



Kissipo Learning for Deep Learning

Topic 23: Introduction to NLP (20min)

Hsueh-Ting Chu

KLDL-W9-T23

Course Schedule

- W1 - Course Introduction
- W2 - DL Programming Basics(1)
- W3 - DL Programming Basics(2)
- W4 - DL with TensorFlow
- W5 - Midterm
- W6 - DL with PyTorch
- W7 - AOI hands-on project
- W8 - RSD hands-on project
- W9 - NLP hands-on project
- W10 - Final exam

DL: Deep Learning

AOI: Automated Optical Inspection

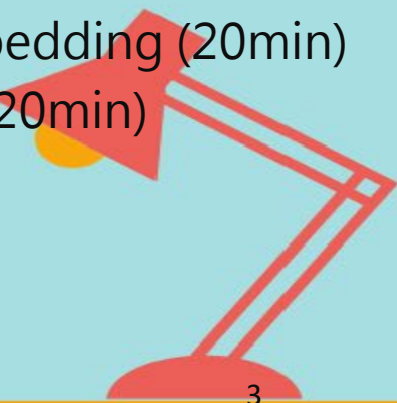
RSD: Road Sign Detection

NLP: Natural Language Processing



Topics

- Topic 01: Introduction to Deep Learning (20min)
- Topic 02: KISSipo Learning for Deep Learning (20min)
- Topic 03: Python quick tutorial (20min)
- Topic 04: Numpy quick tutorial (15min)
- Topic 05: Pandas quick tutorial (15min)
- Topic 06: Scikit-learn quick tutorial (15min)
- Topic 07: OpenCV quick tutorial (15min)
- Topic 08: Image Processing basics (20min)
- Topic 09: Machine Learning basics (20min)
- Topic 10: Deep Learning basics (20min)
- Topic 11: TensorFlow overview (20min)
- Topic 12: CNN with TensorFlow (20min)
- Topic 13: RNN with TensorFlow (20min)
- Topic 14: PyTorch overview (20min)
- Topic 15: CNN with PyTorch (20min)
- Topic 16: RNN with Pytorch (20min)
- Topic 17: Introduction to AOI (20min)
- Topic 18: AOI simple Pipeline (A) (20min)
- Topic 19: AOI simple Pipeline (B) (20min)
- Topic 20: Introduction to Object detection (20min)
- Topic 21: YoloV5 Quick Tutorial (20min)
- Topic 22: Using YoloV5 for RSD (20min)
- **Topic 23: Introduction to NLP (20min)**
- Topic 24: Introduction to Word Embedding (20min)
- Topic 25: Name prediction project (20min)



Week 9 Topics

- Topic 23: Introduction to NLP (20min)
- Topic 24: Introduction to Word Embedding (20min)
- Topic 25: Name prediction project (20min)

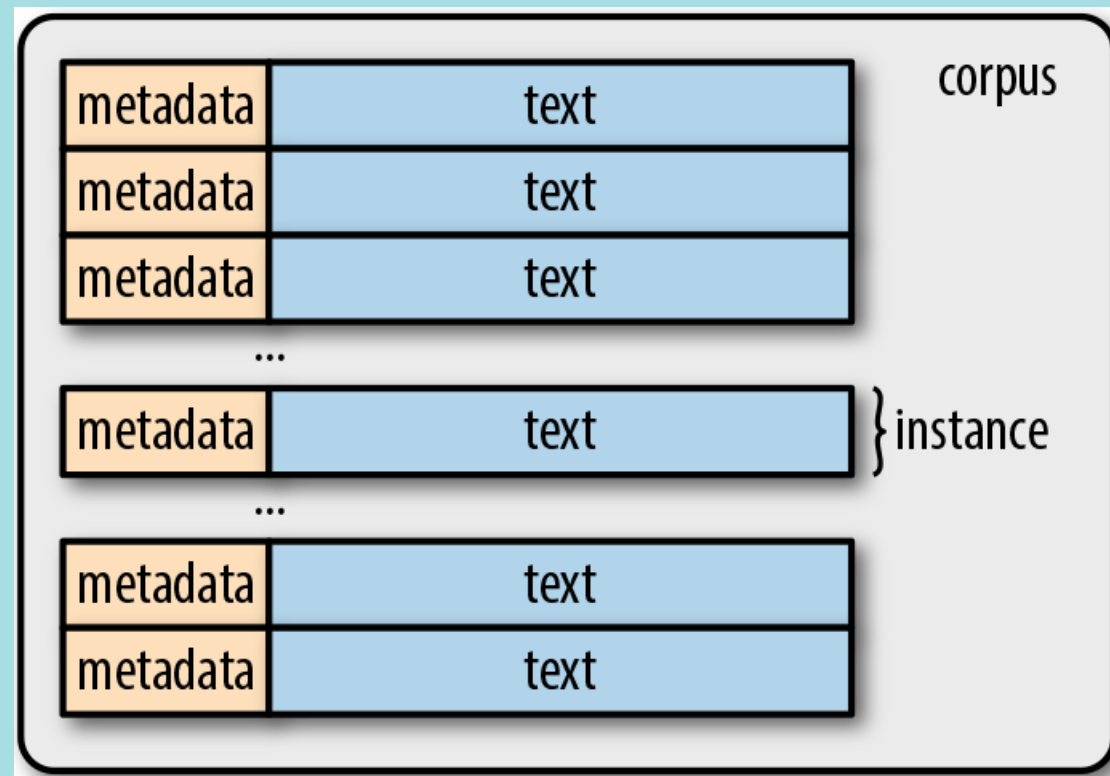


Problems of natural language processing

- Sentiment Analysis
- Text Summarization
- Authorship Attribution
- Recommender System
- Language Translation
- Question Answering



The corpus: the starting point of NLP tasks



The process of breaking a text down into tokens is called *tokenization*.

Sentiment Analysis



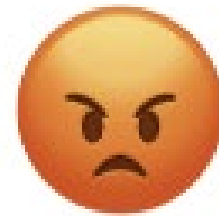
My experience
so far has been
fantastic!

POSITIVE



The product is
ok I guess

NEUTRAL

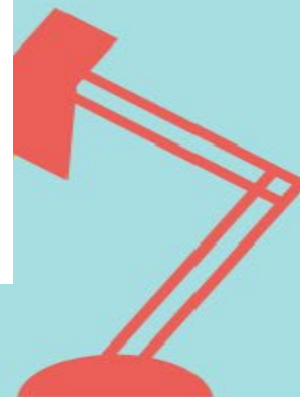
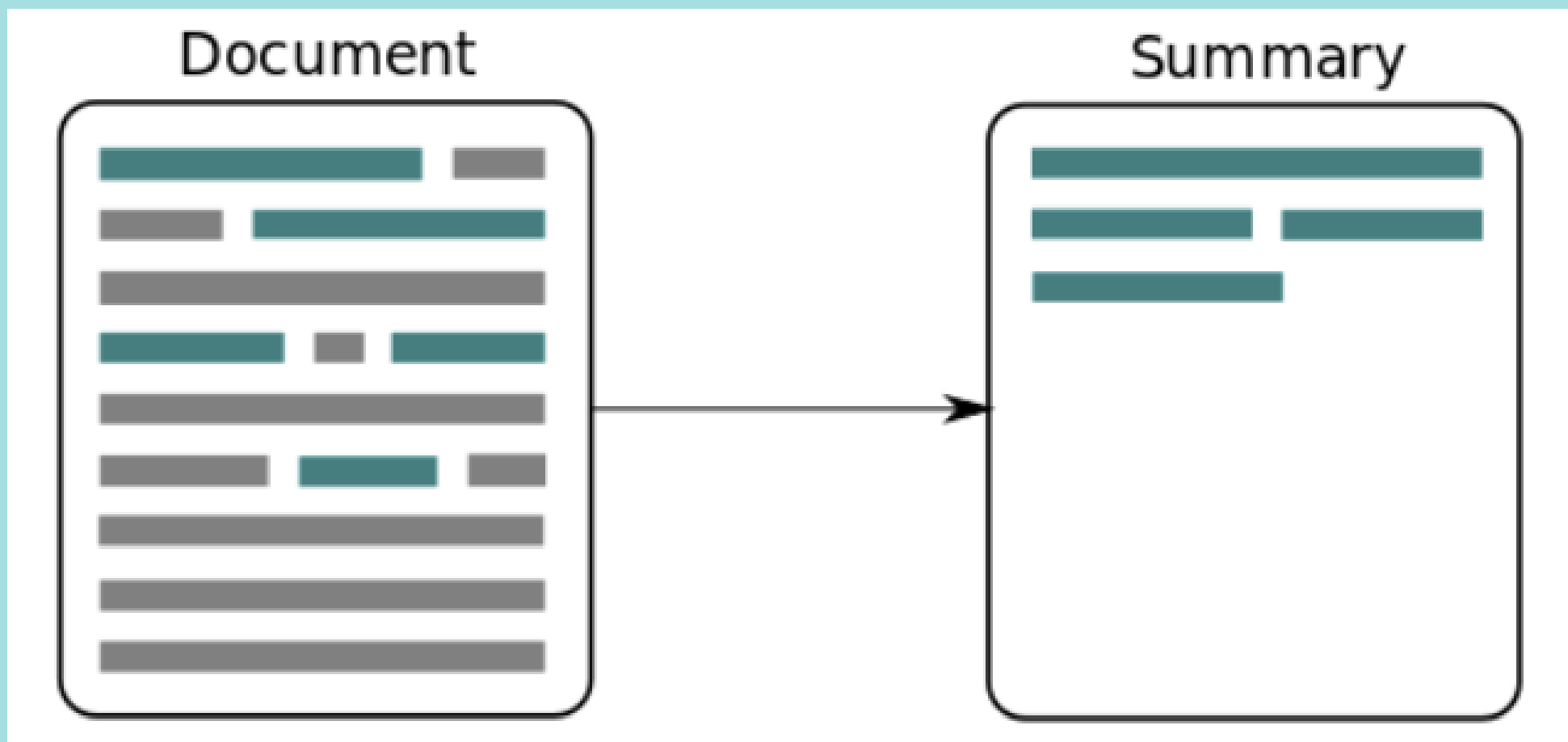


Your support team is
useless

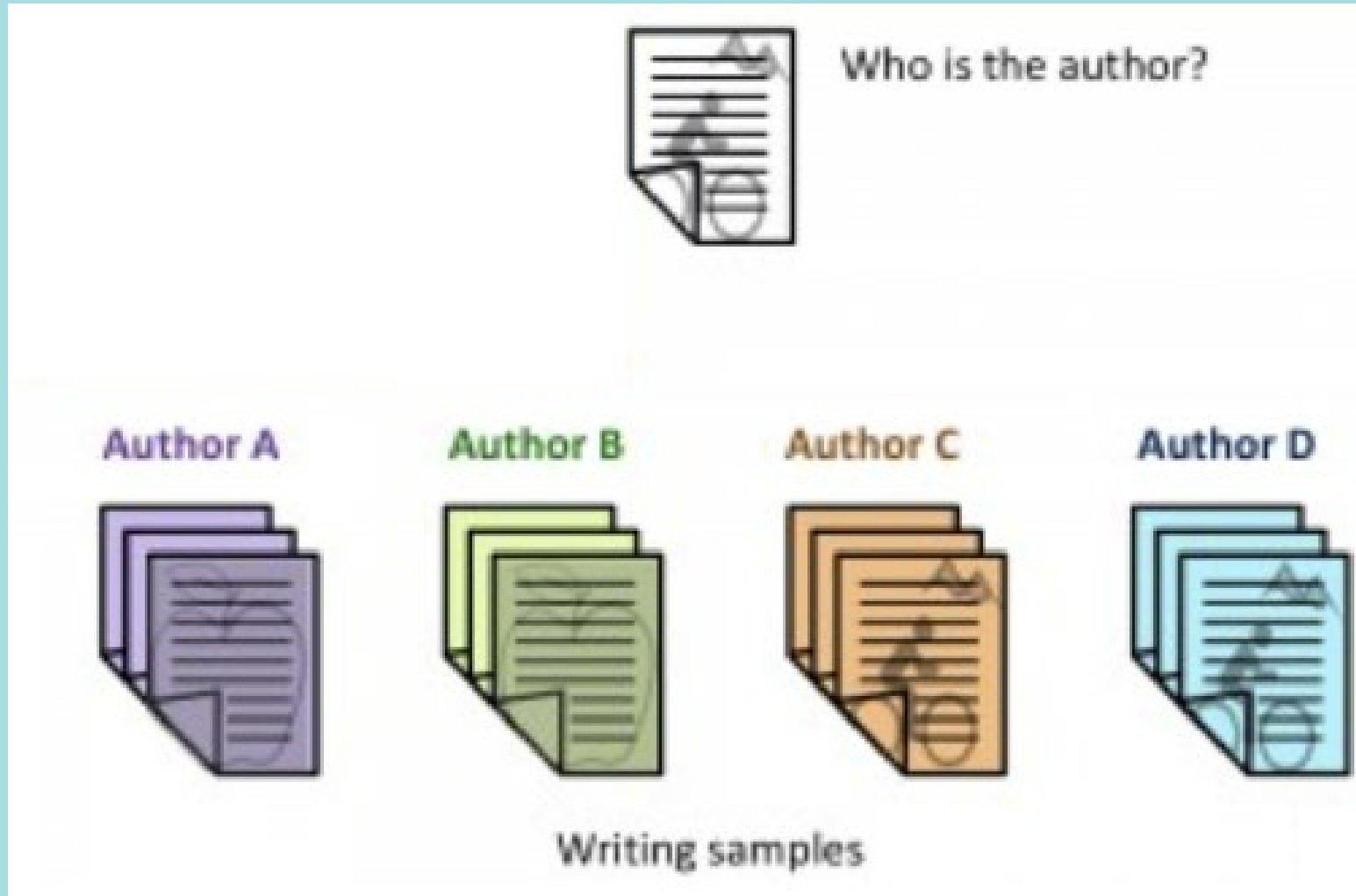
NEGATIVE



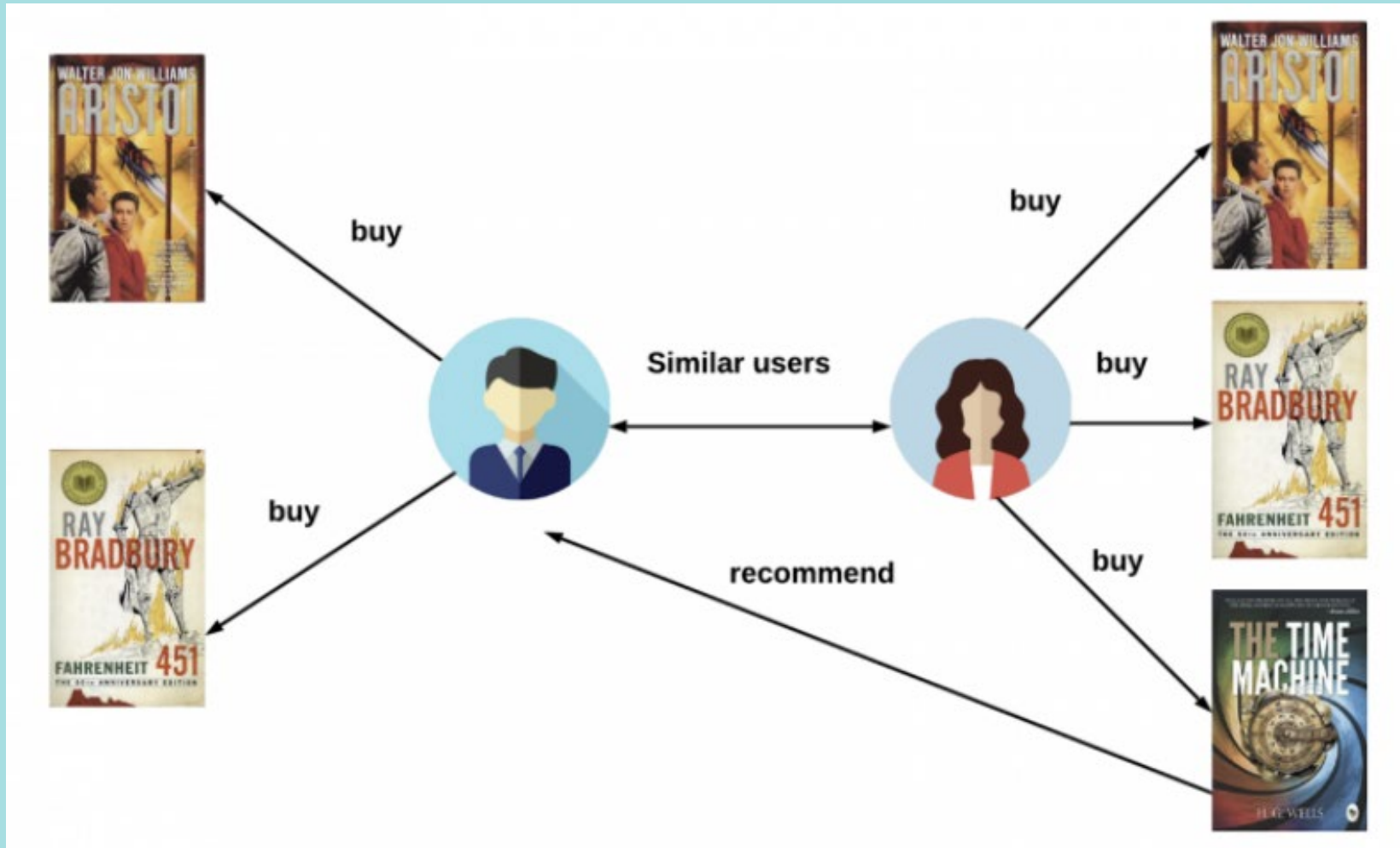
Text Summarization



Authorship Attribution





Recommender System



Language Translation



 Google 翻譯 

文 文字

文件

網站

偵測語言 中文 (簡體) 中文 (繁體) 英文 ↕ 荷蘭文 中文 (簡體) 英文

Collaborative teaching refers to a teaching mode in which two or more teachers and teaching assistants form a teaching team in a professional relationship to plan and cooperate together to carry out teaching activities of a certain unit, a certain field or theme. |

Samenwerkend onderwijs verwijst naar een onderwijsmodus waarin twee of meer leraren en onderwijsassistenten een onderwijsteam vormen in een professionele relatie om onderwijsactiviteiten van een bepaalde eenheid, een bepaald vakgebied of thema uit te voeren en samen te plannen. .

Question Answering



Welcome to ChatGPT



How to develop a search engine



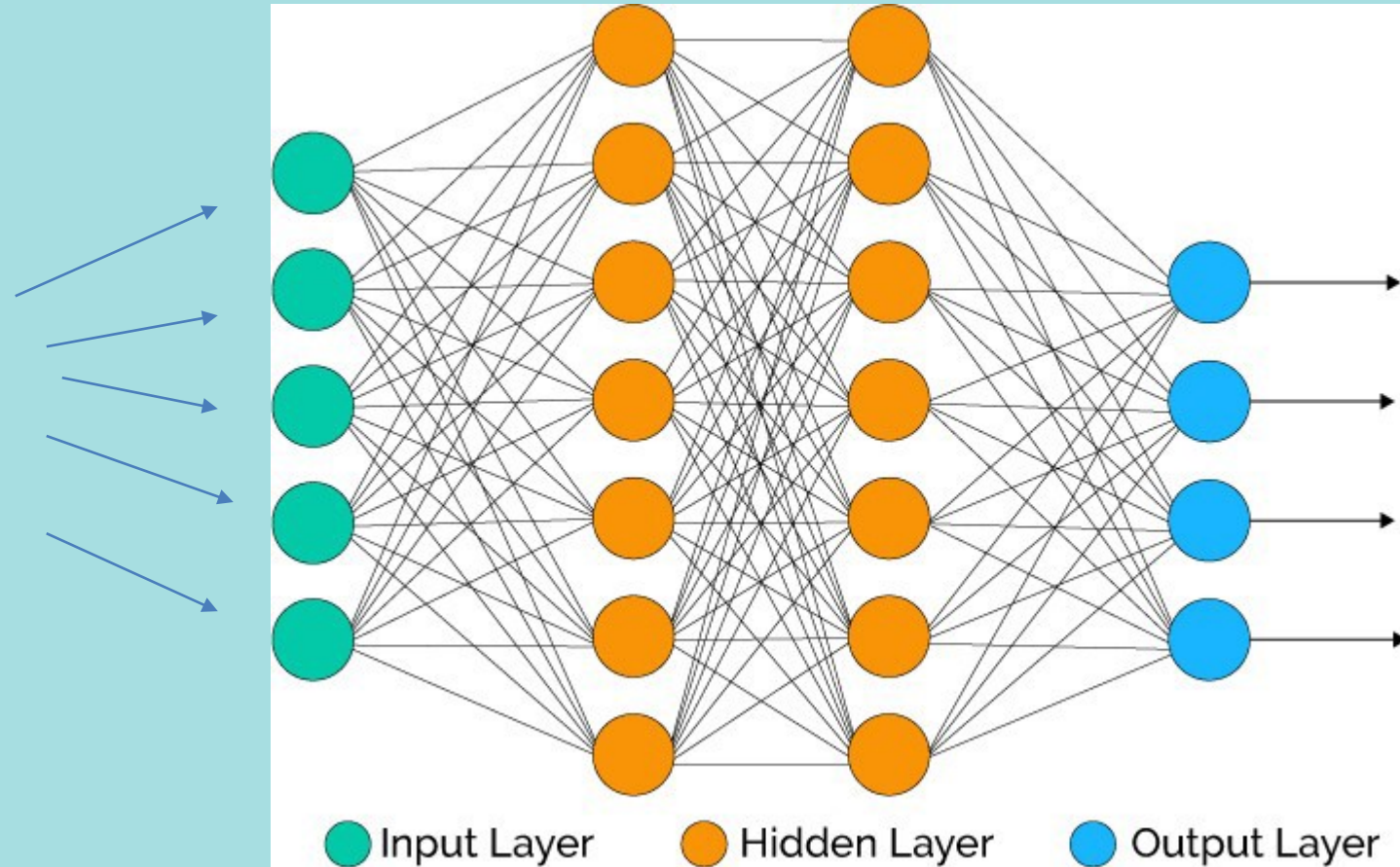
Developing a search engine involves several steps, including:

1. Understanding the requirements and defining the scope of the search engine: You should determine the specific type of data and content you want your search engine to index and provide results for. This could be anything from a small set of documents to a large database of web pages.
2. Data collection: This involves gathering the data that you want your search engine to index. This could involve web crawling, data scraping, or acquiring data from an API.
3. Indexing: This involves processing the collected data and creating an



NLP with Deep Learning Models

Input:
Monica is a dog



Output of Natural Language Processing

- What is natural language processing
- Input is plain text.
- What is the output?
 - Authorship Attribution->Who
 - Sentiment analysis -> emotions (angry, happy, sad)
 - Text Summarization->short sentences
 - Recommender System->goods
 - Language translation->sentences in another language
 - Question Answering->answers (plain text)



Natural Language Processing Models

- N-grams
- Word Embedding
- Word2Vec
- Sequence to sequence
- Attention
- BERT



n-grams

- the quick brown fox
- **1-gram:** "The," "quick," "brown," and "fox"
(also known as a unigram)
- **2-grams:** "The quick," "quick brown," and "brown fox"
(also known as a bigram)
- **3-grams:** "The quick brown" and "quick brown fox"
(also known as a trigram)
- **4-grams:** "The quick brown fox"



Bag-of-Words (BOW)

- 詞袋(Bag-of-words)

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

The TF-IDF method

$$w_{x,y} = \text{tf}_{x,y} \times \log \left(\frac{N}{\text{df}_x} \right)$$

TF-IDF

Term x within document y

$\text{tf}_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

- A corpus is a collection of text data.
- TF (term frequency) is # of occurrences of a keyword.
- IDF (inverse document frequency) is the $\text{prob}(x)$ of x in the corpus.
- e.g. $\text{TF-IDF} ('xyz') = \text{TF} ('xyz') * \text{IDF} ('xyz')$

TF-IDF: Term frequency-Inverse document frequency

$$w_{x,y} = \text{tf}_{x,y} \times \log\left(\frac{N}{\text{df}_x}\right)$$

TF-IDF

Term x within document y

$\text{tf}_{x,y}$ = frequency of x in y
 df_x = number of documents containing x
 N = total number of documents

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

	Doc 1	Doc 2	...	Doc n
Term(s) 1	12	2	...	1
Term(s) 2	0	1	...	0
...
Term(s) n	0	6	...	3

Frequency of term in a large set of documents



Common stop words.
Low TF-IDF

Less frequent terms
earn higher TF-IDF
with increased usage

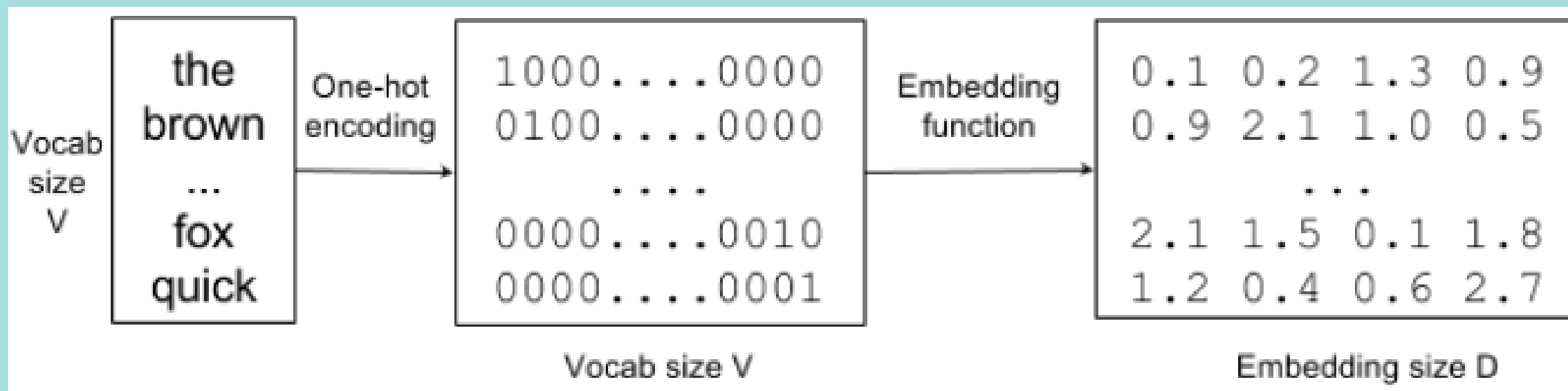
Terms with the
highest TF-IDF may
indicate importance

Frequency of term on a single page



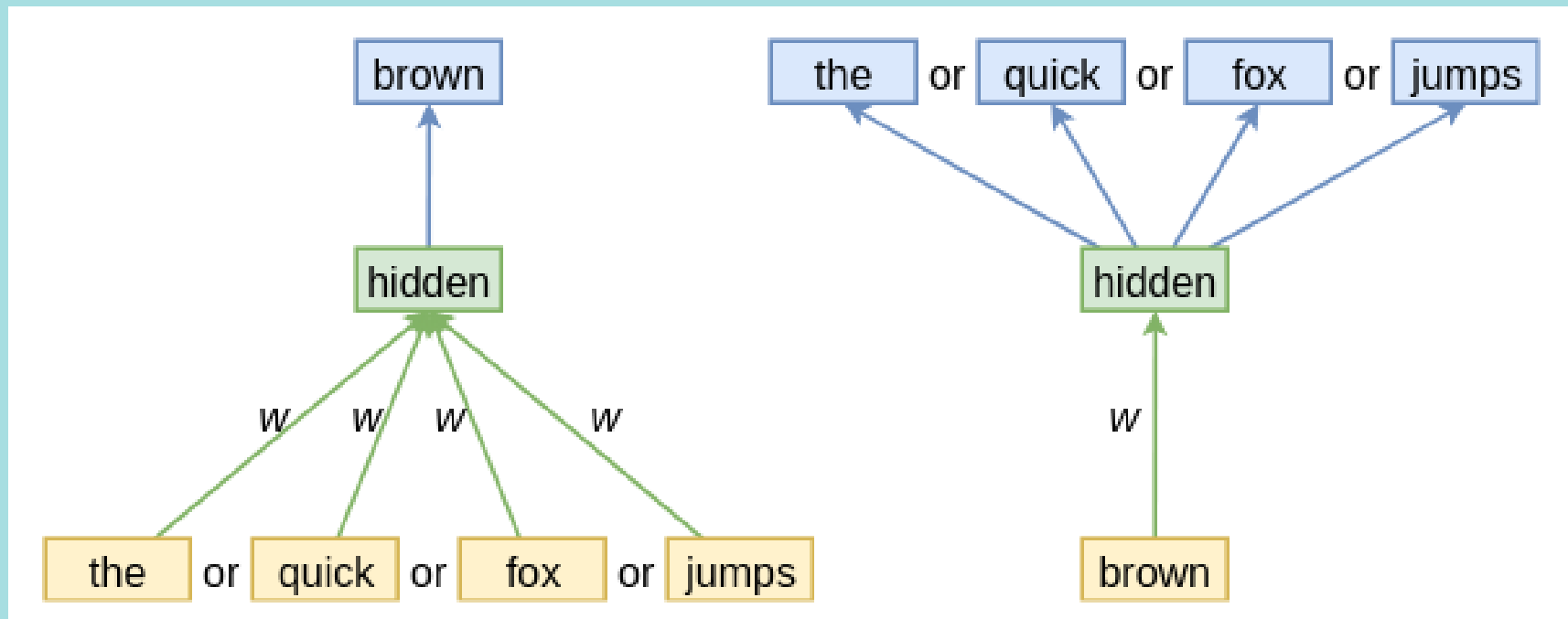
Basic vector models for Neural language

- Words \rightarrow one-hot encoding \rightarrow word embedding vectors

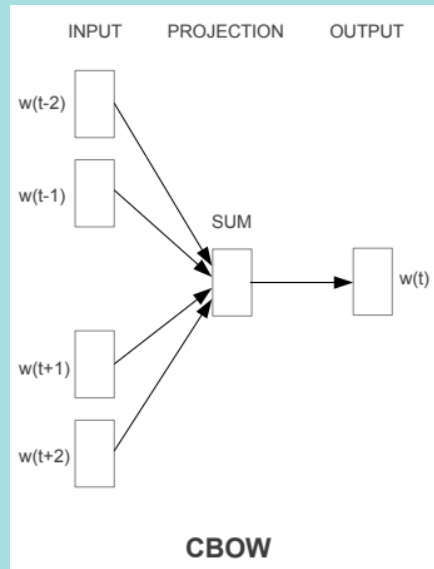


word2vec

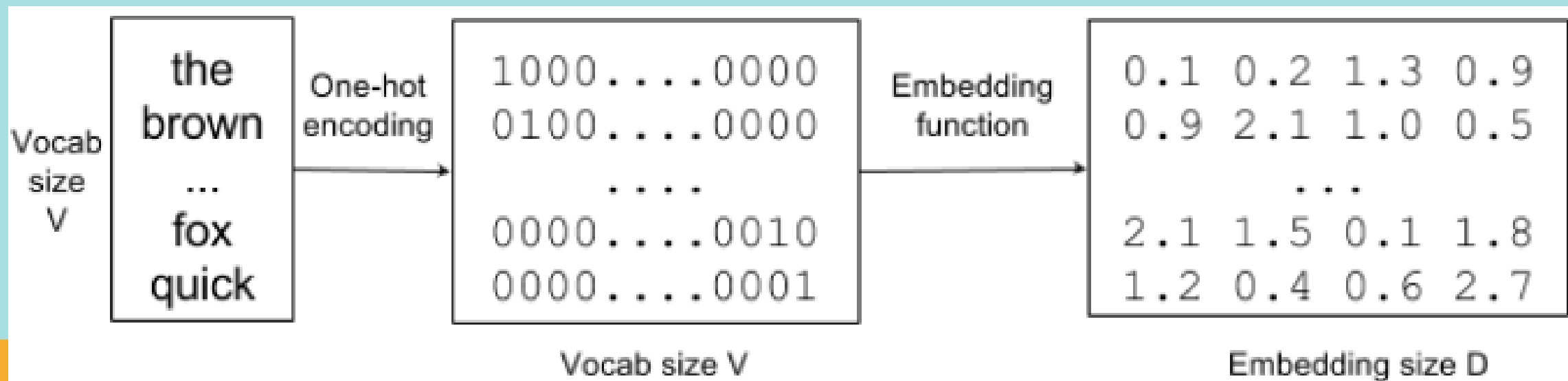
- continuous bag-of-words (CBOW)
- Skip-gram



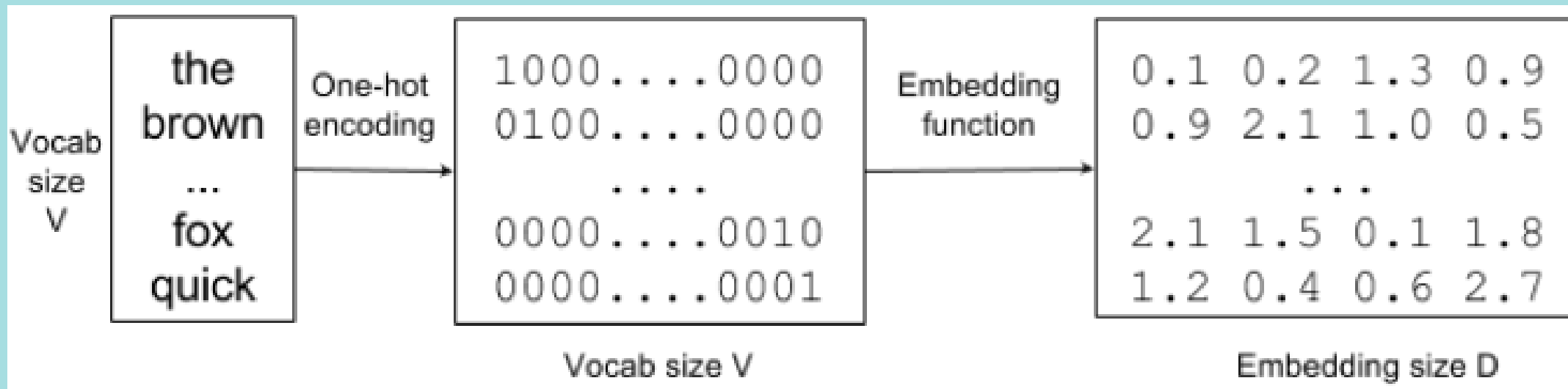
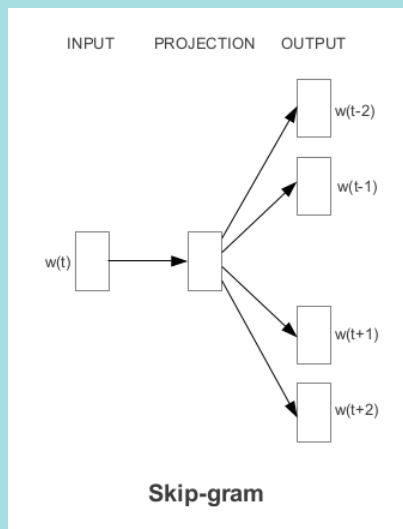
Continuous bag-of-words (CBOW)



詞向量



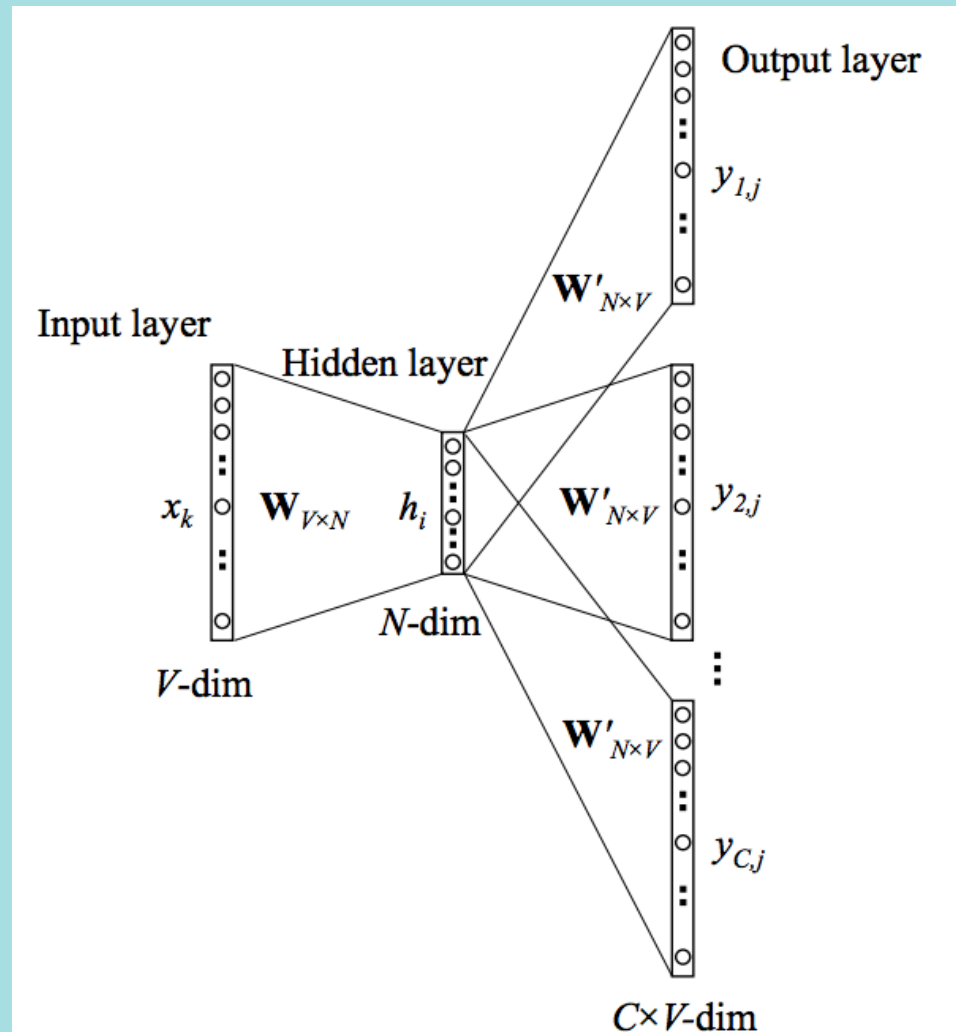
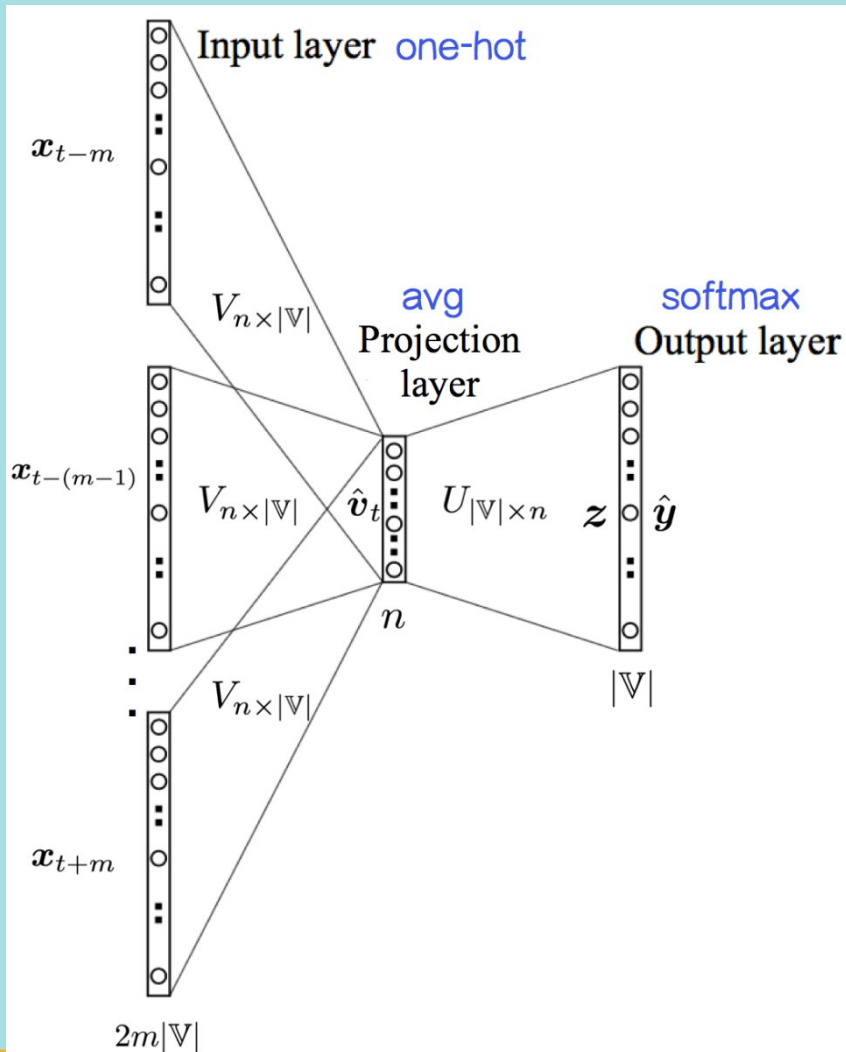
Skip-gram



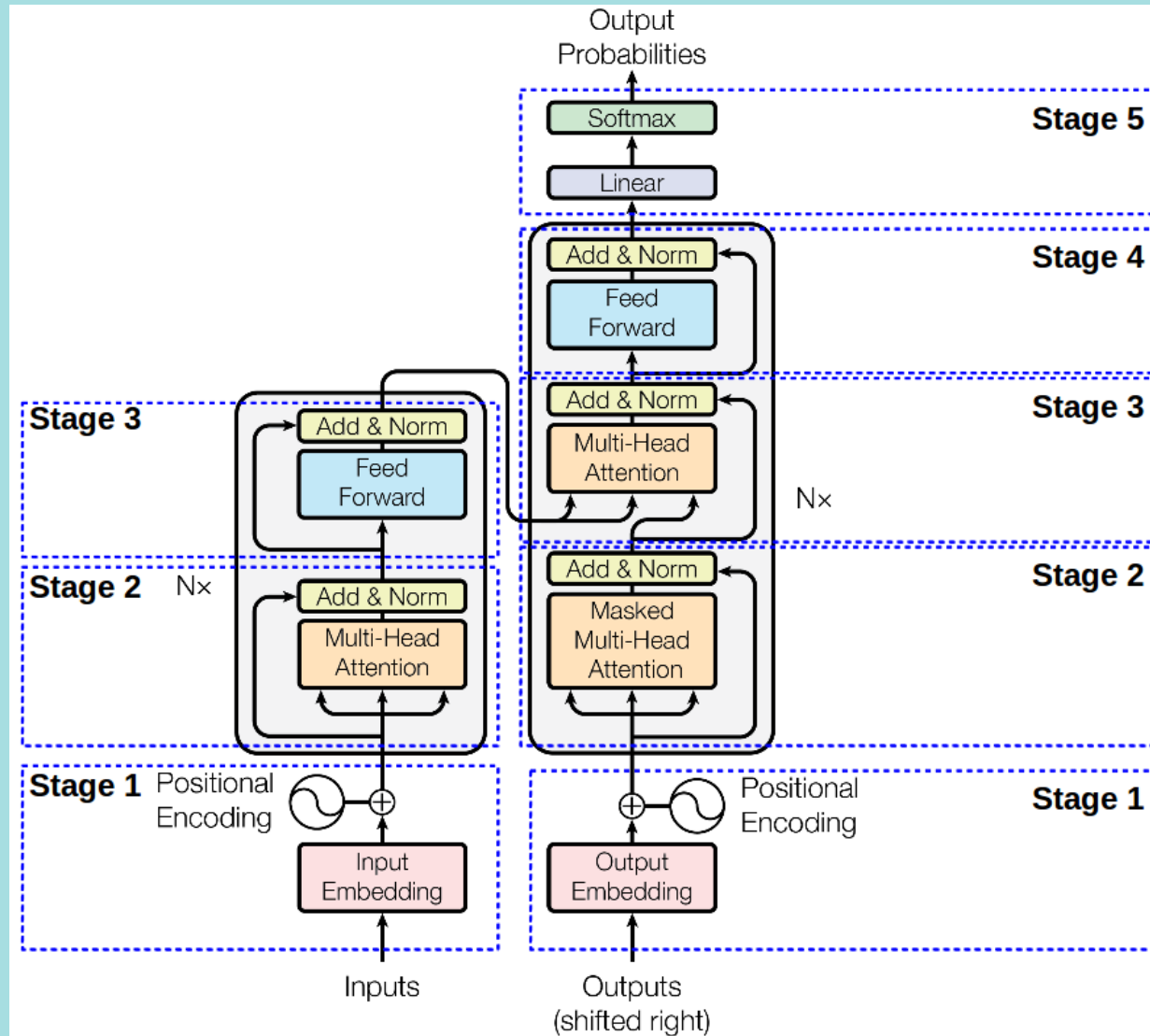
Word2vector

Continuous Bag of Words Model (CBOW)

Skip-Gram Model

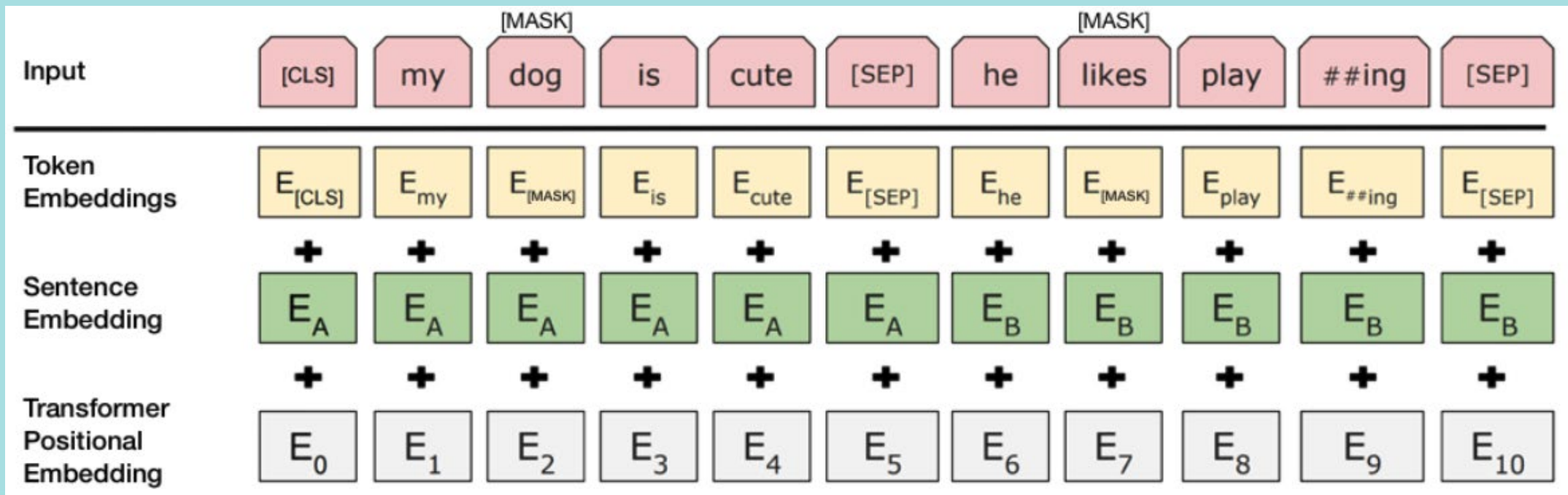


Transformers



BERT (Bidirectional Encoder Representations from Transformers)

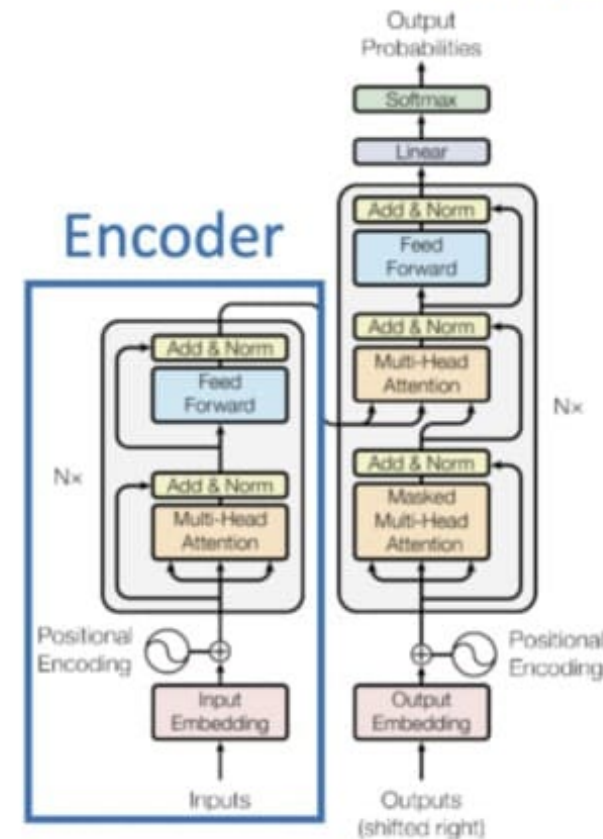
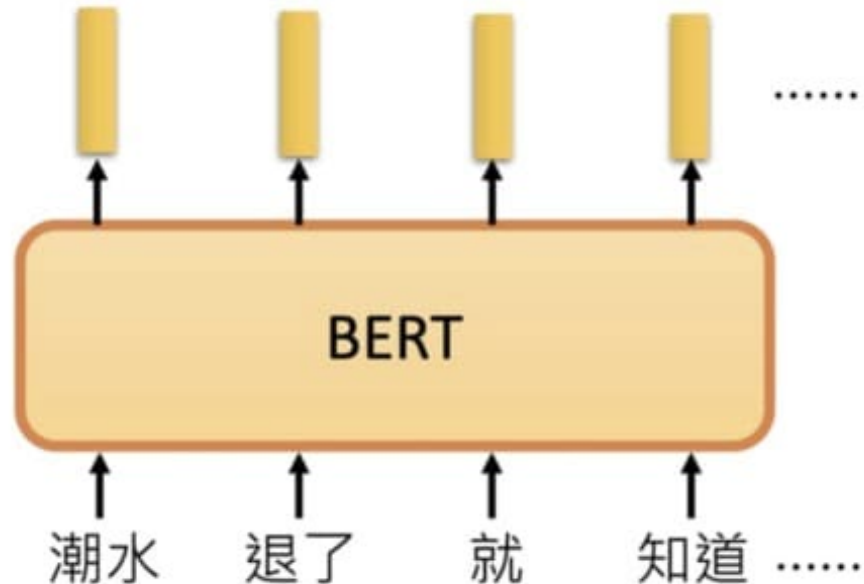
- Masked LM (MLM)
 - 15% of the words in each sequence are replaced with a [MASK] token.
- Next Sentence Prediction (NSP)
 - A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.



Bidirectional Encoder Representations from Transformers (BERT)



- BERT = Encoder of Transformer
Learned from a large amount of text
without annotation

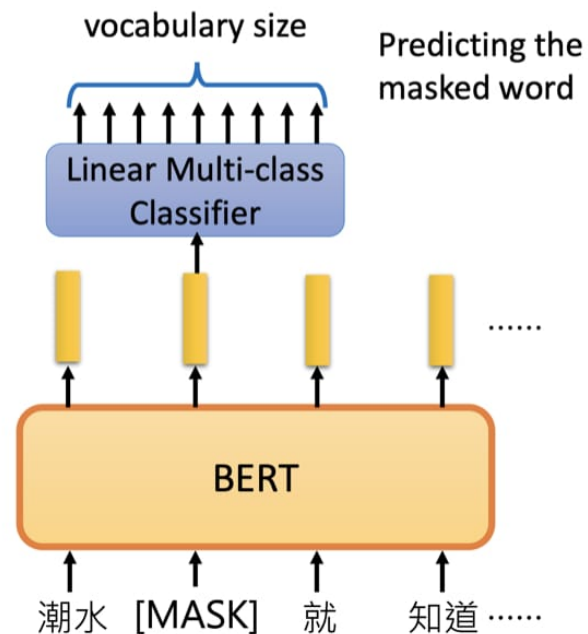


Training of BERT

預訓練任務 1：克漏字填空

Training of BERT

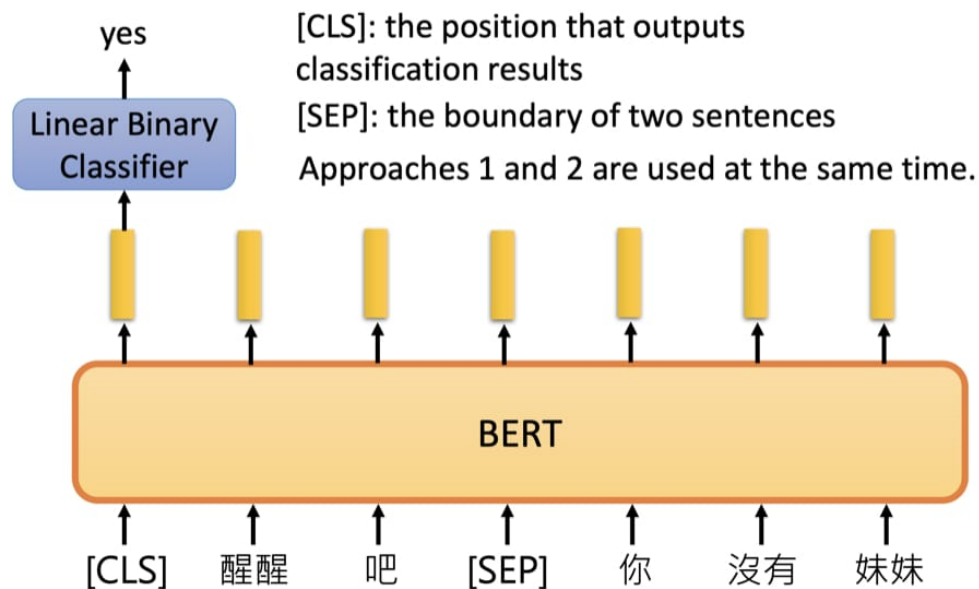
- Approach 1:
Masked LM



預訓練任務 2：下個句子預測

Training of BERT

Approach 2: Next Sentence Prediction



From N-grams to Word2vector to BERT

- Word->Symbol
 - I am a student => ?
- N-grams
 - unigram model, bigram model, trigram model
 - one-hot encoding
- TF-IDF
 - Term frequency
 - Inverse document frequency
- Word2vector
 - Continuous Bag of Words Model (CBOW)
 - Skip-Gram Model
- BERT
 - Transformers
 - Bidirectional Encoder Representation



SQuAD-Stanford Question Answering Dataset

What is SQuAD?

Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable.

SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

Explore SQuAD2.0 and model predictions

SQuAD2.0 paper (Rajpurkar & Jia et al. '18)

SQuAD 1.1, the previous version of the SQuAD dataset, contains 100,000+ question-answer pairs on 500+ articles.

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Nov 06, 2019	ALBERT + DAAF + Verifier (ensemble) PINGAN Omni-Sinitic	90.002	92.425
2 Sep 18, 2019	ALBERT (ensemble model) Google Research & TTIC https://arxiv.org/abs/1909.11942	89.731	92.215
3 Jul 22, 2019	XLNet + DAAF + Verifier (ensemble) PINGAN Omni-Sinitic	88.592	90.859
3 Nov 22, 2019	albert+verifier (single model) Ping An Life Insurance Company AI Team	88.355	91.019
4 Sep 16, 2019	ALBERT (single model) Google Research & TTIC https://arxiv.org/abs/1909.11942	88.107	90.902

A reading comprehension dataset consisting of 100,000+ questions on a set of Wikipedia articles.

Thanks!

Q&A

