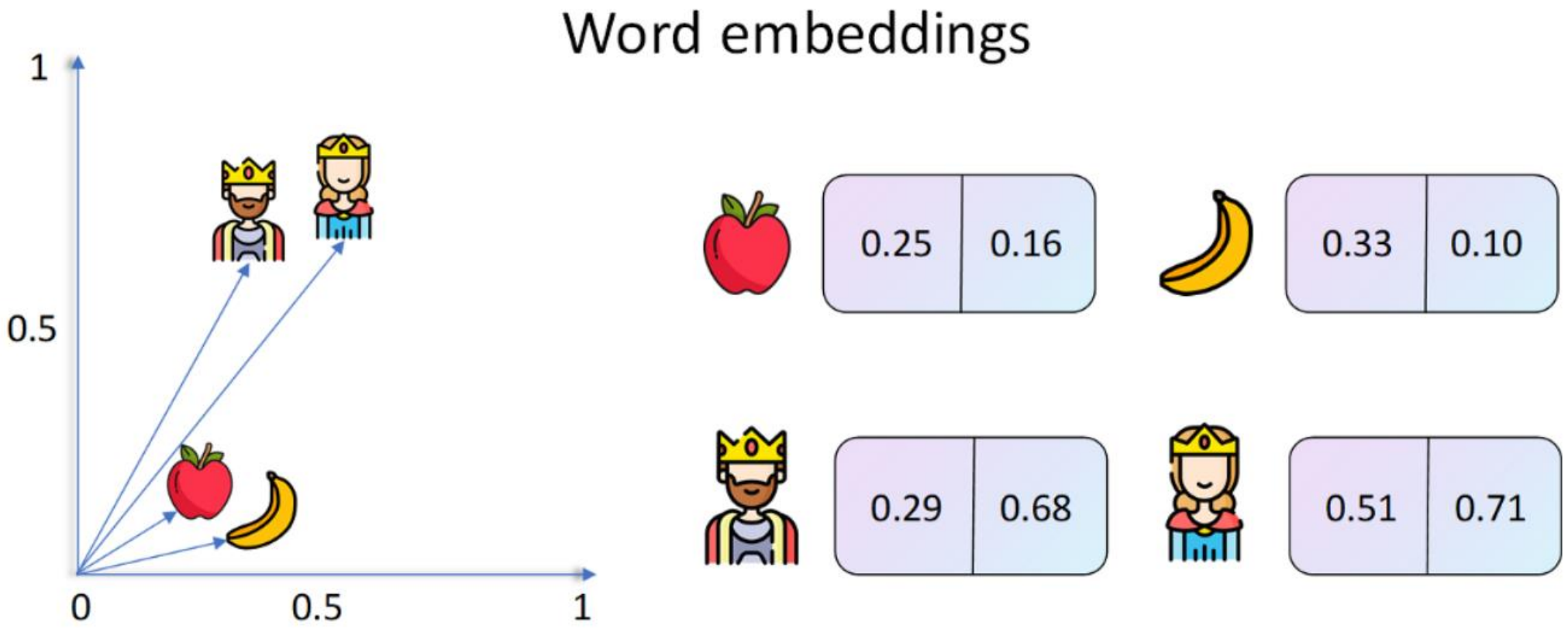




What is Word Embedding? (2/2)

❖ Representations of text in an n-dimensional space

✓ where two similar words are represented by almost similar vectors that are very closely placed in a vector space.



A simple example: Word embedding

- **vocab_size = 30**
  - ✓ Usually the vocabulary size is thousands
- **seq_length = 5**
  - ✓ Usually, sentences consist of more than five words

"This is a small vector"

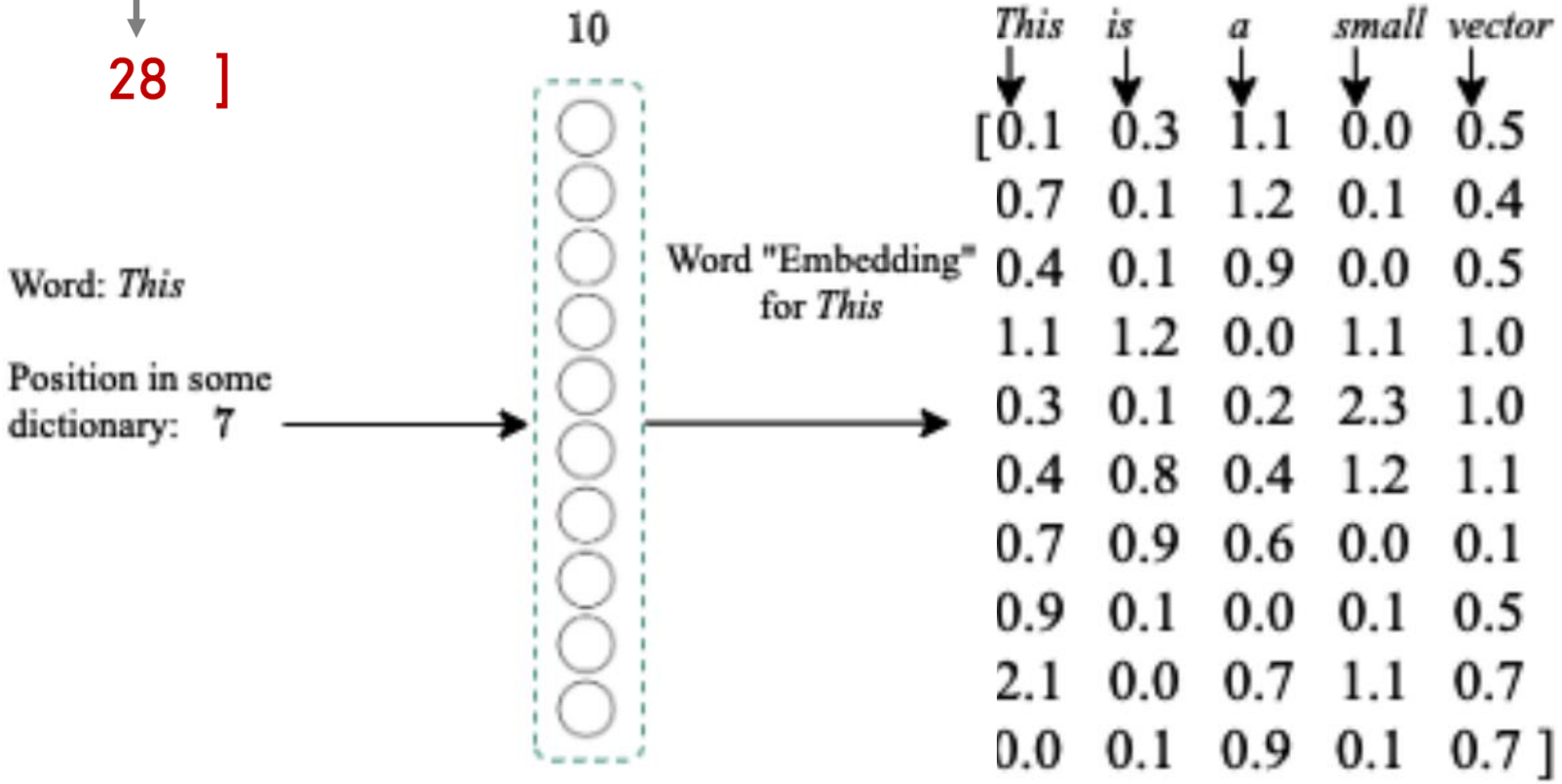
[7 0 6 1 28]

A simple example: Word embedding

- **vocab_size = 30, seq_length = 5**

"This is a small vector"

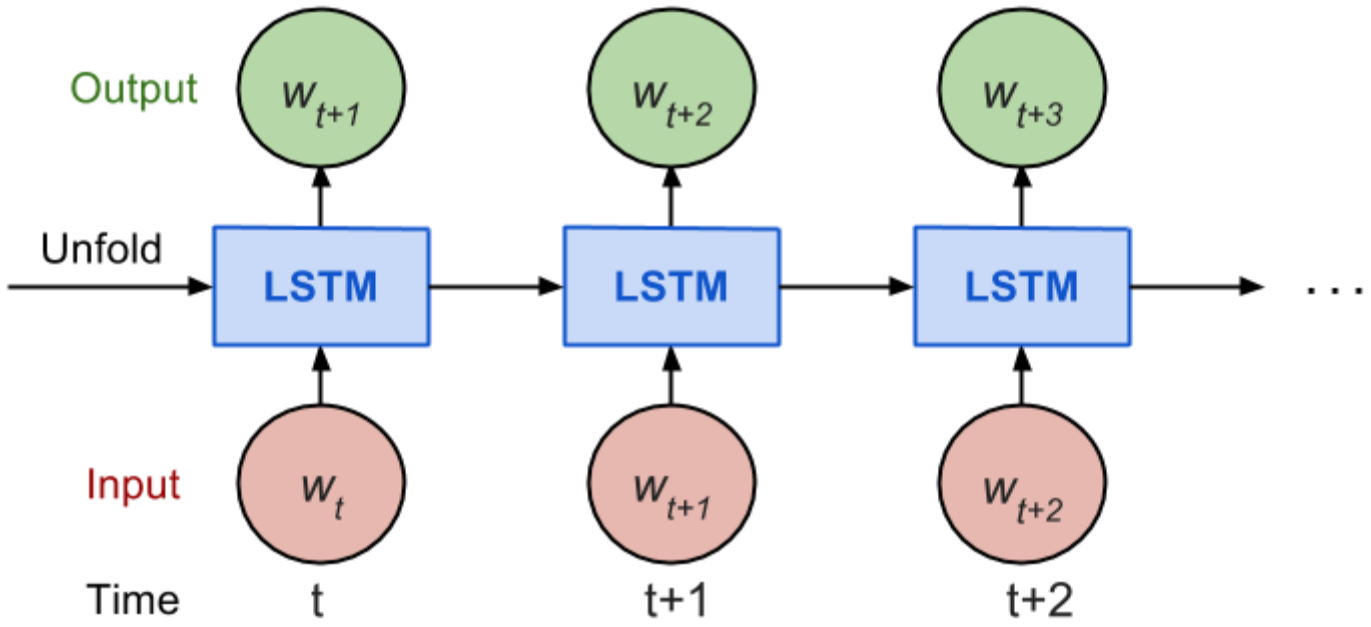
[7 0 6 1 28]



Character-level language model

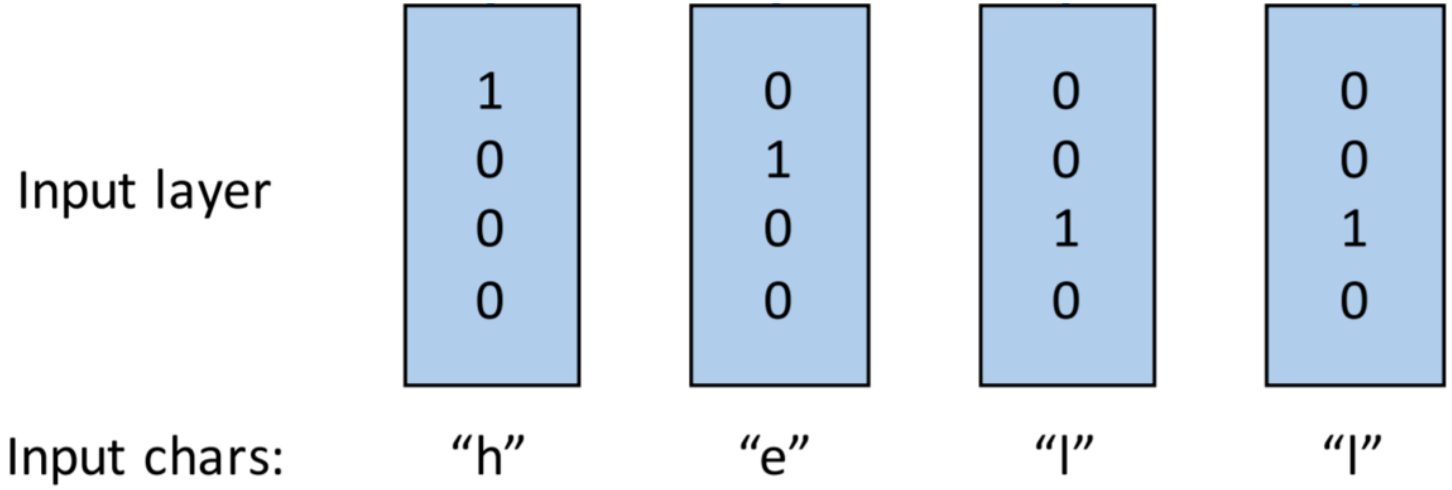
Text Generation : Many-to-Many Application

❖ architecture of the model

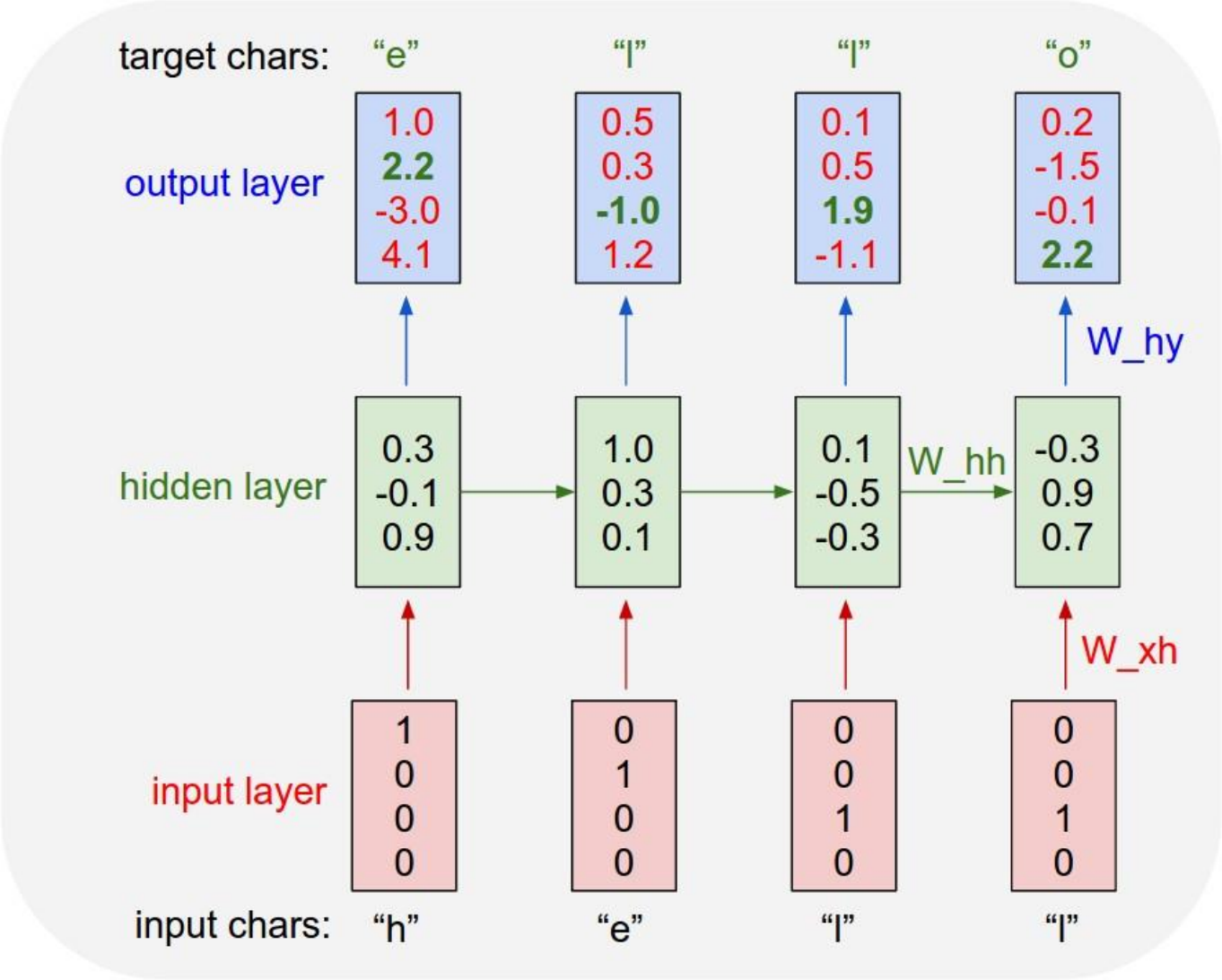


Example training sequence : “hello”

- Andrej Karpathy blog
  - ✓ http://karpathy.github.io/2015/05/21/rnn-effectiveness/
- Character-level language model : we only had a vocabulary of “hell”: [h,e,l,o]
  - ✓ Encode each character into a vector using 1-of-k encoding



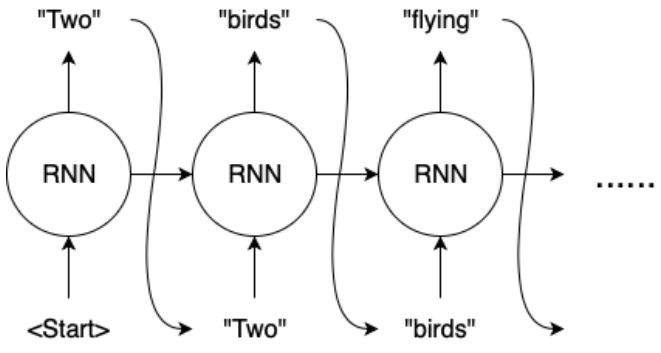
Character-level language model



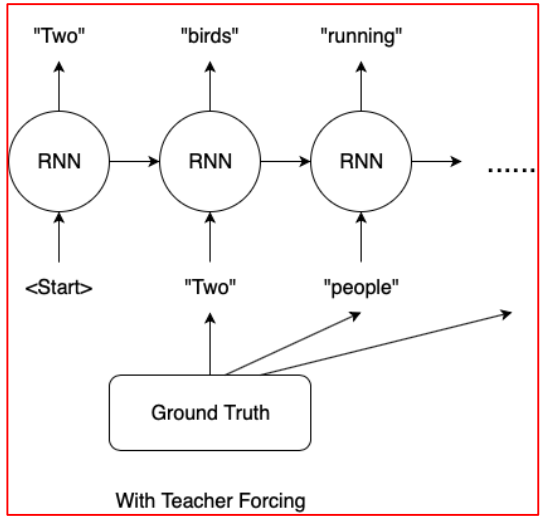
We want the green numbers to be high and red numbers to be low.

Teacher forcing

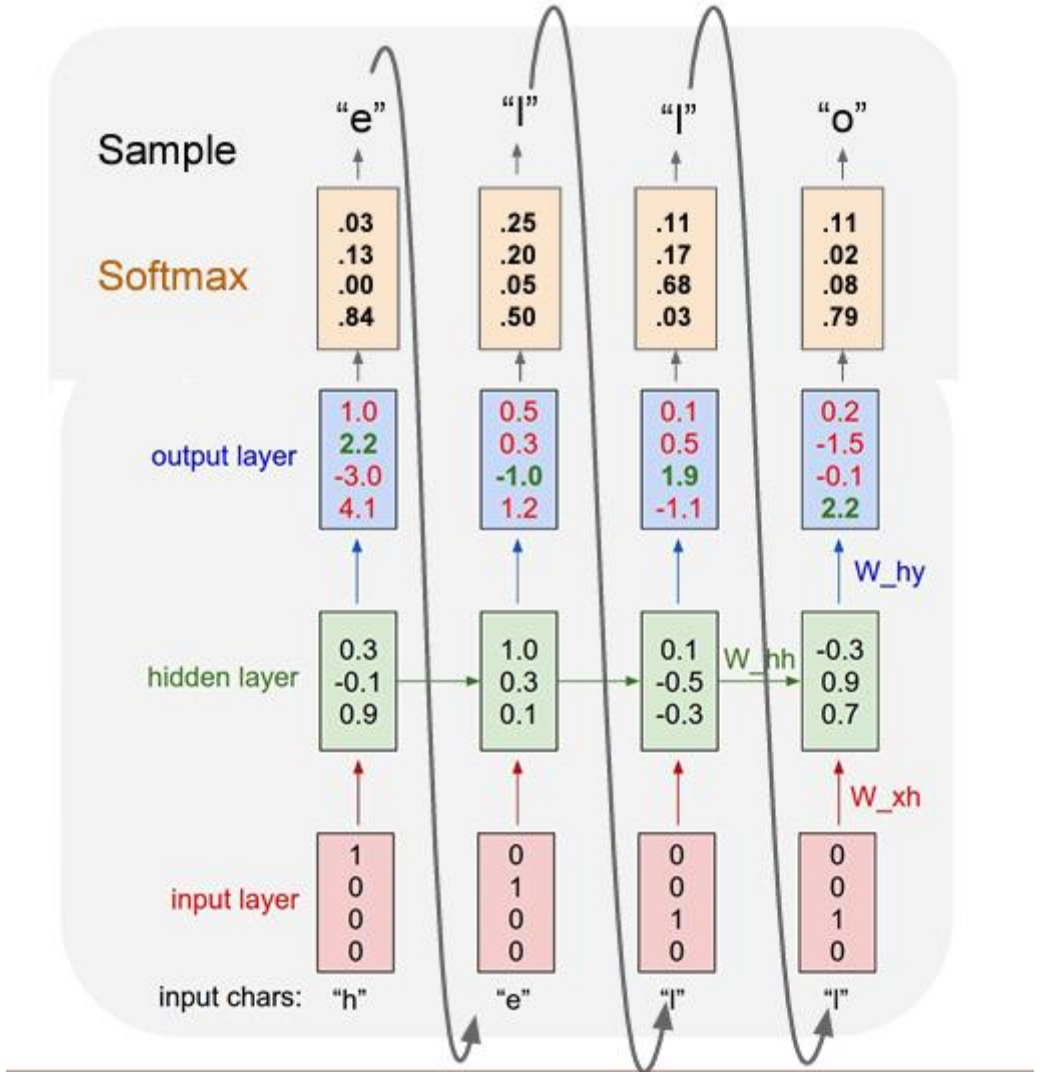
Teacher forcing : Character-level language model



Without Teacher Forcing

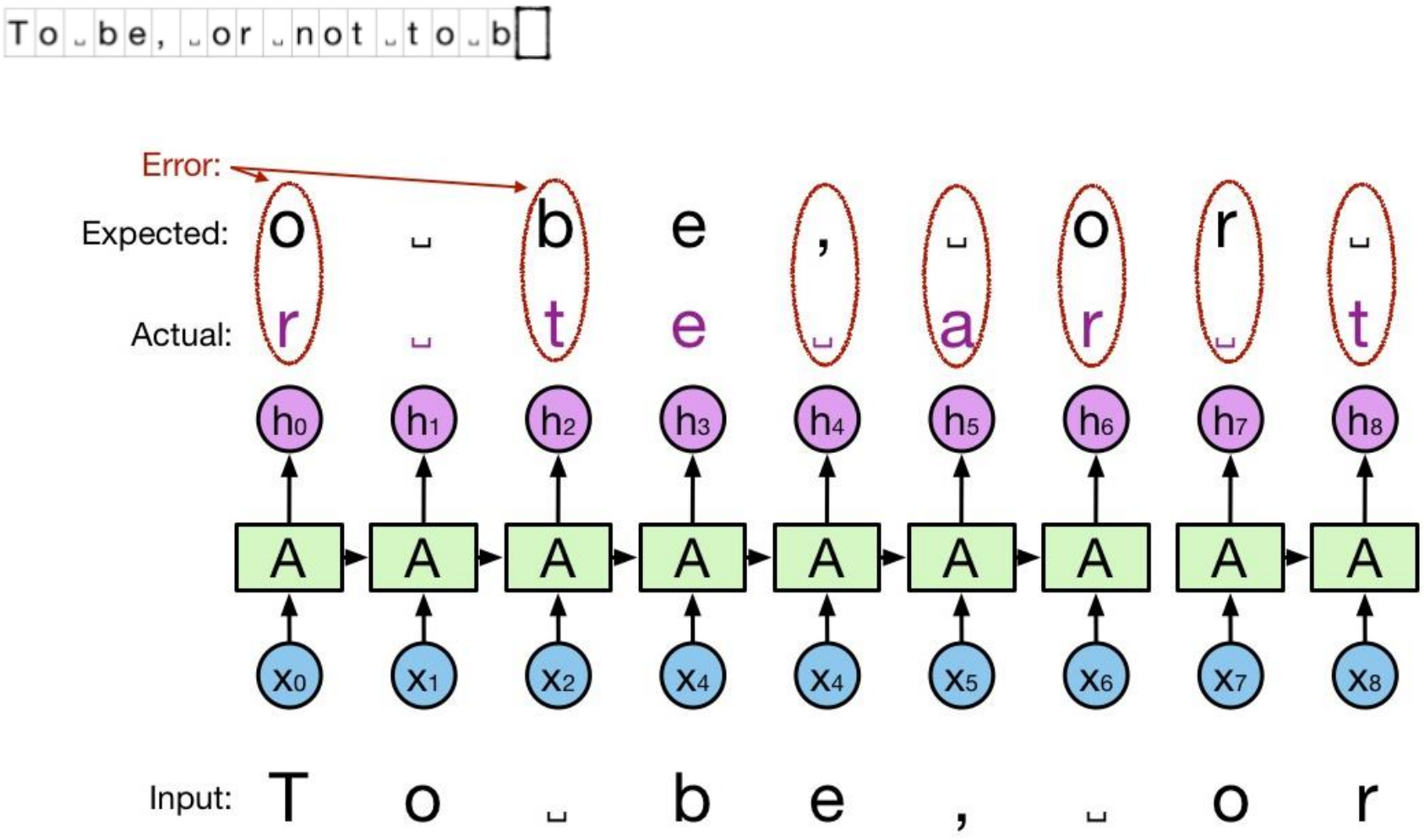


With Teacher Forcing



Generating Text Using a Char-RNN

Prediction of the next character



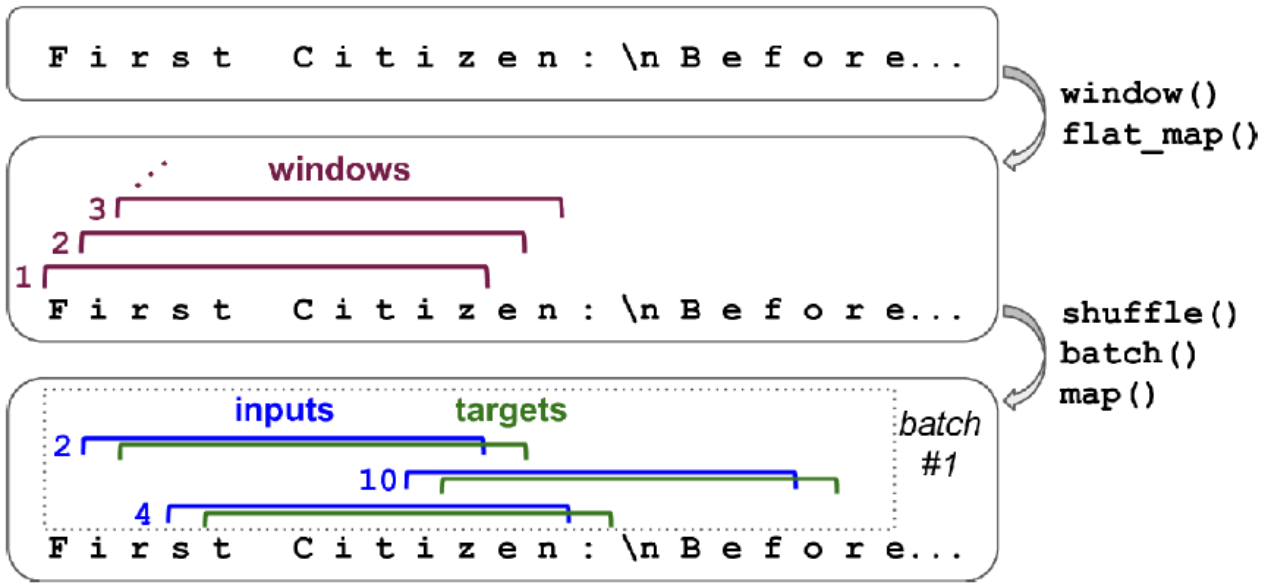
Creating the Training Dataset

❖ Downloading the data from Andrej Karpathy's Char-RNN project :

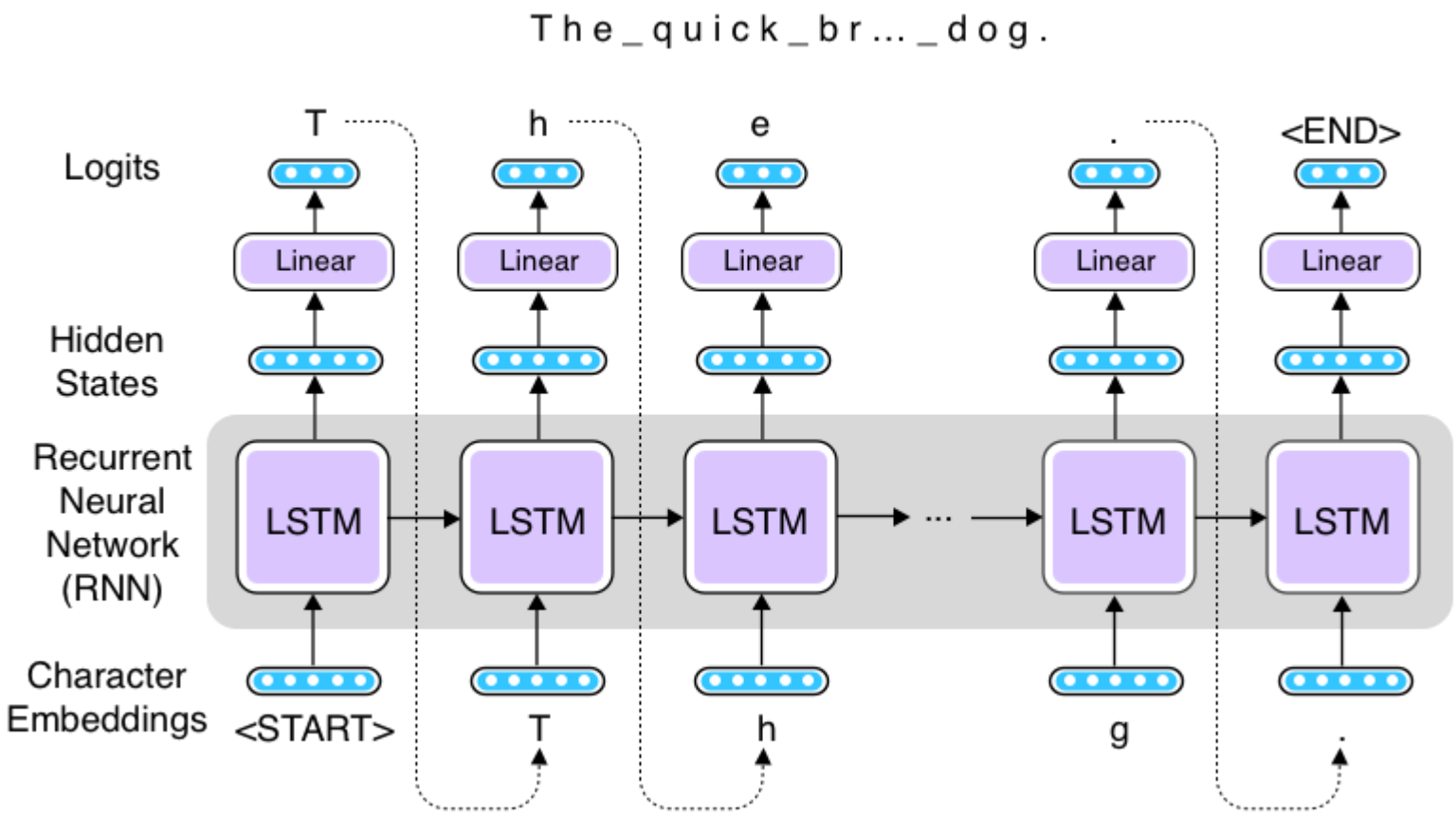
```
shakespeare_url = "https://raw.githubusercontent.com/karpathy/char-rnn/master/data/tinyshakespeare/input.txt"
```

```
print(shakespeare_text[:248])
```

```
First Citizen:
Before we proceed any further, hear me speak.

All:
Speak, speak.

First Citizen:
You are all resolved rather to die than to famish?

All:
Resolved. resolved.

First Citizen:
First, you know Caius Marcius is chief enemy to the people
```



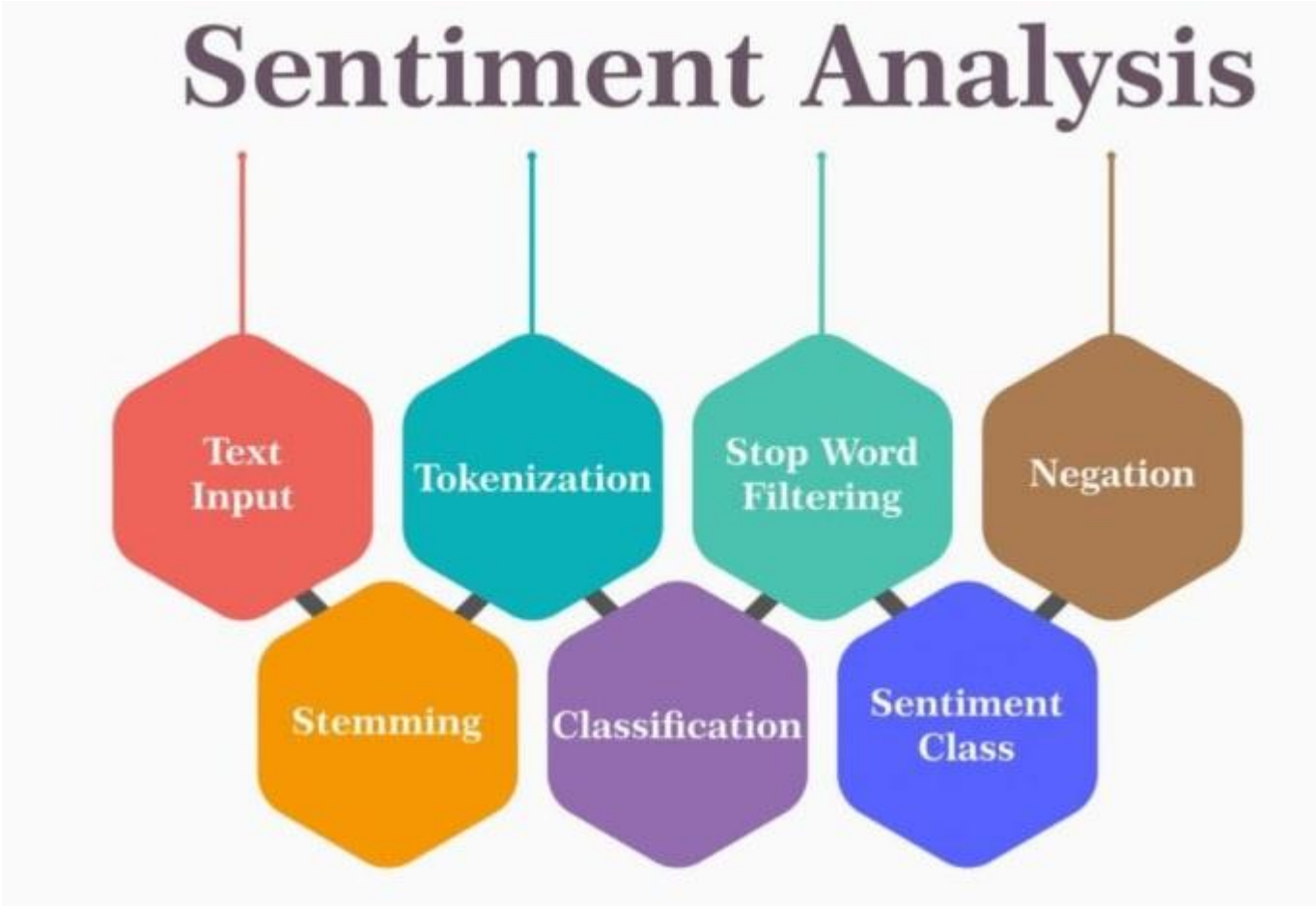
Text generation using RNN



<source> http://www.realworldnlpbook.com/blog/training-a-shakespeare-reciting-monkey-using-rl-and-seqgan.html

Text Classification Using RNN

Sentiment Analysis : Positive, Neutral, Negative



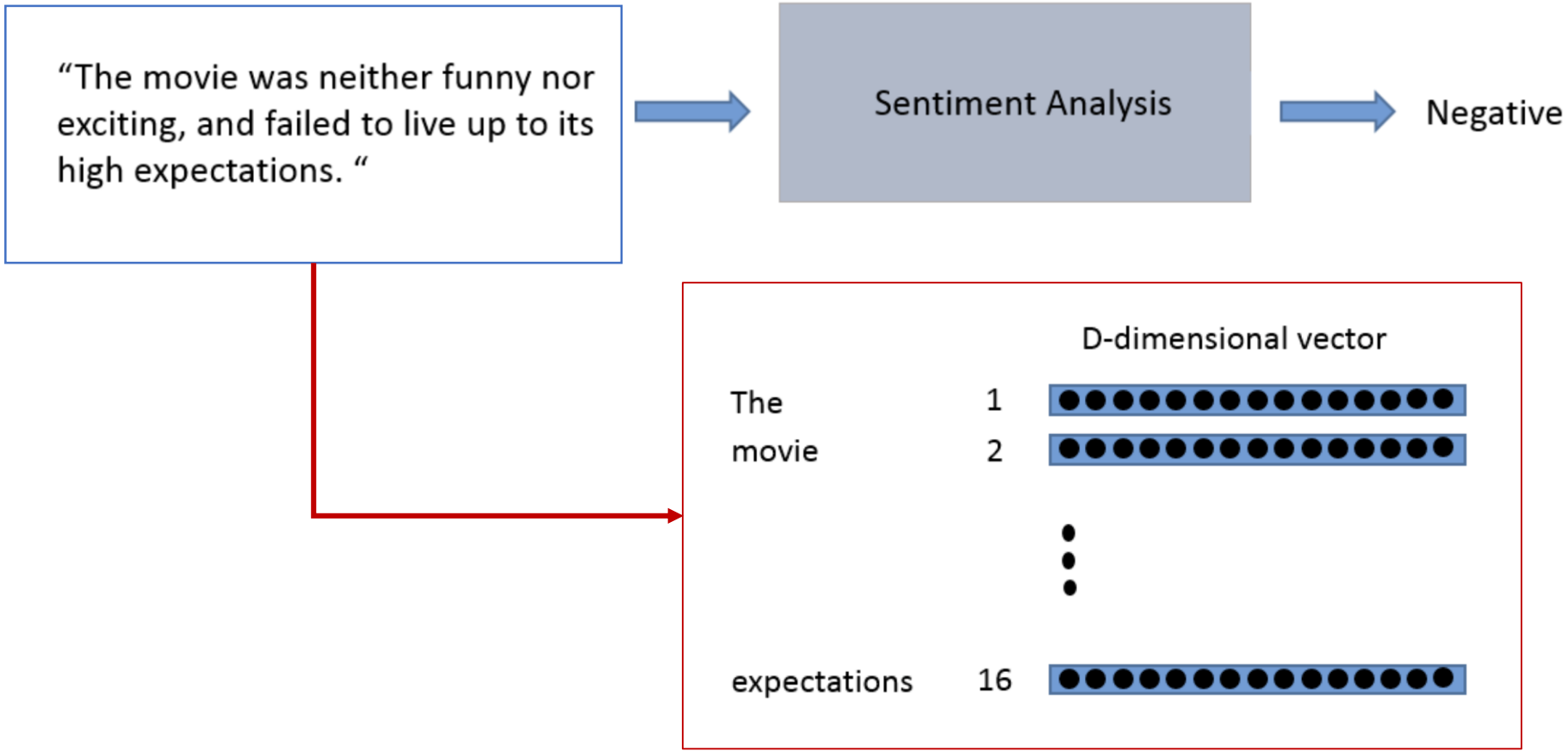
Sentiment Analysis : IMDB (Internet Movie Database

❖ Text and Target

✓ Positive, Negative, Neutral, Happy, Sad

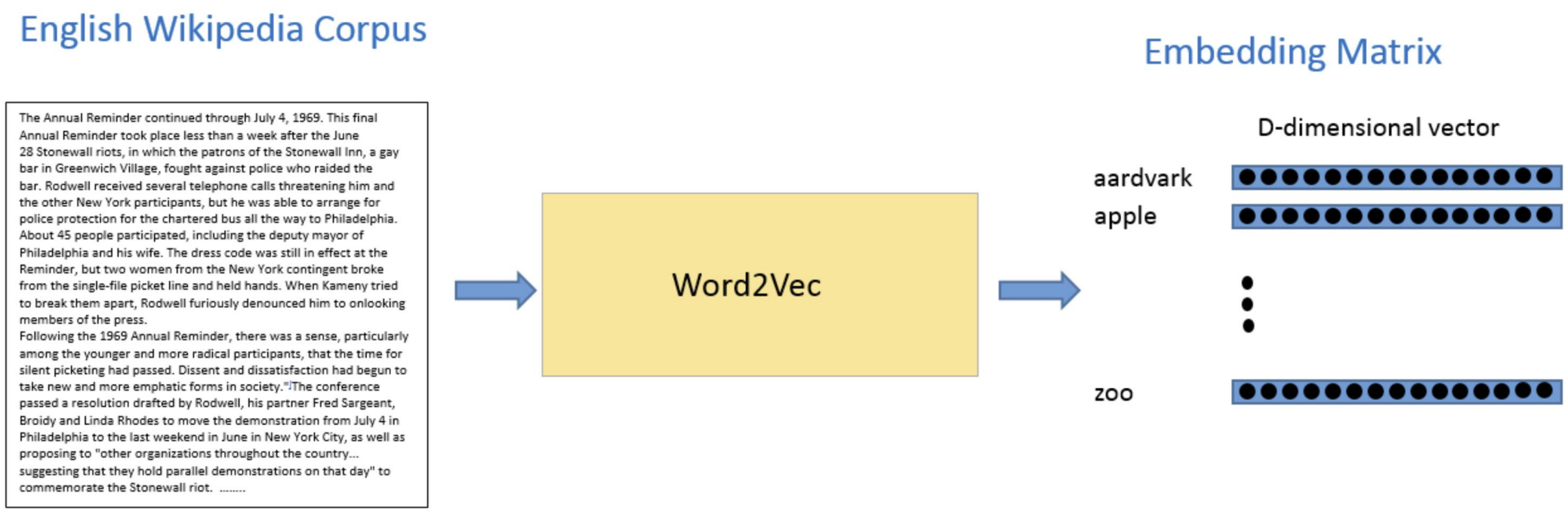
	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The filming tec...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
5	Probably my all-time favorite movie, a story o...	positive
6	I sure would like to see a resurrection of a u...	positive
7	This show was an amazing, fresh & innovative i...	negative
8	Encouraged by the positive comments about this...	negative

❖ Convert each word in the sentence to a vector

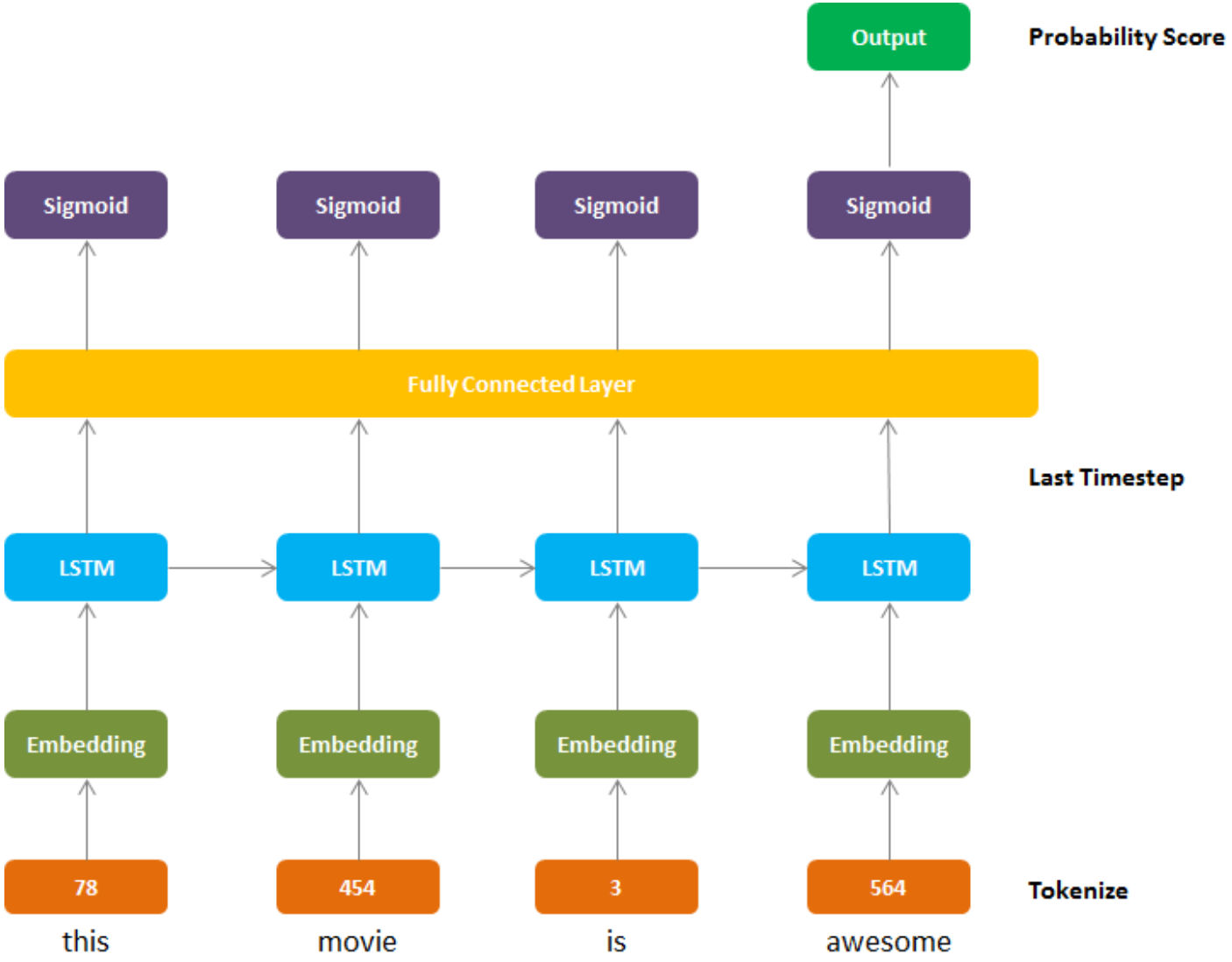


Embedding matrix

- The output of a Word2Vec model is called an embedding matrix
  - This embedding matrix will contain vectors for every distinct word in the training corpus.



Many-to-One Sequence Model



KISTI SINCE 1962

2022

Korea Institute of Science
and Technology Information

TRUST KISTI