Deep Learning Bootcamp

A hands-on intensive workshop covering neural fundamentals,
computer vision, and natural language processing using modern deep learning tools.

by: Lama Ayash

Day 3

Course Outlines

01 Natural Language Processing

02 NLP Tasks

03 Text Preprocessing

04 Text Representation

05 Recurrent neural network

06 Transformers

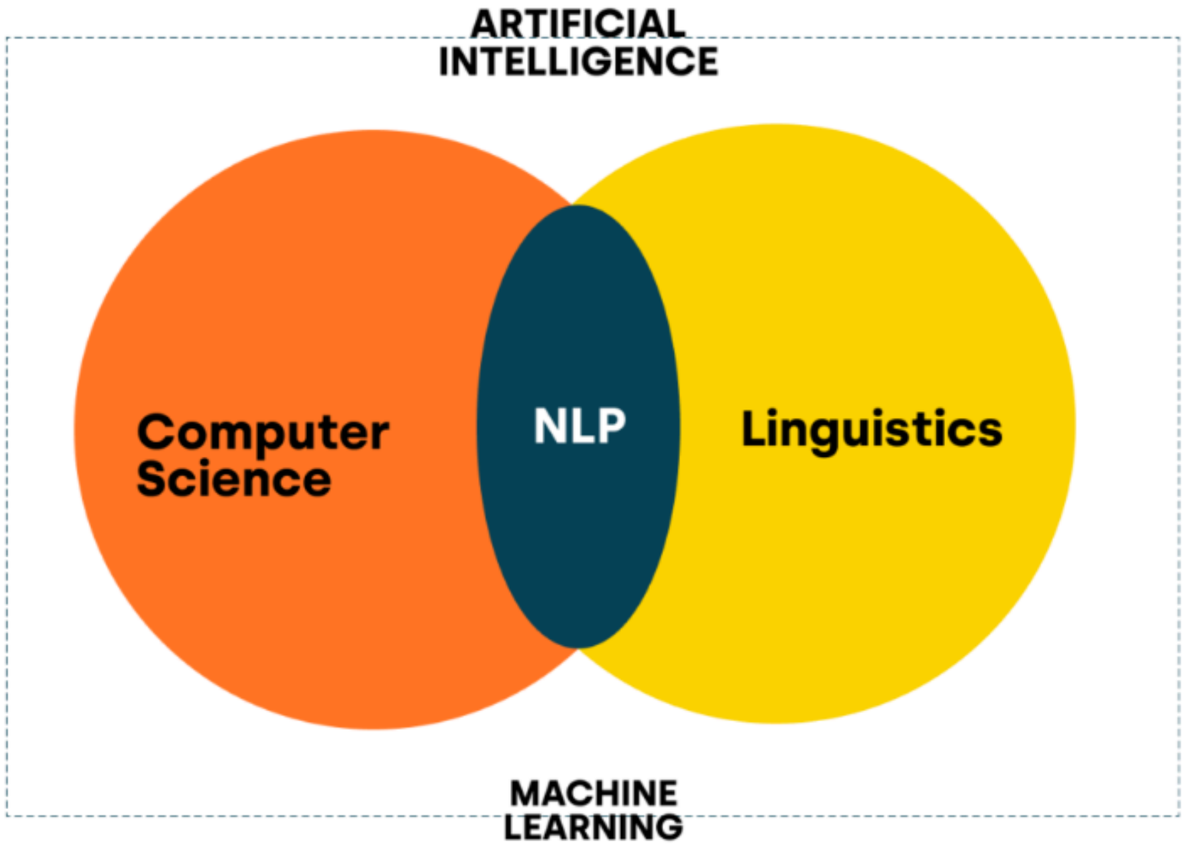
07 Q&A

Natural Language Processing

Language is the primary way humans express ideas, emotions, and intentions

Natural Language Processing

Natural Language Processing (NLP) is a field of Artificial Intelligence that enables computers to understand, interpret, and generate human language in both text and speech form.

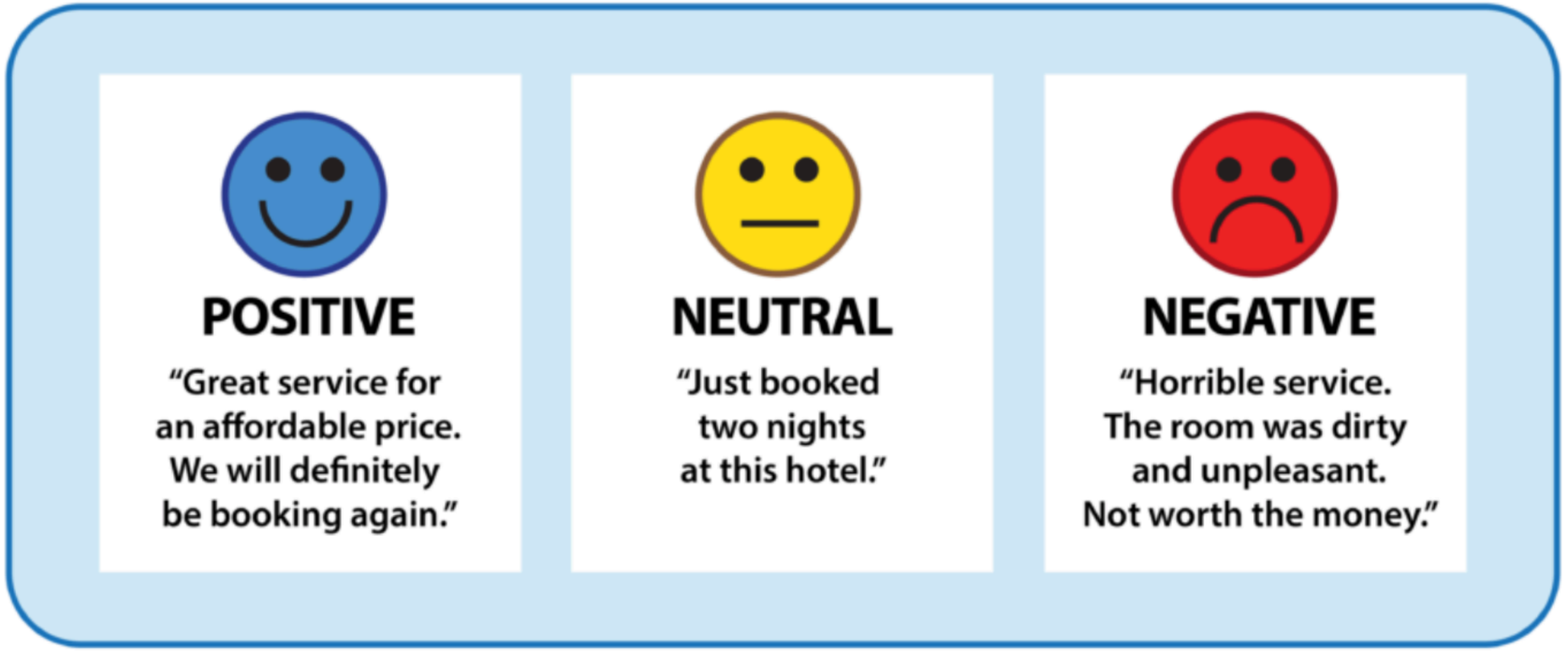


Natural Language Processing

- Virtual assistants

Natural Language Processing

- Sentiment Analysis

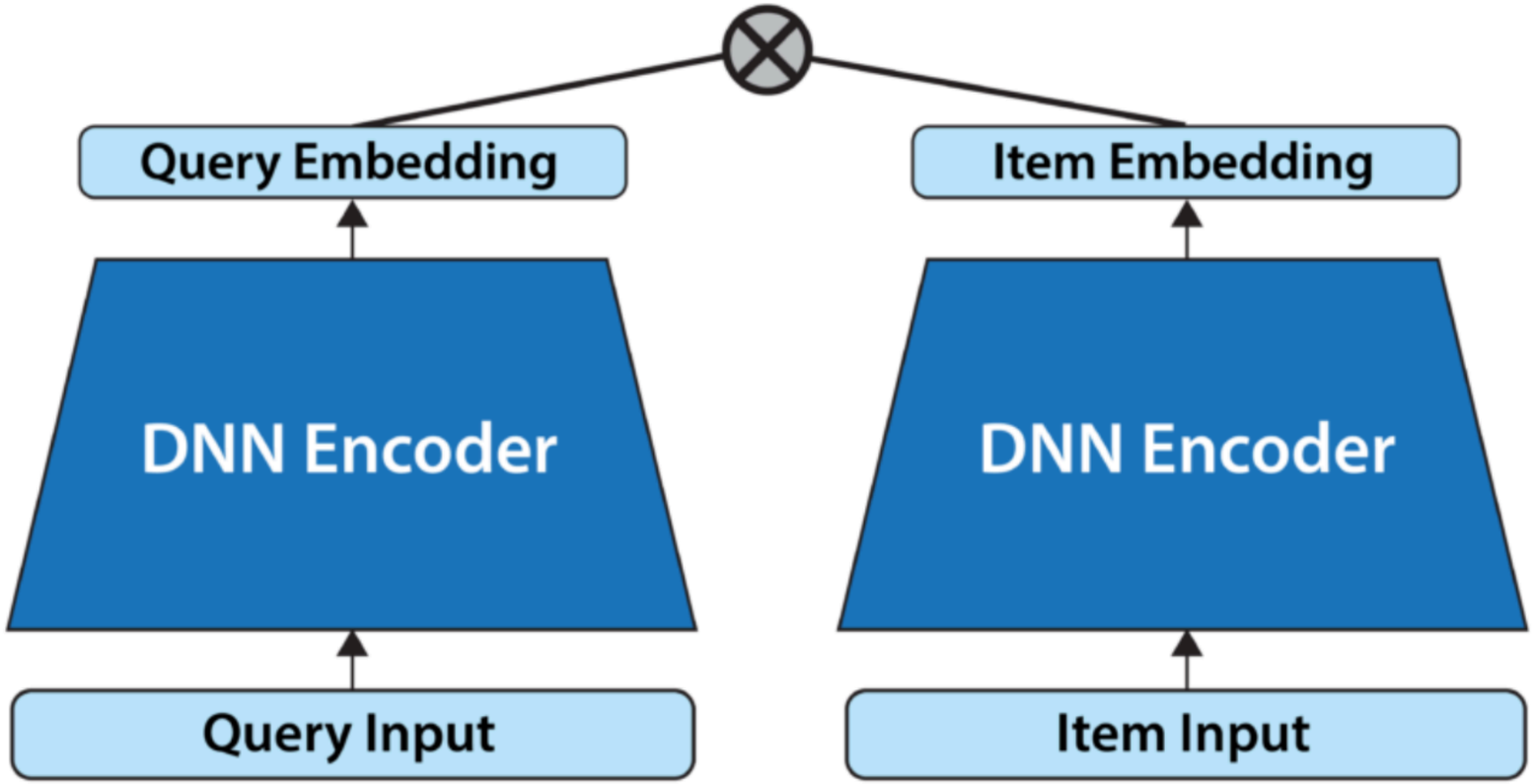


Natural Language Processing

- Named Entity Recognition

Natural Language Processing

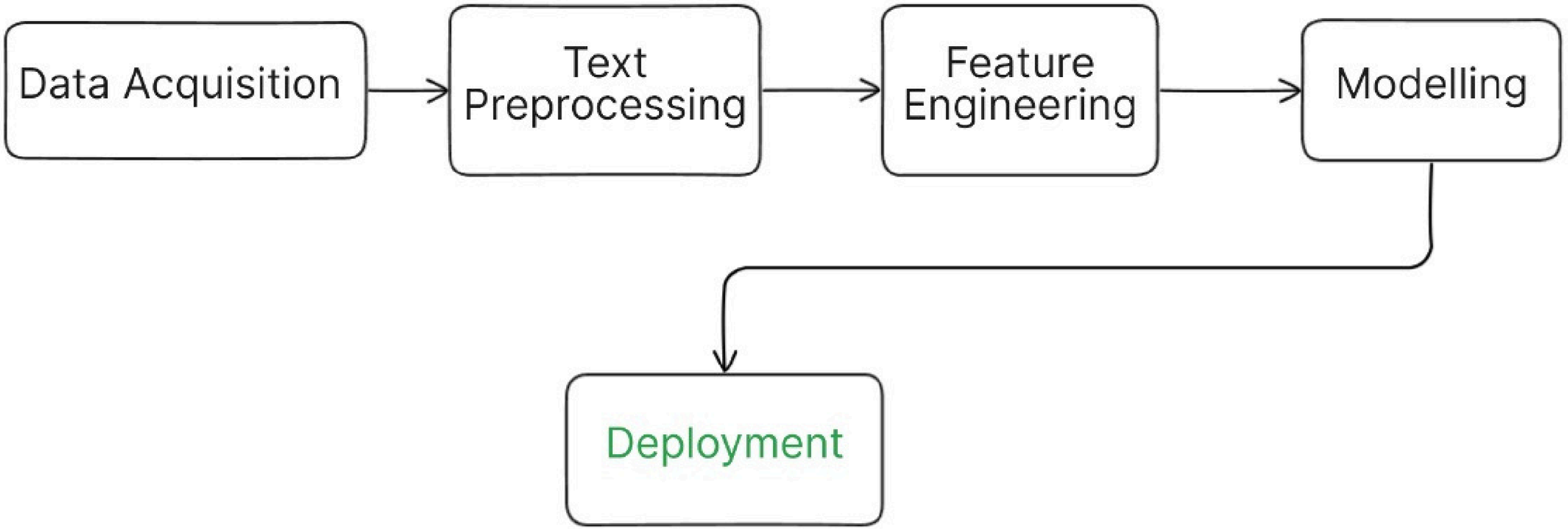
- Information Retrieval



A two-tower network creates a representation of an input query and a group of documents (or items) through two separate networks. Then it compares the representation of the query with that of the documents to find documents that are most relevant to the query.

Natural Language Processing

- NLP Pipeline



Text Preprocessing

- Stop-word Removal

LIST OF ENGLISH STOPWORDS IN NLTK:

their, few, wasn't, has, m, or, did, isn, very, themselves, you've, you'd, do, between, other, t, shan, yourself, does, ours, i, it, should, what, himself, some, itself, there, weren, most, her, mustn, hers, doesn, won, doesn't, hasn, s, y, wouldn't, didn't, him, couldn, after, a, will, ain, than, for, being, which, during, ll, my, isn't, its, any, hadn't, his, then, don, of, shouldn't, out, our, have, such, o, nor, too, re, should've, needn't, same, she's, but, weren't, all, against, down, don't, can, you, under, where, wouldn, only, been, aren't, haven, that, doing, if, up, d, needn, ma, yours, shan't, wasn, because, about, those, he, are, was, at, hasn't, over, until, had, with, you're, below, haven't, mightn, here, own, off, both, whom, while, as, ourselves, they, further, mightn't, these, from, to, them, she, who, were, more, am, why, your, aren, hadn, in, won't, yourselves, no, me, didn, an, so, before, is, on, now, each, how, be, theirs, shouldn, mustn't, above, herself, just, you'll, the, through, again, once, having, by, when, myself, we, it's, this, that'll, couldn't, ve, and, into, not,

Text Preprocessing

- Stemming

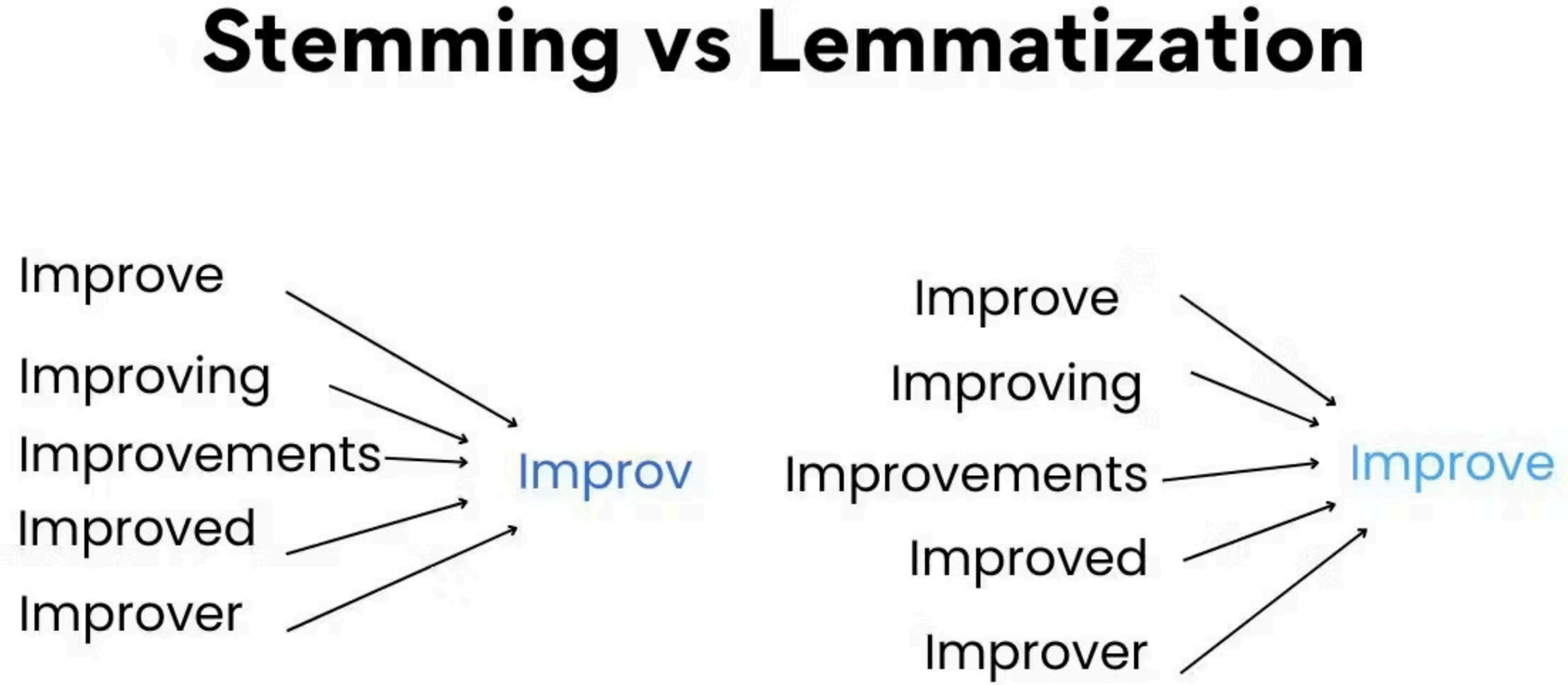
- Stemming is the process of reducing a word to its root form by removing suffixes or prefixes.

	Words	Stemmed words
0	connect	connect
1	connected	connect
2	connection	connect
3	connections	connect
4	connects	connect

	Words	Stemmed words
0	friend	friend
1	friends	friend
2	friended	friend
3	friendly	friendli

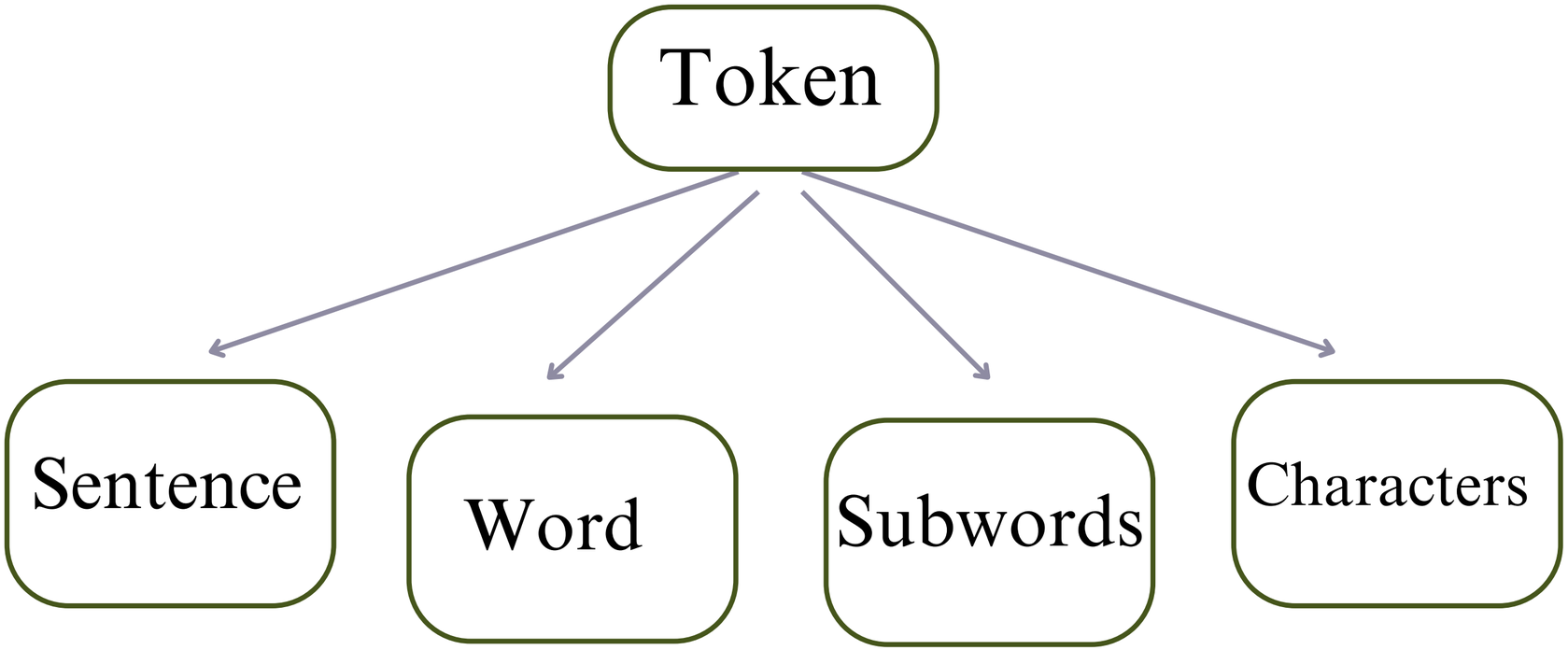
- Lemmatization

  Reduces words to their dictionary base form (lemma) using vocabulary and grammar.



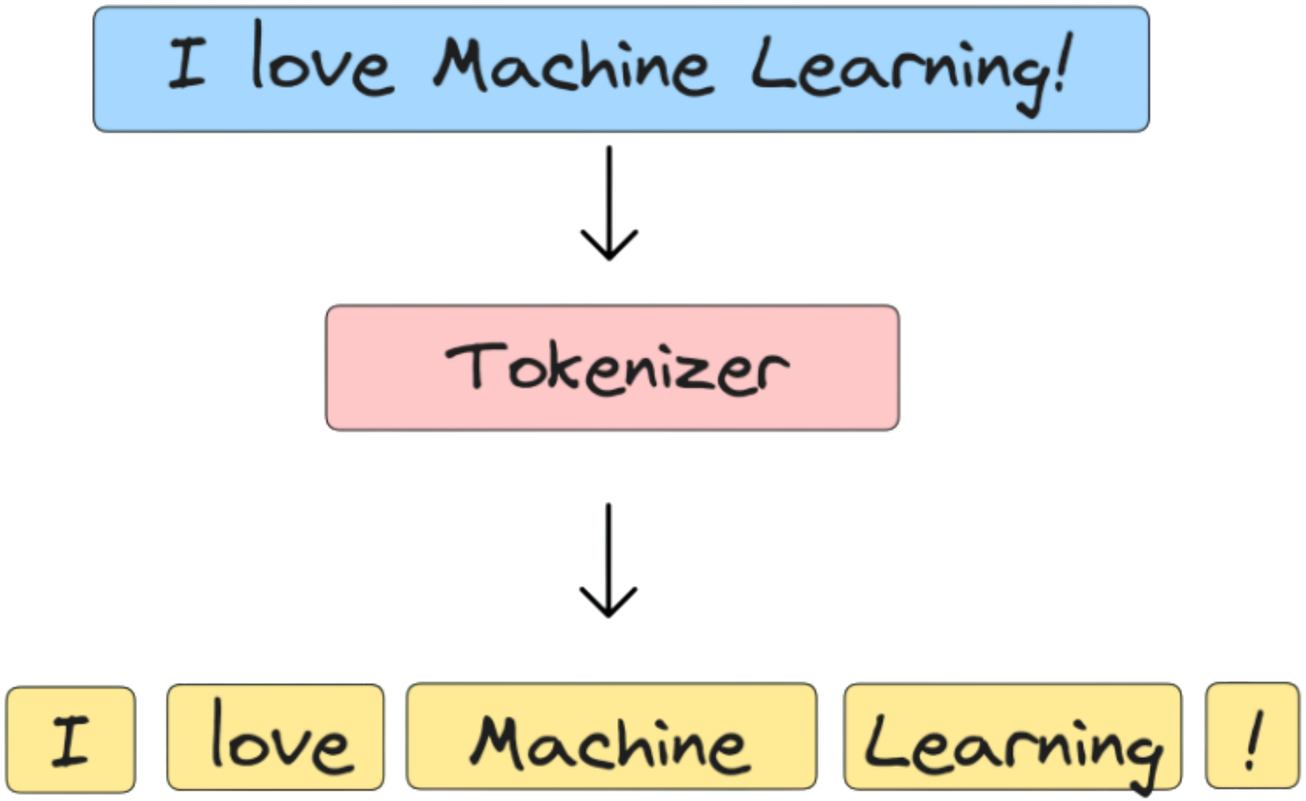
Text Preprocessing

- Tokenization
  - The process of splitting text into smaller units, called **tokens**, which may be:



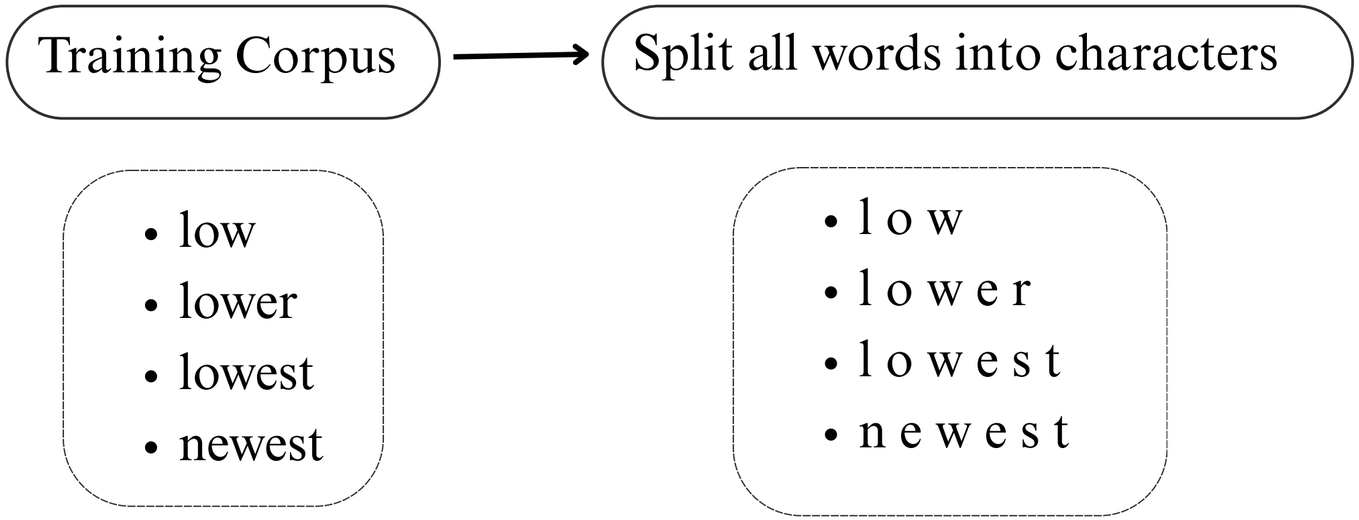
Text Preprocessing

- Tokenization
  - Word Tokenization



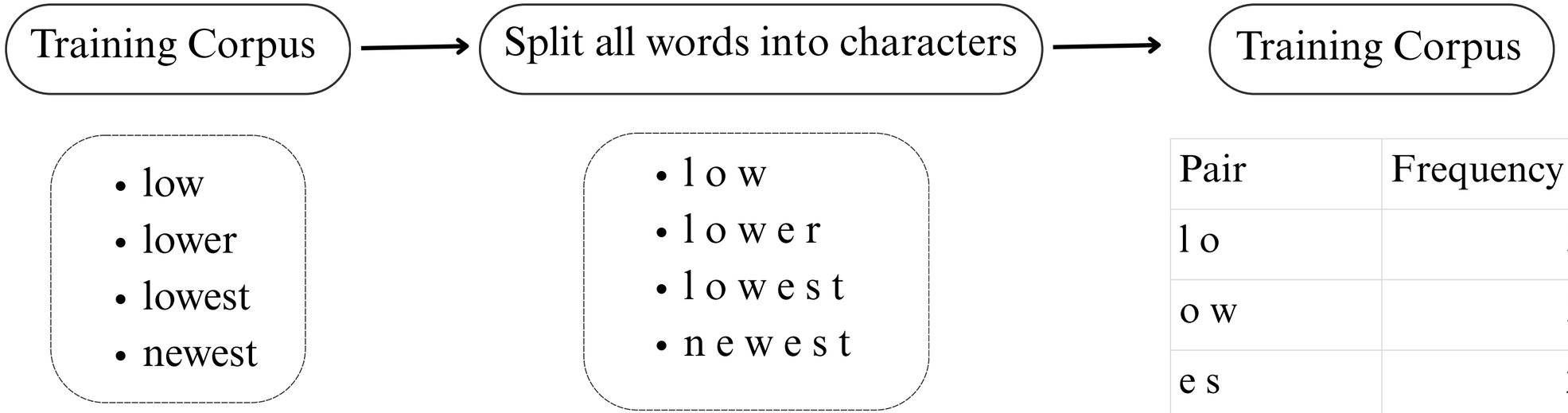
Text Preprocessing

- Tokenization
  - Subword Tokenization



Text Preprocessing

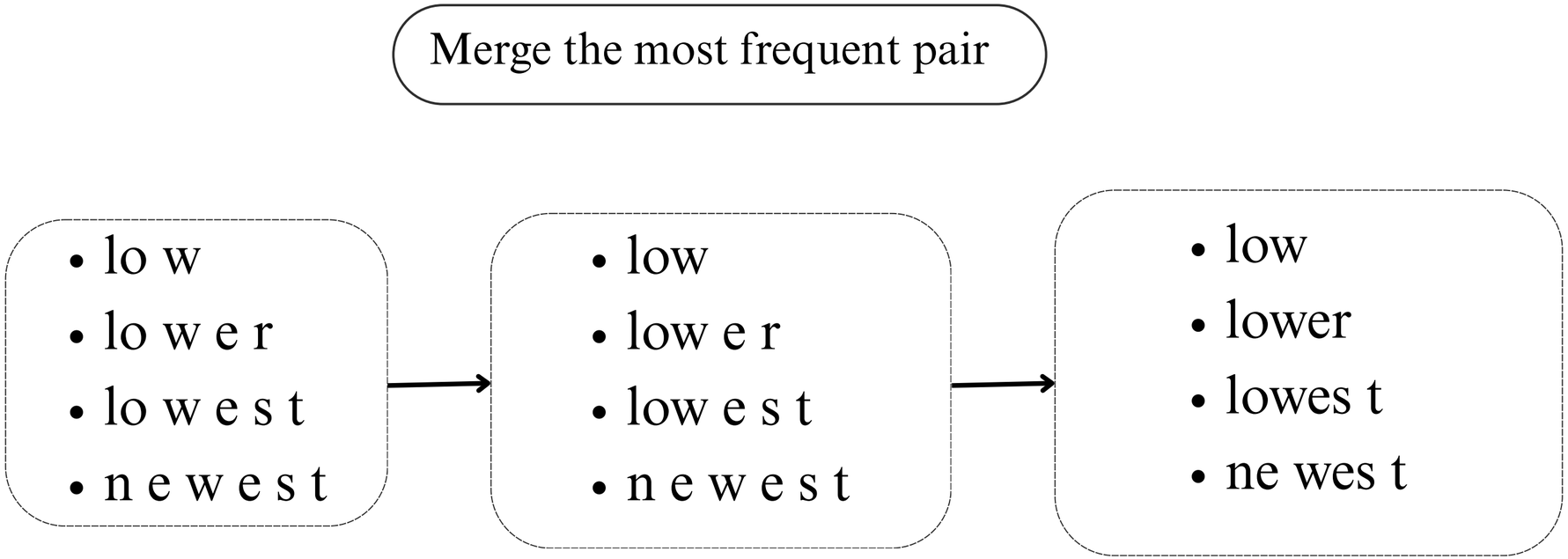
- Tokenization
  - Subword Tokenization



Pair	Frequency
l o	3
o w	3
e s	2
e r	1
e t	1
n e	1

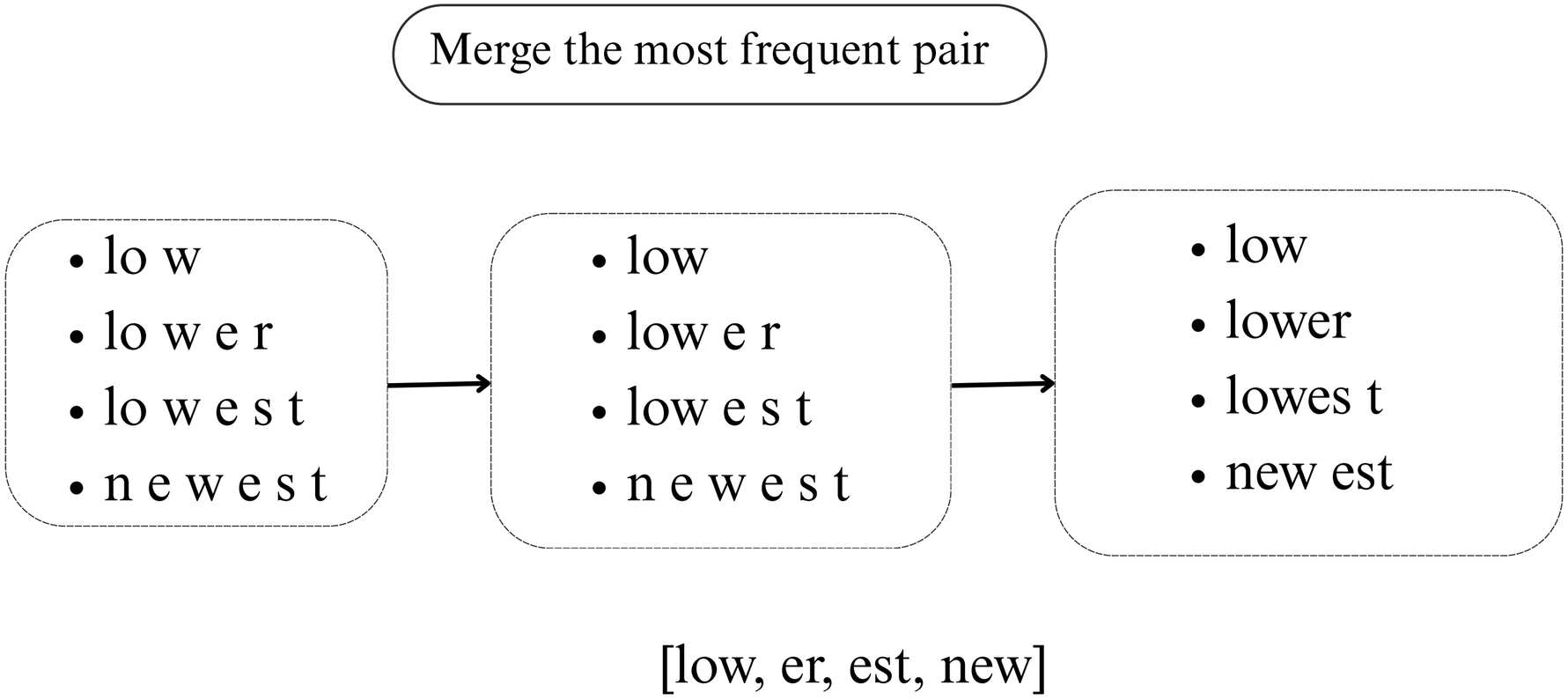
Text Preprocessing

- Tokenization
  - Subword Tokenization



Text Preprocessing

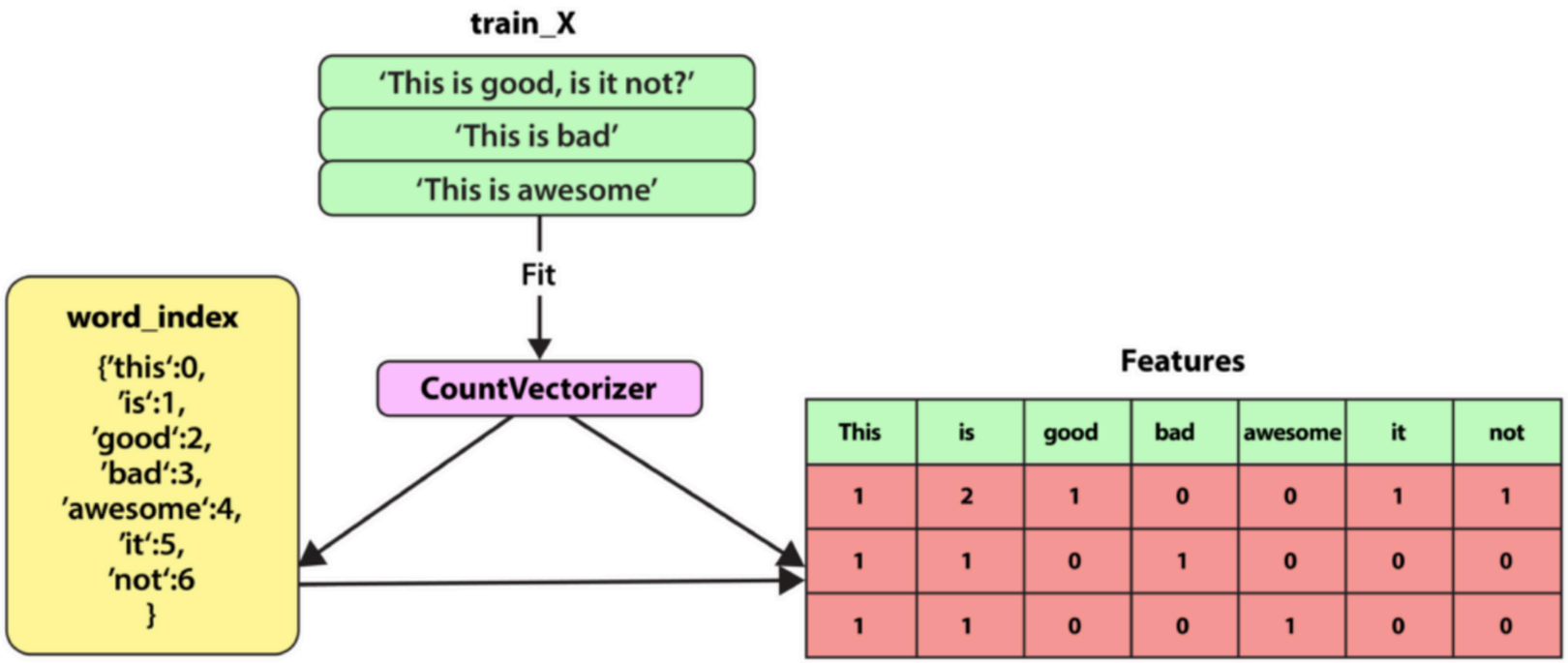
- Tokenization
  - Subword Tokenization



[low, er, est, new]

Text Representation

- Converting text into numerical form so that machine learning models can process it
  - Bag-of-Words



Bag-of-Words (through the CountVectorizer method) encodes the total number of times a document uses each word in the associated corpus.

Text Representation

- Converting text into numerical form so that machine learning models can process it
  - Bag-of-Words

Document	NLP	is	fun	very	powerful
Doc1: NLP is fun	?	?	?	?	?
Doc2: NLP is very fun	?	?	?	?	?
Doc3: NLP is powerful	?	?	?	?	?

Text Representation

- Converting text into numerical form so that machine learning models can process it
    - Bag-of-Words

Document	NLP	is	fun	very	powerful
Doc1: NLP is fun	1	1	1	0	0
Doc2: NLP is very fun	1	1	1	1	0
Doc3: NLP is powerful	1	1	0	0	1

- TF-IDF
  - Term Frequency (TF)
  - Measures how often a word appears in a document

$$TF_{ij} = \frac{\text{number of times term } i \text{ appears in document } j}{\text{total number of terms in document } j}$$

- TF-IDF

  - Inverse Document Frequency (IDF)
  - Measures how rare or common a word is across the corpus

$$IDF_i = \log\left(\frac{\text{total number of documents}}{\text{number of documents containing term } i}\right)$$

Text Representation

- TF-IDF

$$TF_{ij} = \frac{\text{number of times term } i \text{ appears in document } j}{\text{total number of terms in document } j} \qquad IDF_i = \log\left(\frac{\text{total number of documents}}{\text{number of documents containing term } i}\right)$$

$$TF\text{-}IDF_{ij} = TF_{ij} \times IDF_i$$

- High IDF ⇒ word is rare but valuable
- Low IDF ⇒ word is common (e.g., “the”, “a”, “and”)

Text Representation

- TF-IDF

document	words
D_1	Data Base System Concepts
D_2	Introduction to Algorithms
D_3	Computational Geometry: Algorithms and Applications
D_4	Data Structures and Algorithm Analysis on Massive Data Sets
D_5	Computer Organization

	D_1	D_2	D_3	D_4	D_5
w_1	1	0	0	2	0
w_2	1	0	0	0	0
w_3	0	1	1	1	0
w_4	0	0	1	0	1
w_5	0	0	1	0	0
w_6	0	0	0	1	0
w_7	0	0	0	1	0
w_8	0	0	0	0	1

Text Representation

- TF-IDF

document	words
D_1	Data Base System Concepts
D_2	Introduction to Algorithms
D_3	Computational Geometry: Algorithms and Applications
D_4	Data Structures and Algorithm Analysis on Massive Data Sets
D_5	Computer Organization

	D_1	D_2	D_3	D_4	D_5
w_1	1	0	0	2	0
w_2	1	0	0	0	0
w_3	0	1	1	1	0
w_4	0	0	1	0	1
w_5	0	0	1	0	0
w_6	0	0	0	1	0
w_7	0	0	0	1	0
w_8	0	0	0	0	1

w_1	1.32
w_2	2.32
w_3	0.74
w_4	1.32
w_5	2.32
w_6	2.32
w_7	2.32
w_8	2.32

Text Representation

- TF-IDF

document	words
D_1	Data Base System Concepts
D_2	Introduction to Algorithms
D_3	Computational Geometry: Algorithms and Applications
D_4	Data Structures and Algorithm Analysis on Massive Data Sets
D_5	Computer Organization

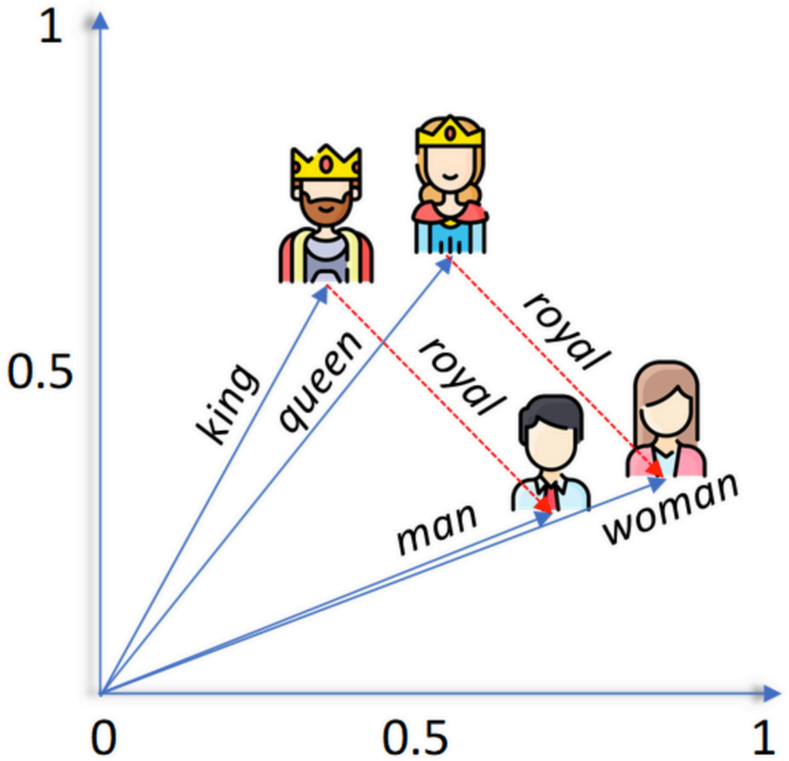
	D_1	D_2	D_3	D_4	D_5
w_1	1	0	0	2	0
w_2	1	0	0	0	0
w_3	0	1	1	1	0
w_4	0	0	1	0	1
w_5	0	0	1	0	0
w_6	0	0	0	1	0
w_7	0	0	0	1	0
w_8	0	0	0	0	1

w_1	1.32
w_2	2.32
w_3	0.74
w_4	1.32
w_5	2.32
w_6	2.32
w_7	2.32
w_8	2.32

p_1	$(1.32, 2.32, 0, 0, 0, 0, 0, 0)$
p_2	$(0, 0, 0.74, 0, 0, 0, 0, 0)$
p_3	$(0, 0, 0.74, 1.32, 2.32, 0, 0, 0)$
p_4	$(2.09, 0, 0.74, 0, 0, 2.32, 2.32, 0)$
p_5	$(0, 0, 0, 1.32, 0, 0, 0, 2.32)$

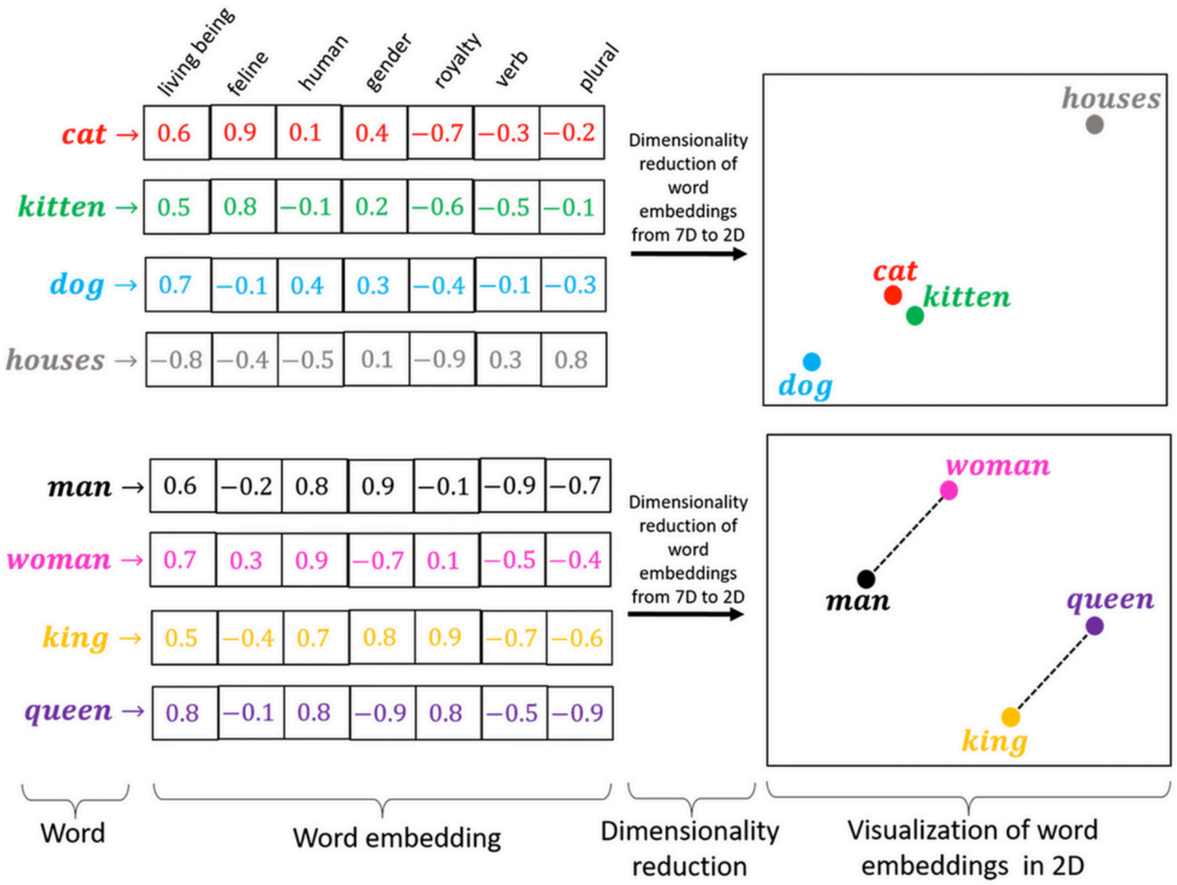
Text Representation

- Vector Space Model
  - VSM Represents words or documents as vectors in an n-dimensional space.
    - Based on the distributional hypothesis: Words that occur in similar contexts have similar meanings.



Text Representation

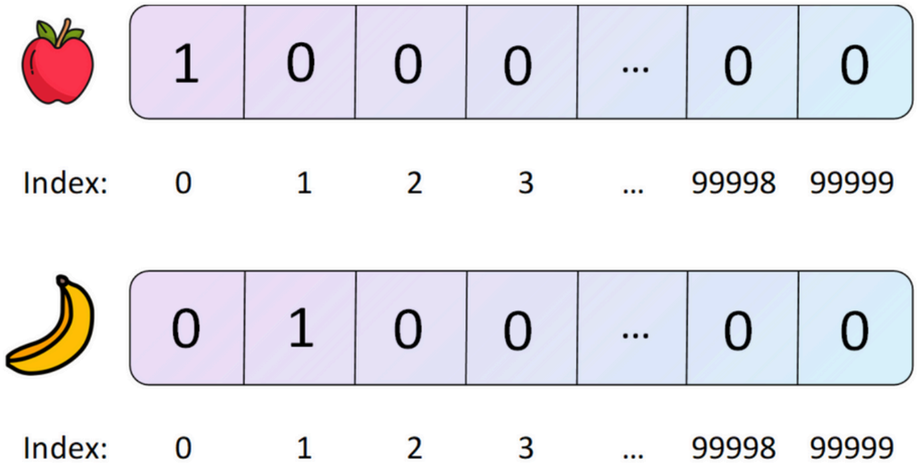
- Word2Vec
  - Represents words as vectors such that words appearing in similar contexts have similar meanings.
  - Similar words have similar vector representations



Text Representation

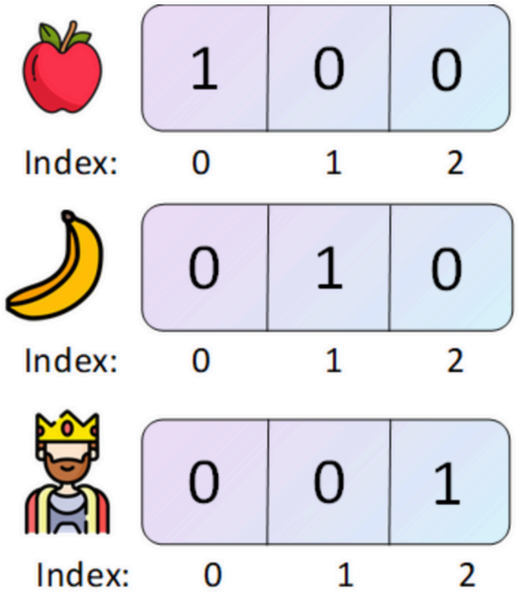
- Word2Vec

- Sparse Representation



- No semantic meaning
- Very high-dimensional and sparse

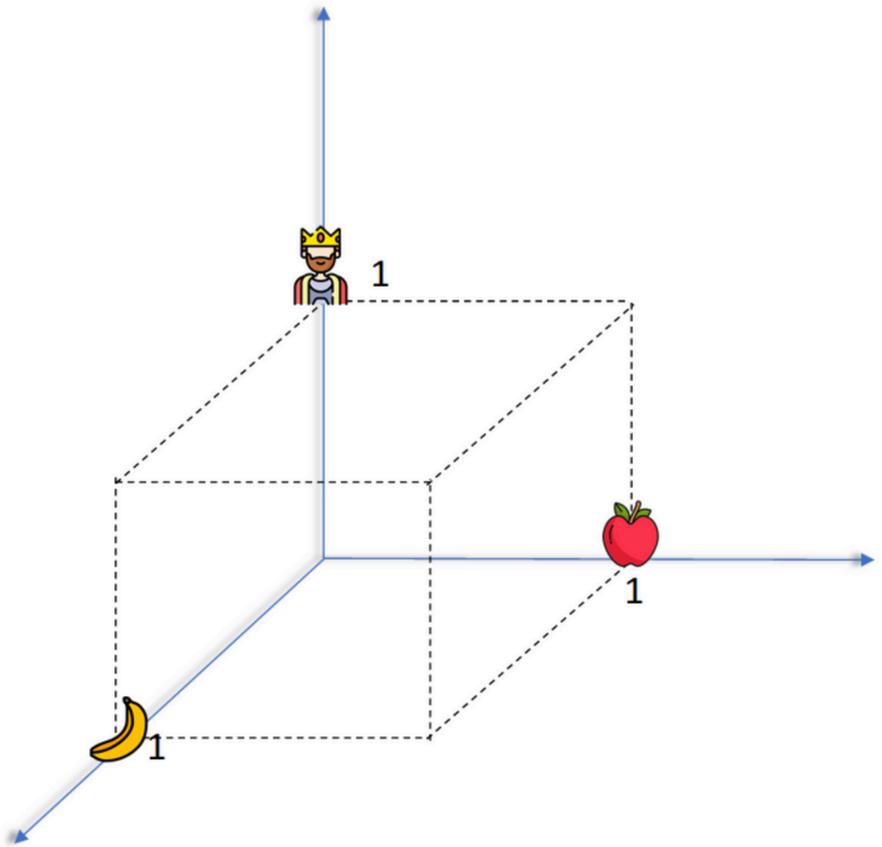
- Dense Representation



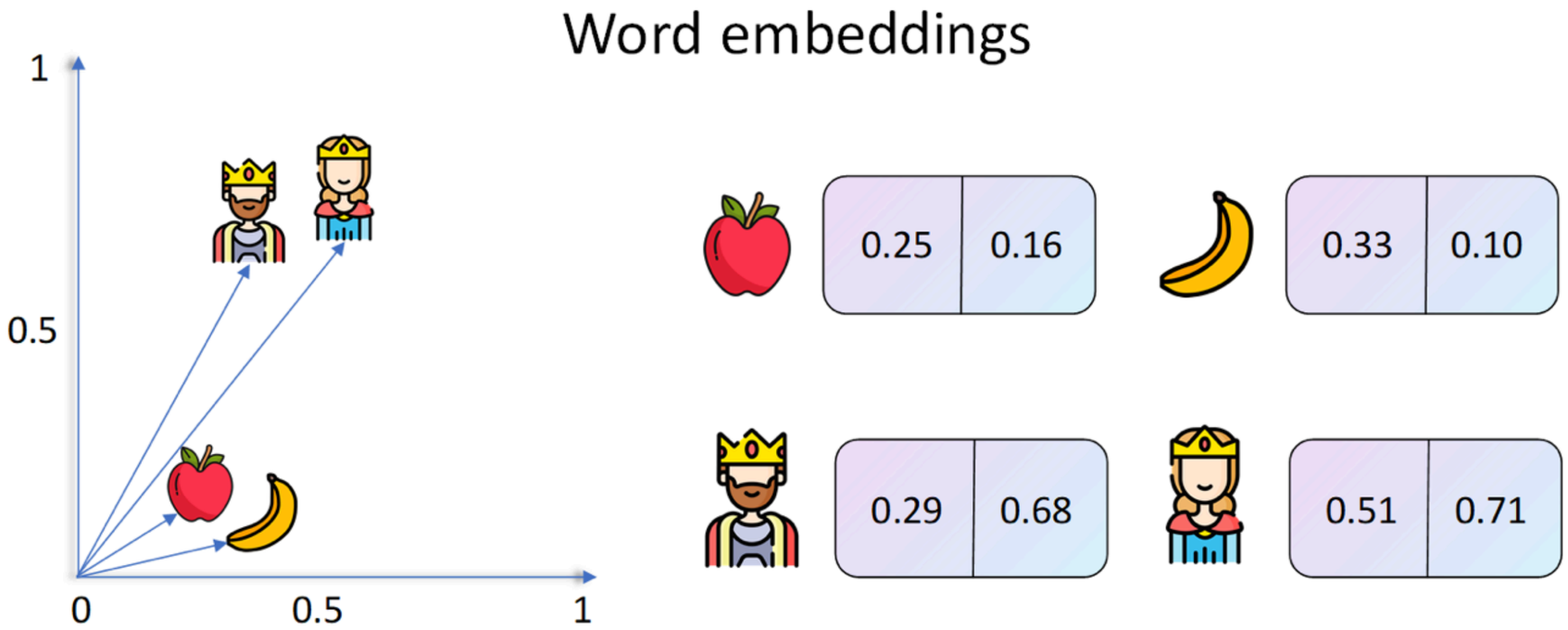
- Captures semantic meaning

Text Representation

- Word2Vec
  - If we plot these vectors in a 3D coordinate system, each axis corresponds to one dimension of the vocabulary



- Word2Vec
  - Similar words have similar vectors, leading to meaningful distances in embedding space.



Text Representation

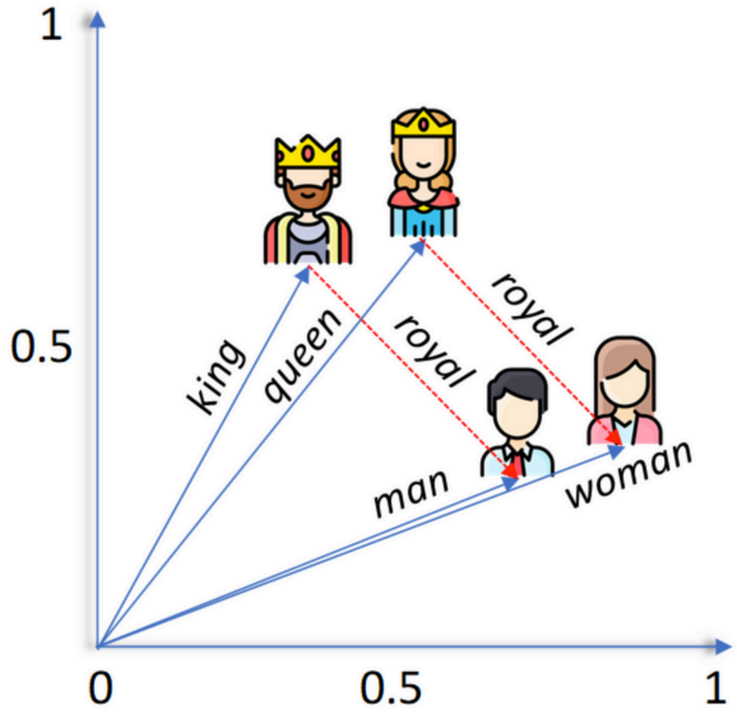
- Word2Vec
  - Word embeddings preserve relationships, not just individual word meanings.

$$\vec{man} = \begin{bmatrix} 0.29 \\ 0.20 \end{bmatrix} \quad \vec{woman} = \begin{bmatrix} 0.51 \\ 0.23 \end{bmatrix}$$

$$\vec{king} = \begin{bmatrix} 0.29 \\ 0.68 \end{bmatrix} \quad \vec{queen} = \begin{bmatrix} 0.51 \\ 0.71 \end{bmatrix}$$

$$\vec{king} - \vec{man} = \begin{bmatrix} 0.29 \\ 0.68 \end{bmatrix} - \begin{bmatrix} 0.29 \\ 0.20 \end{bmatrix} = \begin{bmatrix} 0.00 \\ 0.48 \end{bmatrix}$$

$$\vec{queen} - \vec{woman} = \begin{bmatrix} 0.51 \\ 0.71 \end{bmatrix} - \begin{bmatrix} 0.51 \\ 0.23 \end{bmatrix} = \begin{bmatrix} 0.00 \\ 0.48 \end{bmatrix}$$

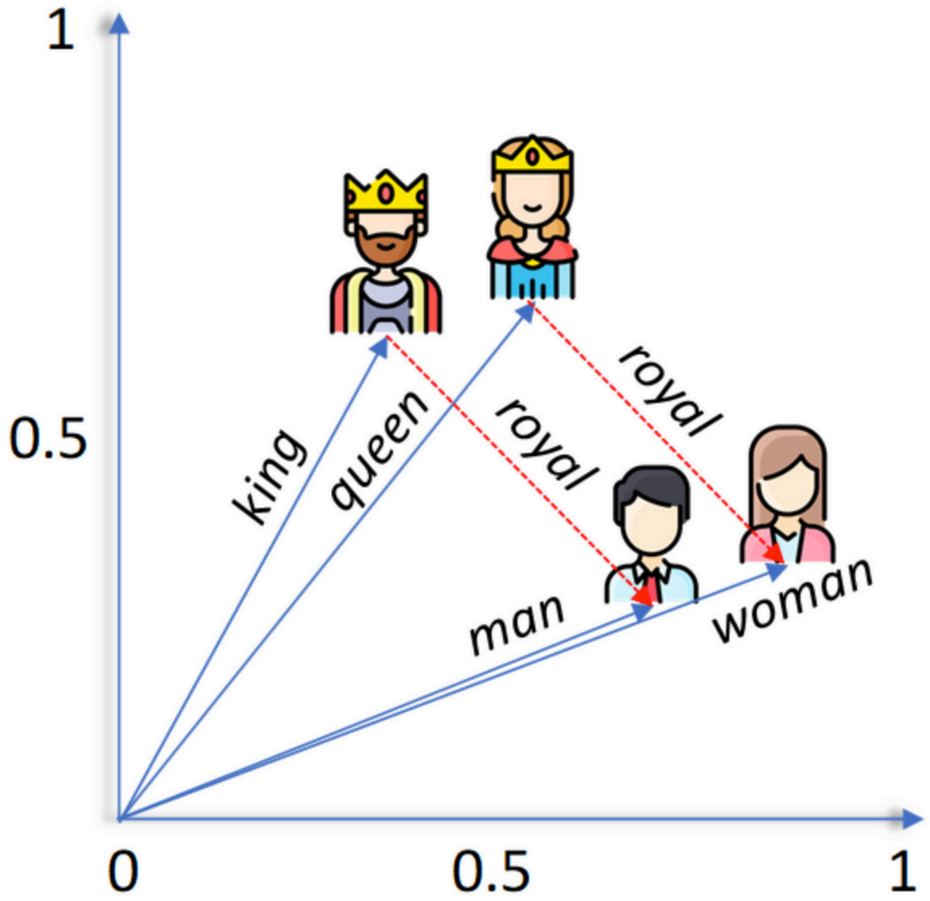


Text Representation

- Word2Vec
  - Word embeddings preserve relationships, not just individual word meanings.

$$\vec{king} - \vec{man} + \vec{woman} = \vec{queen}$$

$$\vec{king} - \vec{man} = \vec{queen} - \vec{woman}$$



- Word2Vec
  - Word2Vec learns embeddings using either CBOW or Skip-Gram training objectives.

Continious bag of words: *"I love drinking apple smoothies"*

Skip-gram: "I love drinking apple smoothies"

- CBOW

"I love drinking apple smoothies"

$$\vec{I} = \begin{bmatrix} 0.10 \\ 0.20 \end{bmatrix} \quad \vec{love} = \begin{bmatrix} 0.30 \\ 0.40 \end{bmatrix} \quad \vec{drinking} = \begin{bmatrix} 0.20 \\ 0.50 \end{bmatrix} \quad \vec{smoothies} = \begin{bmatrix} 0.40 \\ 0.60 \end{bmatrix}$$

$$\vec{apple} = \begin{bmatrix} 0.60 \\ 0.30 \end{bmatrix}$$

- CBOW

"I love drinking apple smoothies"

$$(\vec{I}, \vec{love}, \vec{drinking}, \vec{smoothies}) \rightarrow \vec{apple}$$

$$\vec{c} = \frac{\begin{bmatrix}0.10\\0.20\end{bmatrix} + \begin{bmatrix}0.30\\0.40\end{bmatrix} + \begin{bmatrix}0.20\\0.50\end{bmatrix} + \begin{bmatrix}0.40\\0.60\end{bmatrix}}{4} = \begin{bmatrix}0.25\\0.425\end{bmatrix}$$

- Skip-gram

$$\vec{apple} \rightarrow \{\vec{I}, \vec{love}, \vec{drinking}, \vec{smoothies}\}$$

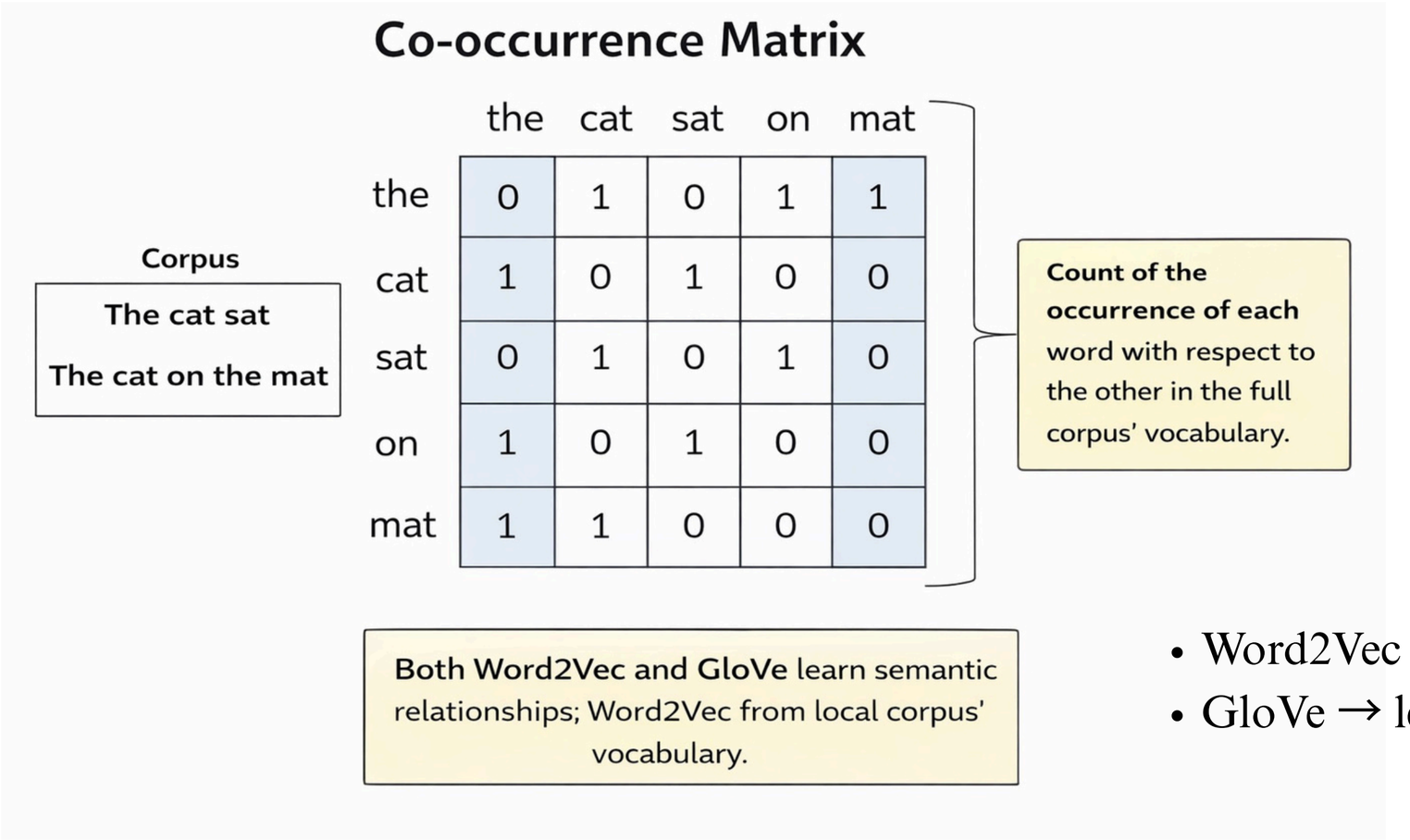
$$\text{Score}(\vec{apple}, \vec{drinking}) = \vec{apple} \cdot \vec{drinking} = (0.60)(0.20) + (0.30)(0.50) = 0.27$$

$$\vec{apple} \rightarrow \vec{I} \quad \vec{apple} \rightarrow \vec{love} \quad \vec{apple} \rightarrow \vec{drinking} \quad \vec{apple} \rightarrow \vec{smoothies}$$

$$\text{CBOW: } \frac{1}{N}\sum_{i=1}^{N} \vec{w}_i \rightarrow \vec{w}_{\text{target}} \qquad \text{Skip-gram: } \vec{w}_{\text{center}} \rightarrow \vec{w}_{\text{context}}$$

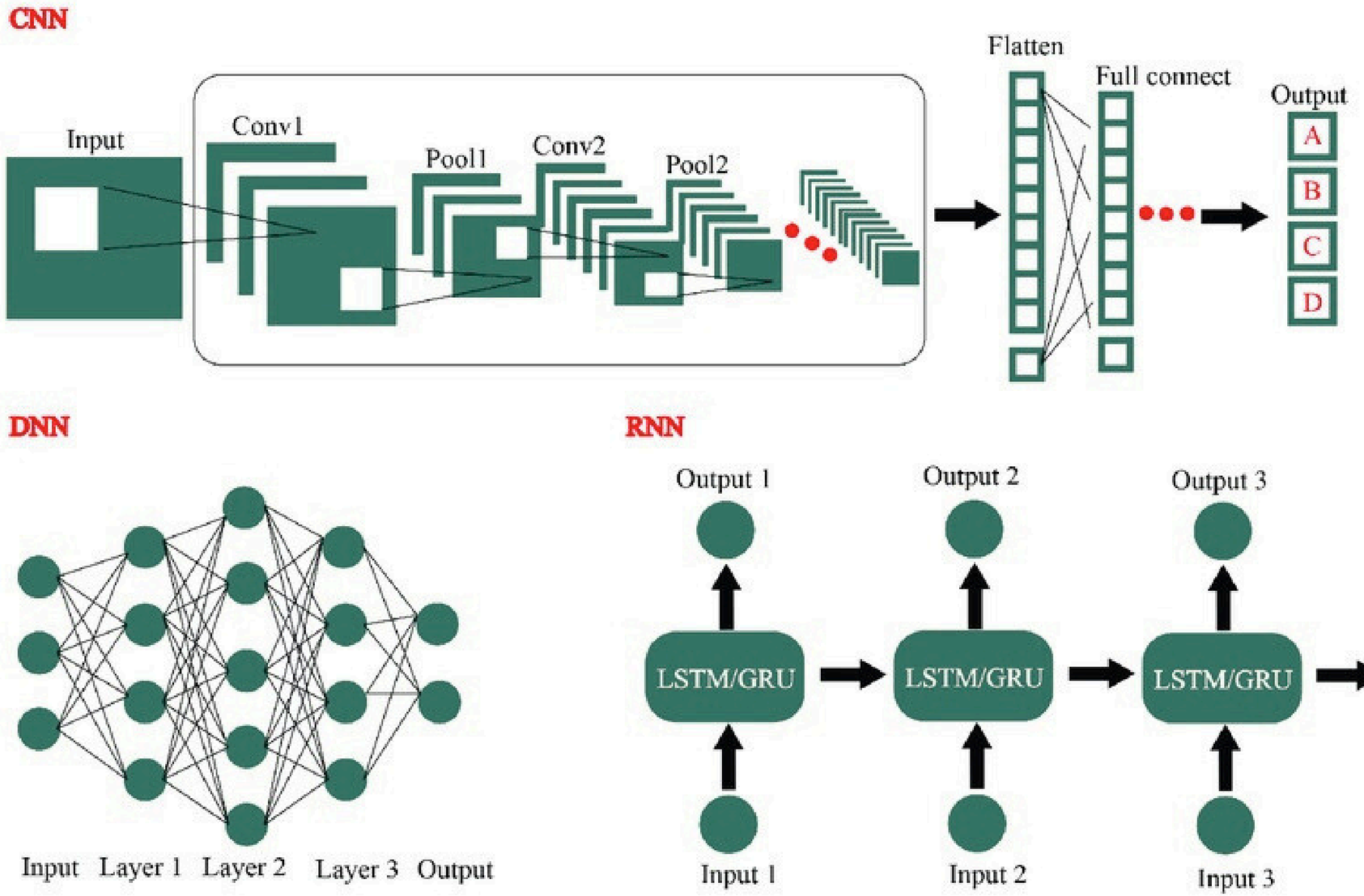
Text Representation

- Global Vectors
  - GloVe is a word embedding method that learns dense vector representations by combining global word co-occurrence statistics with vector space learning.



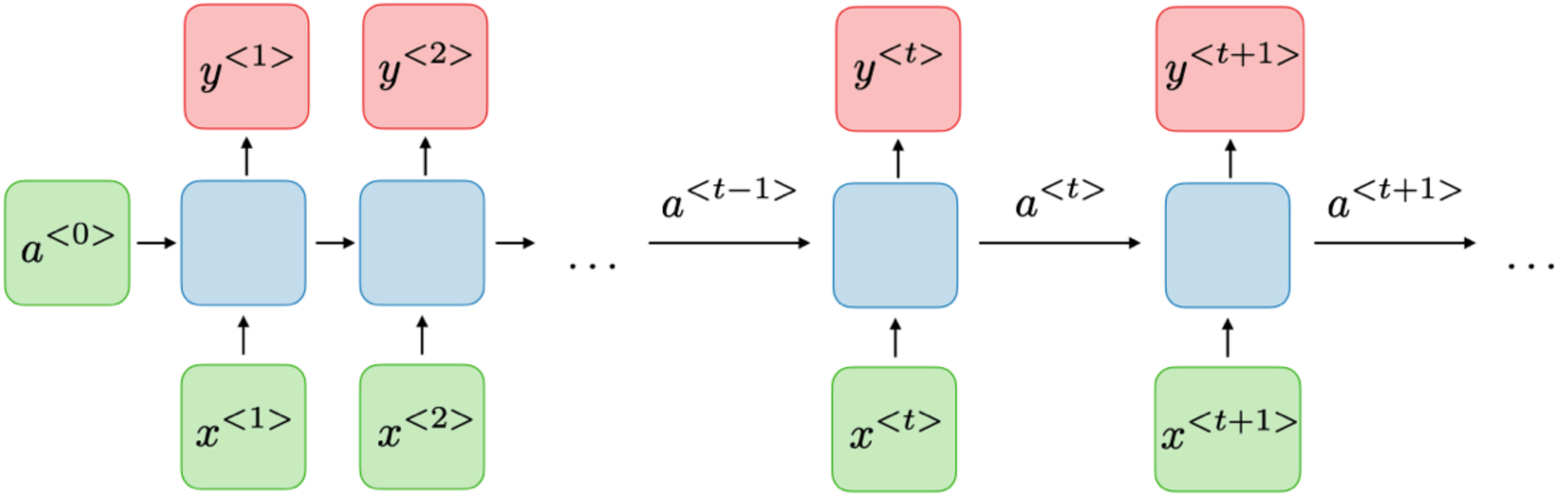
- Word2Vec → learns from local prediction tasks
- GloVe → learns from global co-occurrence statistics

Recurrent neural network



Recurrent neural network

- RNN Basic Structure



$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{and} \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally and g_1, g_2 activation functions.

Recurrent neural network

- RNN Basic Structure

"I love NLP" $x^{\langle 1 \rangle} = 0.5 \quad x^{\langle 2 \rangle} = 1.0 \quad x^{\langle 3 \rangle} = 0.8$

$$a^{\langle 0 \rangle} = 0 \qquad W_{aa} = 0.4, \quad W_{ax} = 0.6, \quad b_a = 0.1$$

$$g_1 = \tanh$$

$$a^{\langle 1 \rangle} = \tanh\left(W_{aa} a^{\langle 0 \rangle} + W_{ax} x^{\langle 1 \rangle} + b_a\right)$$

$$a^{\langle 1 \rangle} = \tanh(0.4 \cdot 0 + 0.6 \cdot 0.5 + 0.1) = \tanh(0.4) \approx 0.38$$

Recurrent neural network

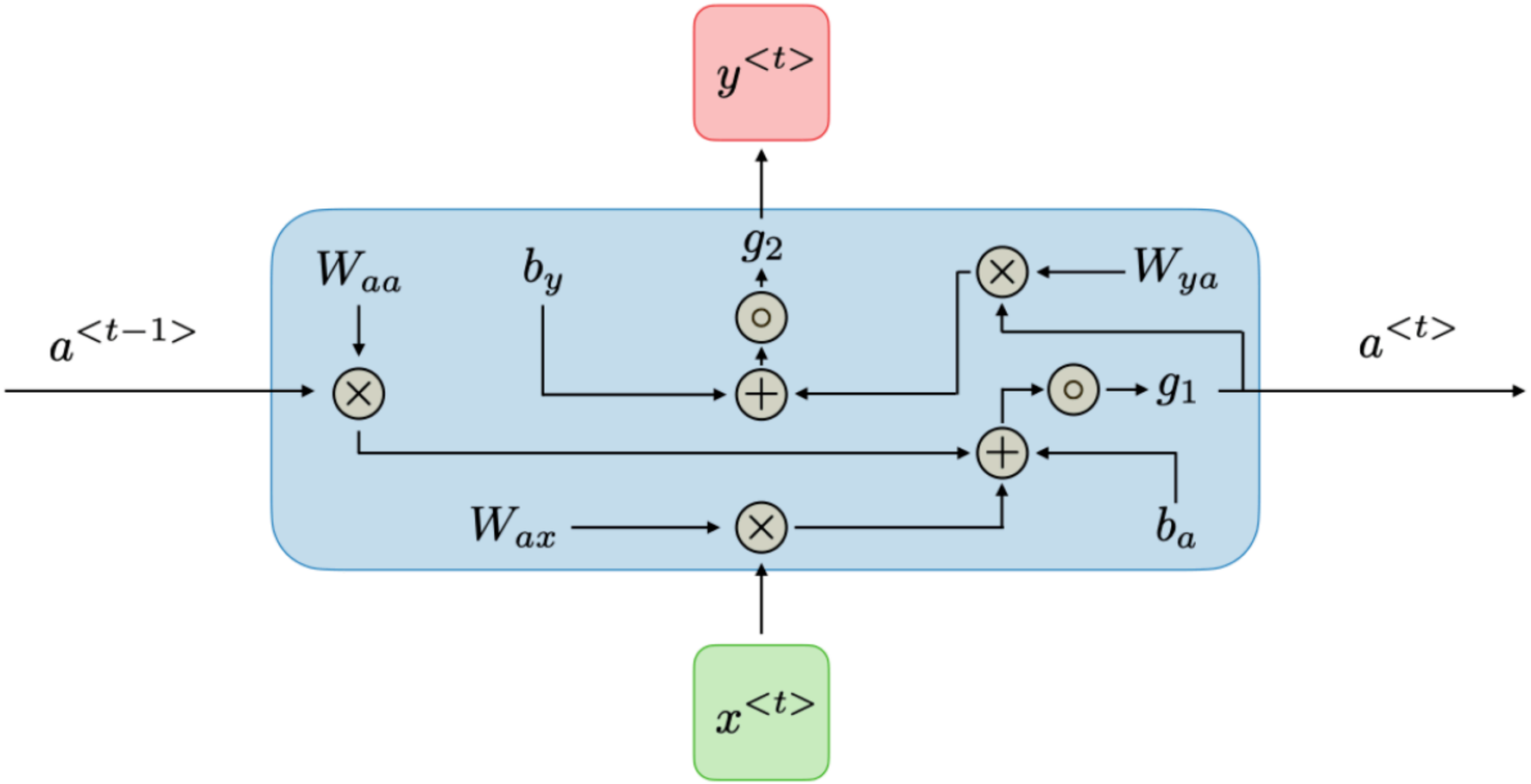
- RNN Basic Structure

$$a^{\langle 2 \rangle} = \tanh(0.4 \cdot 0.38 + 0.6 \cdot 1.0 + 0.1) \ = \tanh(0.852) \approx 0.69$$

$$a^{\langle 3 \rangle} = ?$$

Recurrent neural network

- RNN Basic Structure

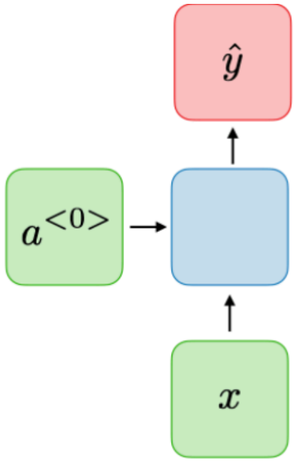
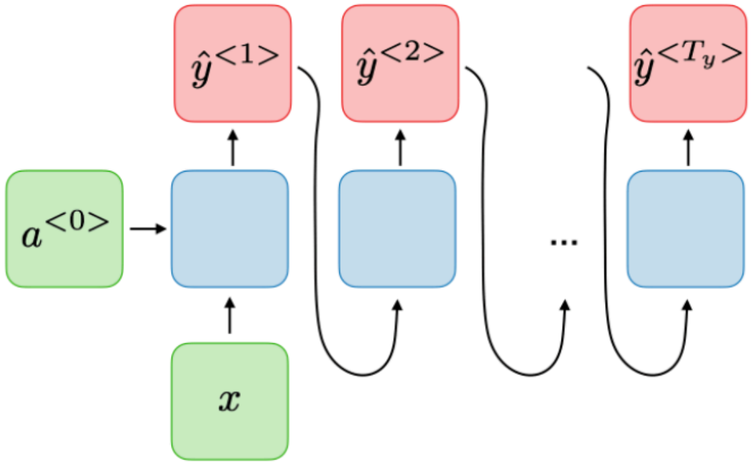


Recurrent neural network

- RNN Basic Structure

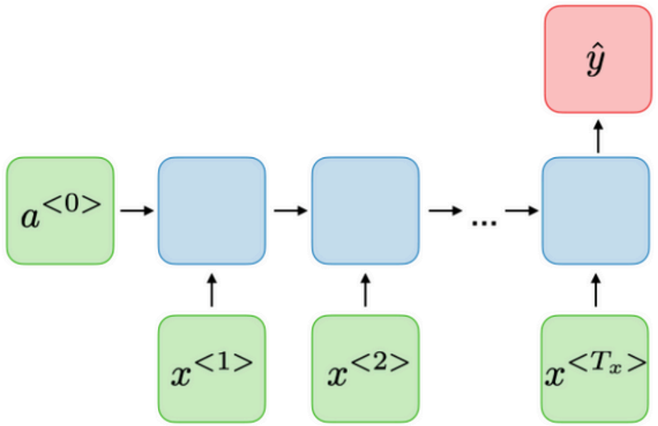
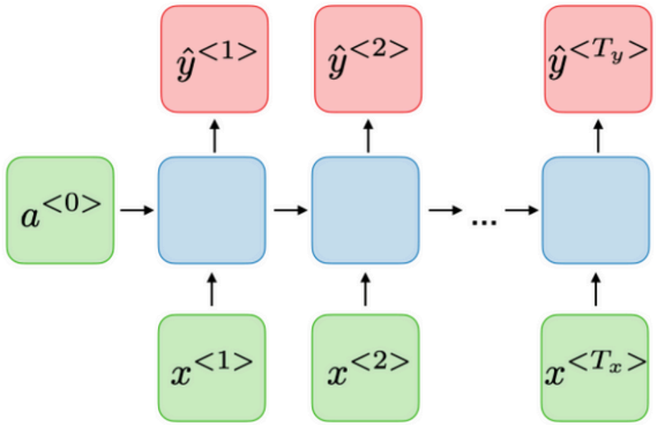
Advantages	Drawbacks
• Possibility of processing input of any length • Model size not increasing with size of input • Computation takes into account historical information • Weights are shared across time	• Computation being slow • Difficulty of accessing information from a long time ago • Cannot consider any future input for the current state

Recurrent neural network

One-to-one $T_x = T_y = 1$		Traditional neural network
One-to-many $T_x = 1, T_y > 1$		Music generation

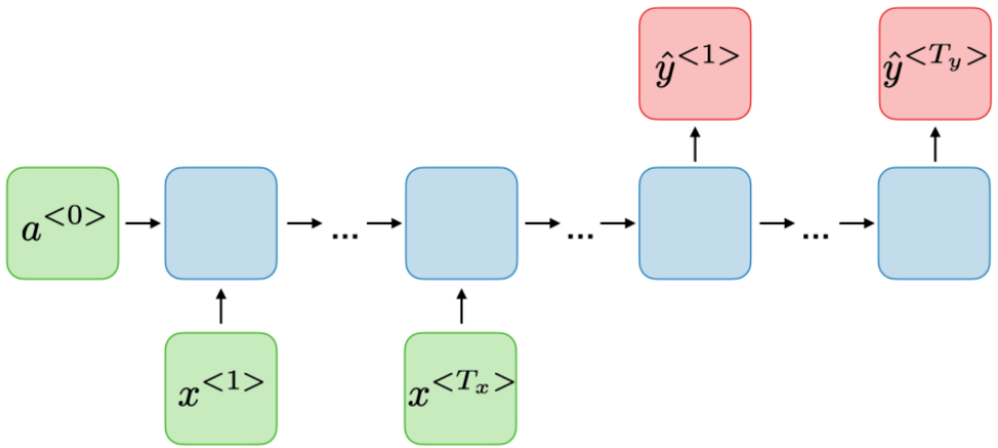
Recurrent neural network

- RNN Types

Many-to-one $T_x > 1, T_y = 1$		Sentiment classification
Many-to-many $T_x = T_y$		Name entity recognition

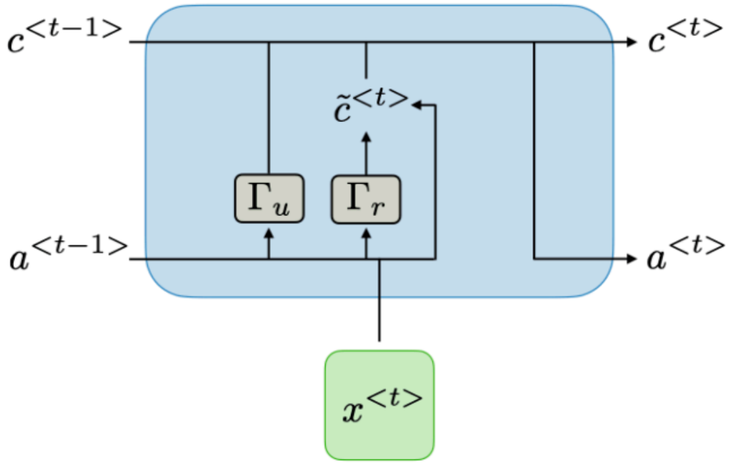
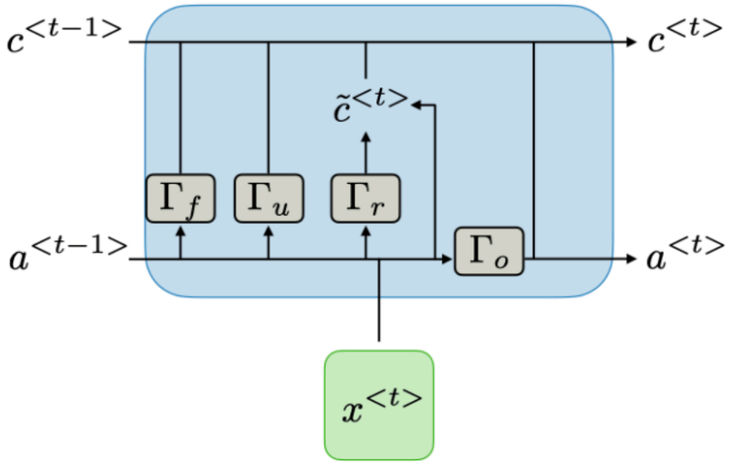
Recurrent neural network

- RNN Types

Many-to-many $T_x \neq T_y$		Machine translation

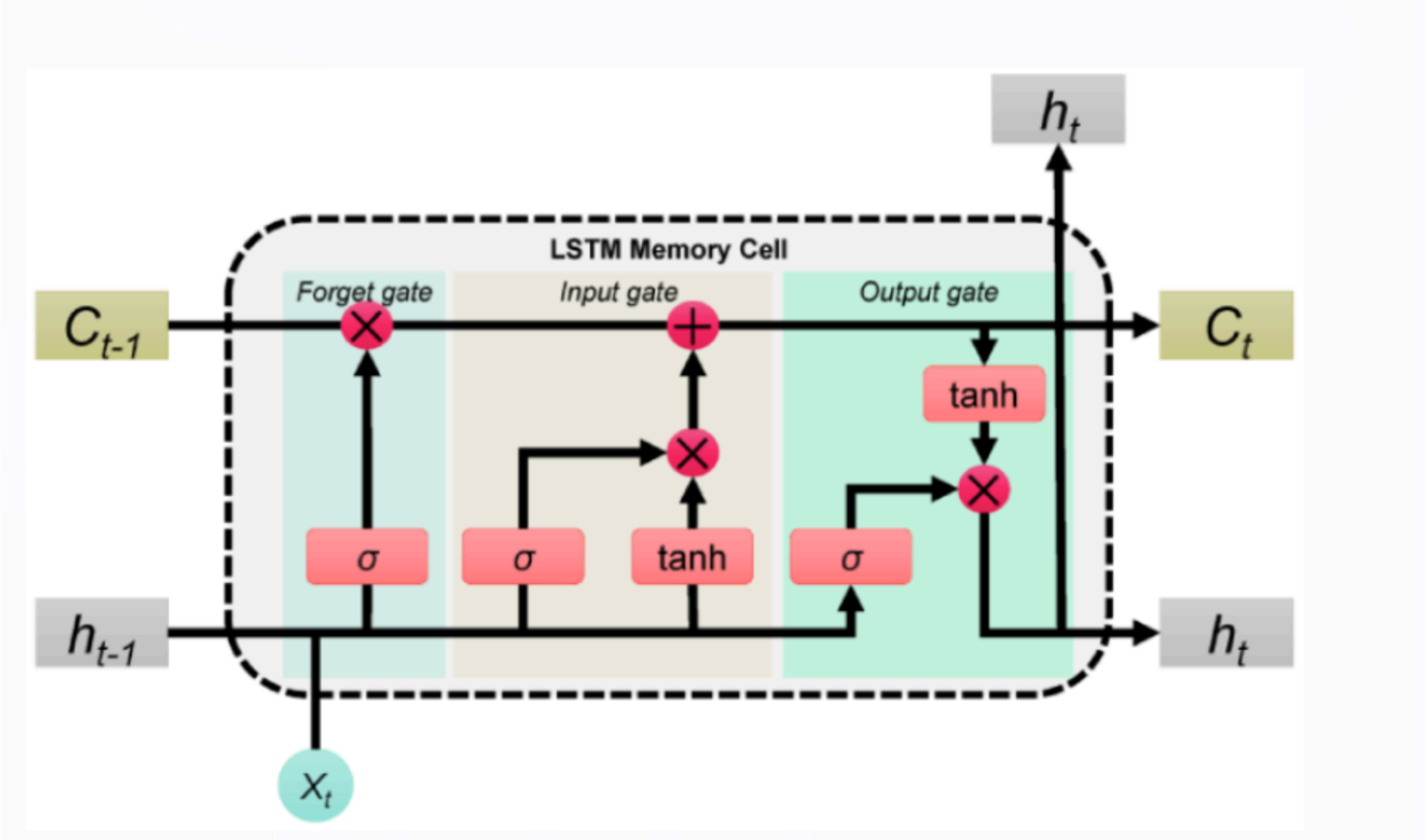
Recurrent neural network

- GRU/LSTM

Characterization	Gated Recurrent Unit (GRU)	Long Short-Term Memory (LSTM)
$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + (1 - \Gamma_u) \star c^{<t-1>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$
$a^{<t>}$	$c^{<t>}$	$\Gamma_o \star c^{<t>}$
Dependencies		

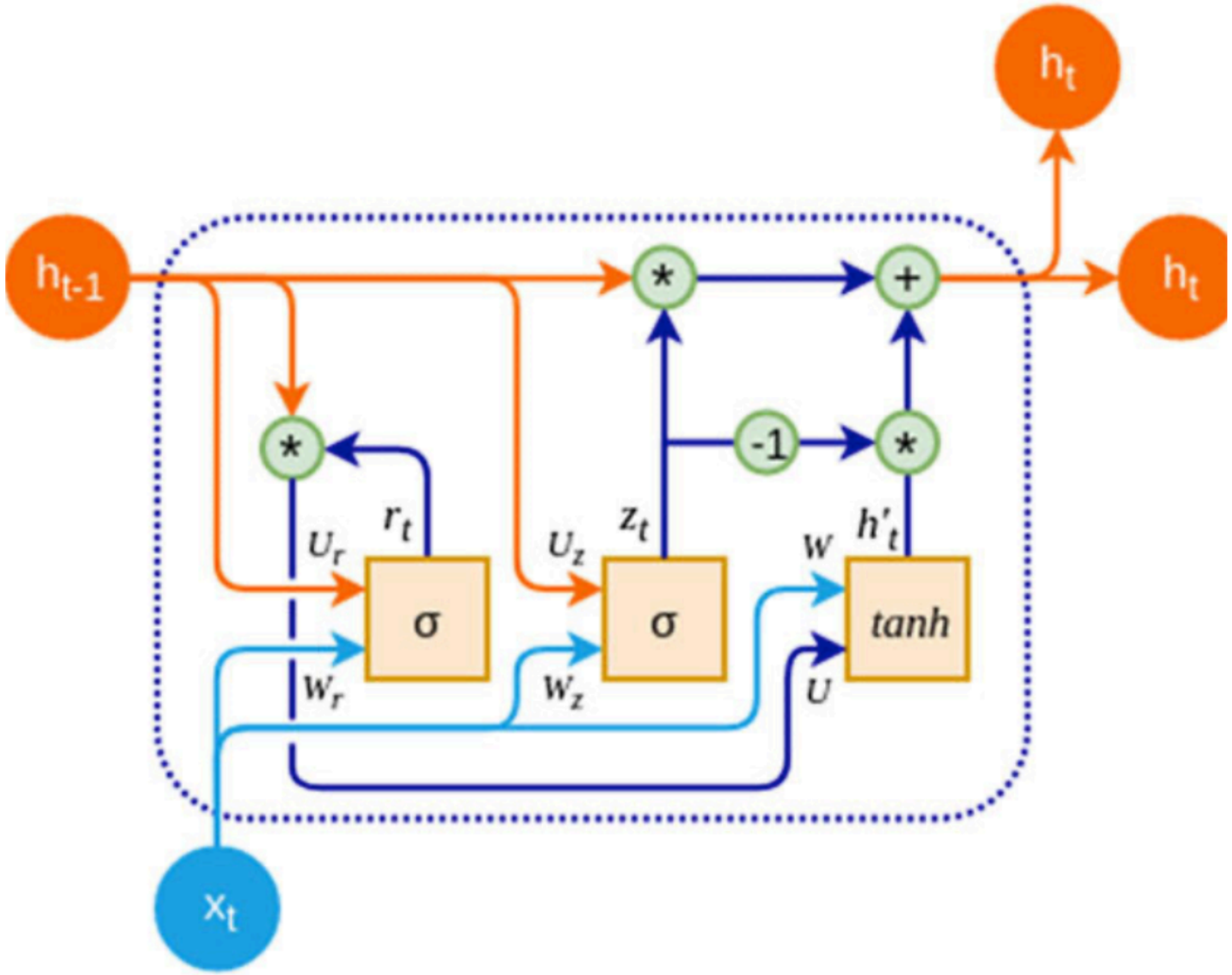
Recurrent neural network

- LSTM



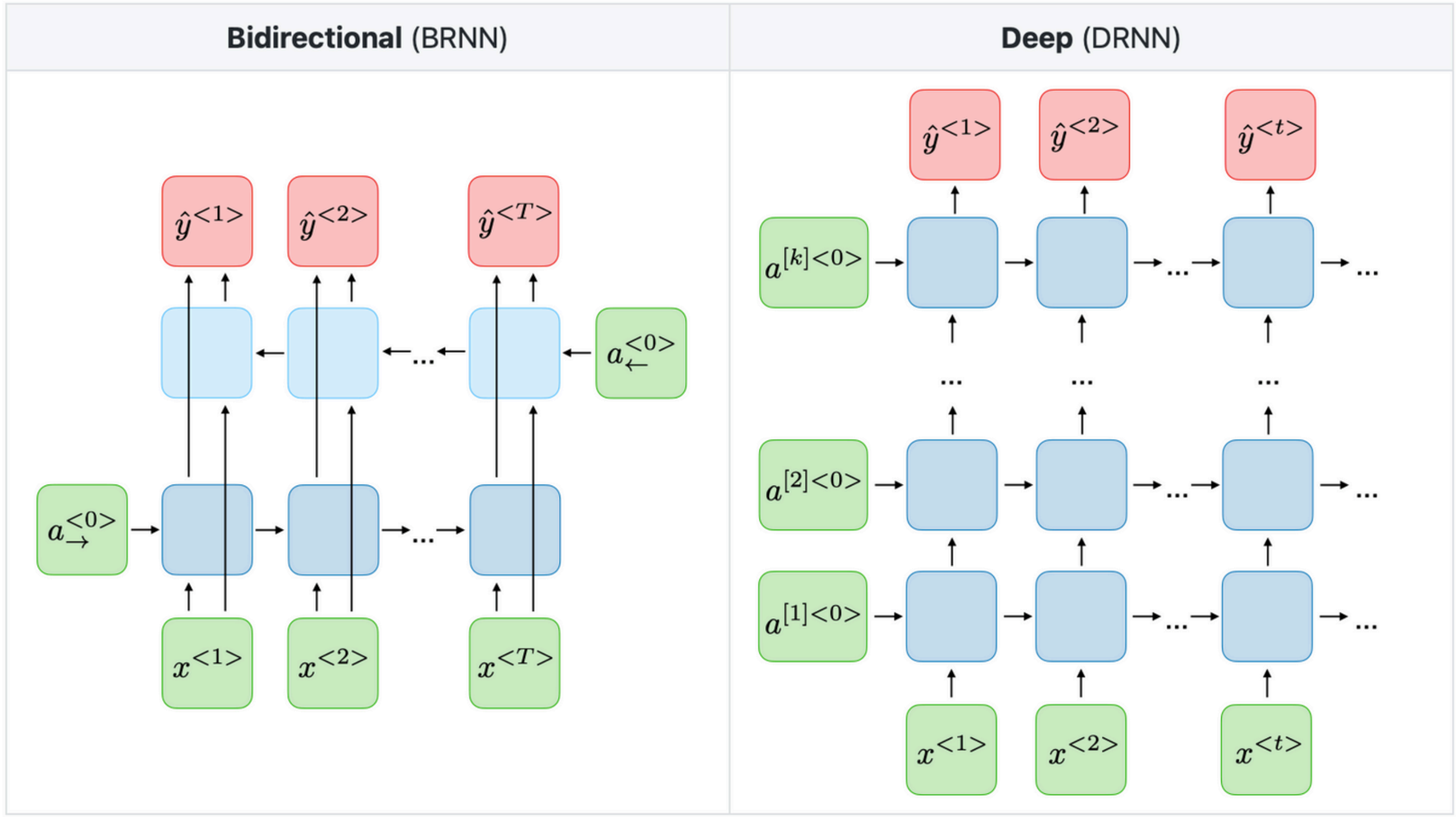
Recurrent neural network

- GRU



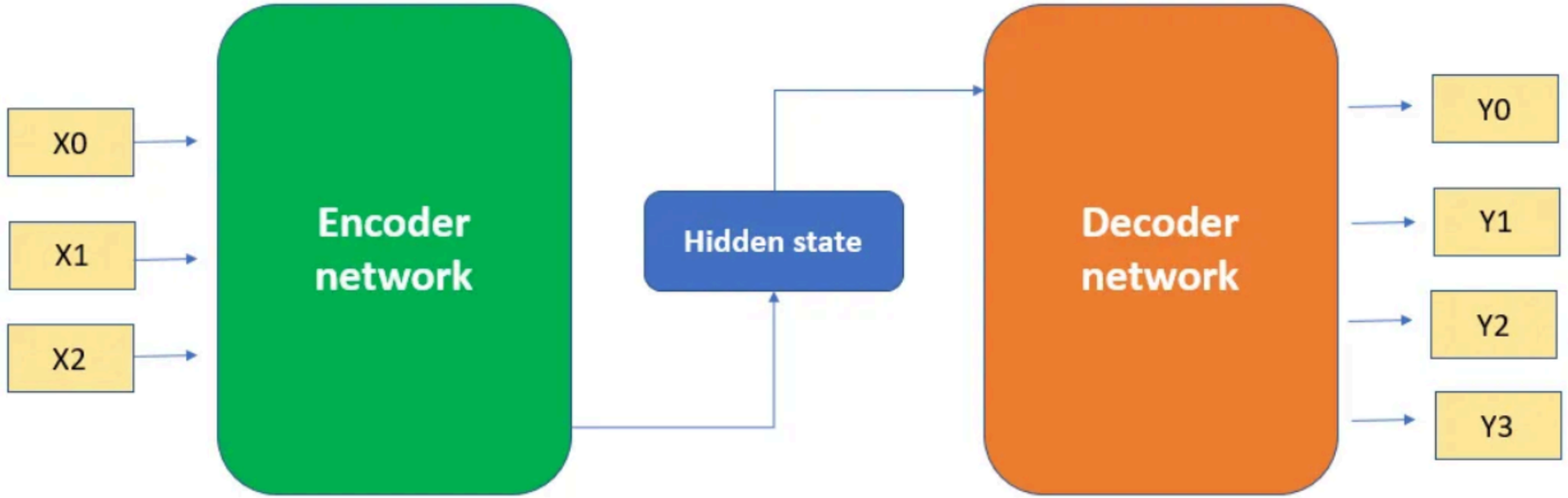
Recurrent neural network

- Variants of RNNs



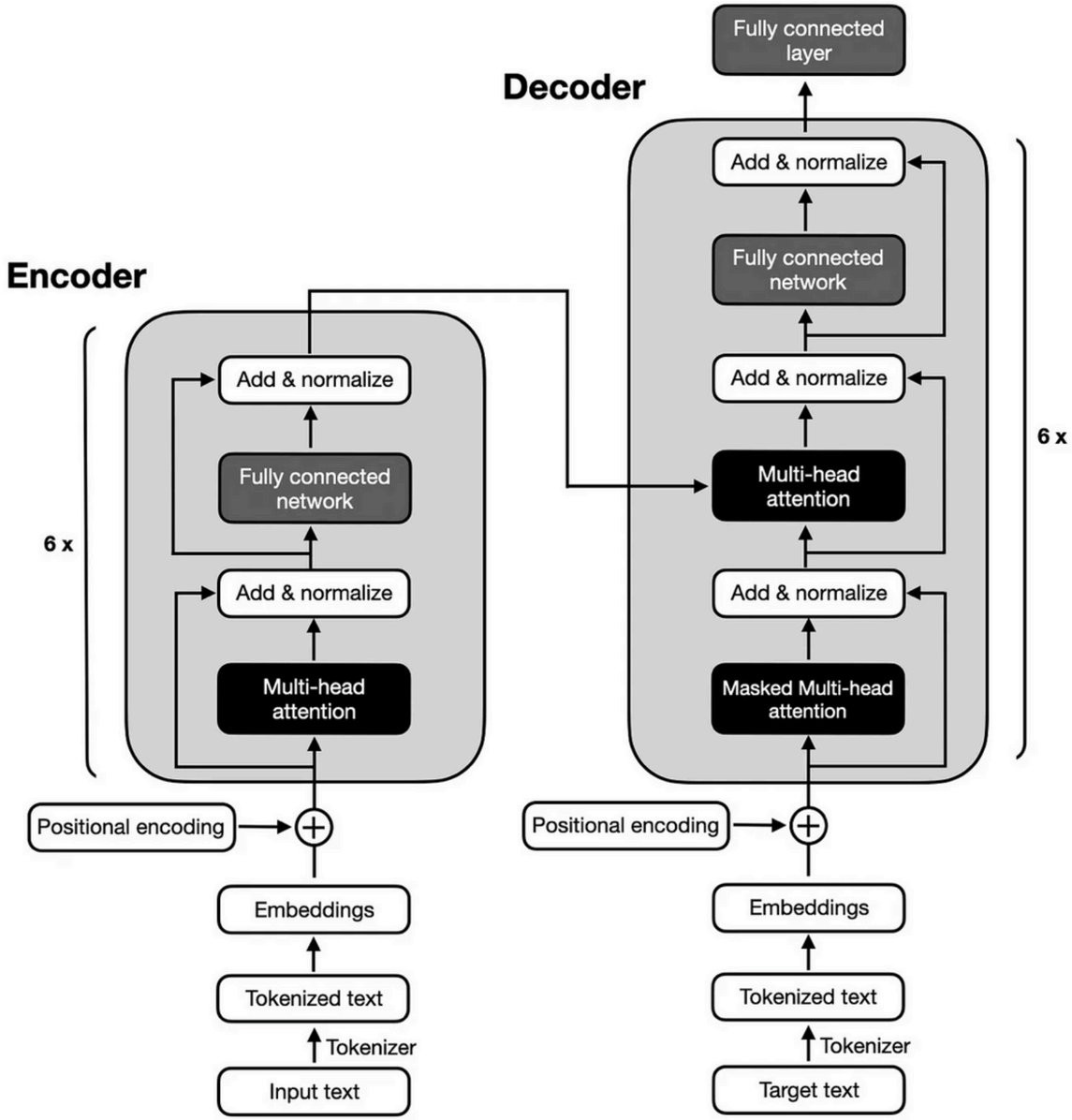
Transformers

Encoder-Decoder Model



Transformers

Transformer Architecture




Q
A

Thank You