# Word Embedding

Topic 6

**Dwi Intan Af'idah, S.T., M.Kom**

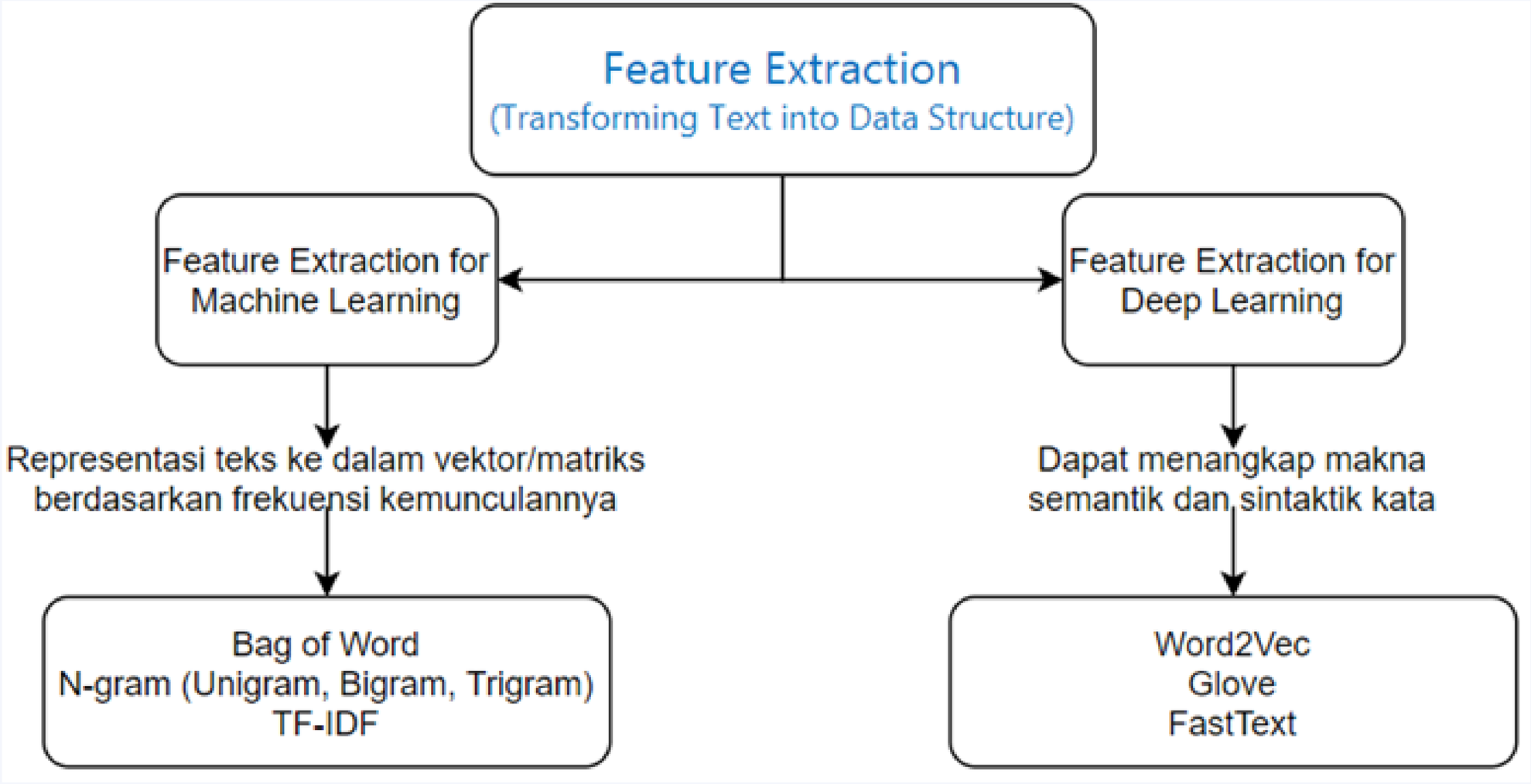
# Material

01 What are word embedding?

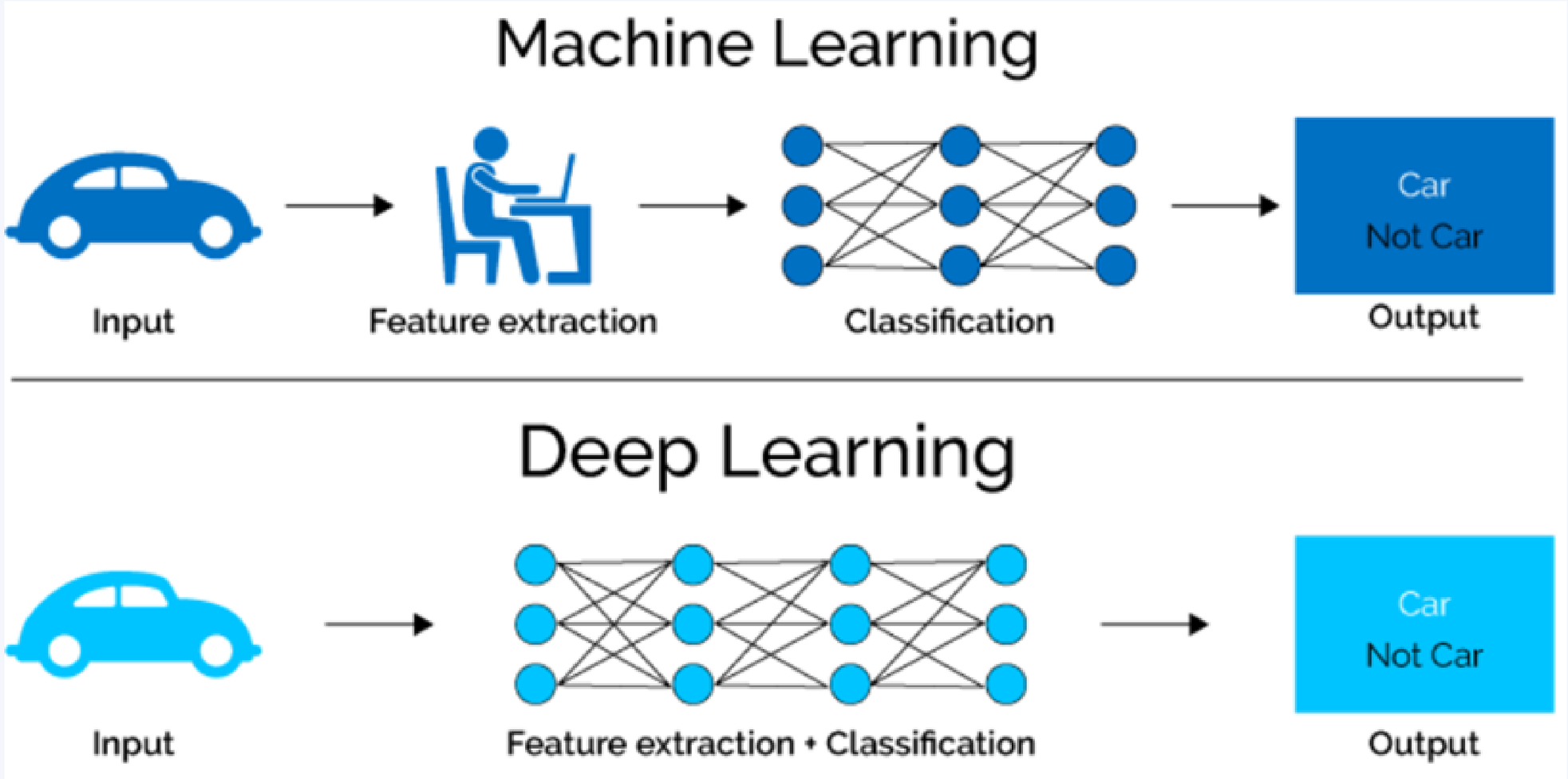
02 Word Embedding Algorithms

03 Programming of Word2Vec

# Growth of Feature Extraction



# Feature Extraction vs Word Embedding



# What are word embedding?

# Word Embedding

A word embedding is a learned representation for text where words that have the same meaning have a similar representation.

It is this approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems.

--**Neural Network Methods in Natural Language Processing, 2017.**

Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in a way that resembles a neural network, and hence the technique is often lumped into the field of deep learning.

--**A Neural Probabilistic Language Model, 2003.**

The distributed representation is learned based on the usage of words. This allows words that are used in similar ways to result in having similar representations, naturally capturing their meaning.

--**“A synopsis of linguistic theory 1930-1955“, in Studies in Linguistic Analysis**

# Understanding Word Embedding

Word Embeddings adalah pembelajaran representasi kata yang dimana setiap kata diwakili menggunakan vektor dalam ruang n-dimensi.

Kata-kata dengan arti yang sama seharusnya memiliki kesamaan representasi.

Representasi ini juga dapat membantu dalam mengidentifikasi sinonim, antonim, dan berbagai hubungan antar kata lainnya.

# Simple Example for Word Embedding

Vocabulary:
Man, woman, boy, girl, prince, princess, queen, king, monarch

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| man | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| woman | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| boy | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| girl | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| prince | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| princess | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| queen | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| king | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| monarch | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Each word gets a 1x9 vector representation

# Characteristics of Word Embedding

- Every word has a unique word embedding (or “vector”), which is just a list of numbers for each word.
- The word embeddings are multidimensional; typically for a good model, embeddings are between 50 and 500 in length.
- For each word, the embedding captures the “meaning” of the word.
- Similar words end up with similar embedding values.

# Word Embedding Pretraining

Pembelajaran menggunakan dataset yang besar pada domain permasalahan tertentu

**01** Pretraining menggunakan dataset dari korpus artikel dalam jumlah besar

**02** Pretraining menggunakan dataset yang dimiliki

**03** Tidak melakukan pretaining, tapi menggunakan hasil pretraining yang sudah ada

# Word Embedding Algorithms

# Word Embeddings Algorithms

## Word2Vec

(i) Pada tahun 2013, Mikolov dkk. memperkenalkan word2vec dengan dua metode utamanya yaitu Skip-gram dan Continous Bag of Words (CBOW) (Mikolov et al., 2013).

## Glove

(ii) Kemudian pada tahun berikutnya, diperkenalkan word embedding baru oleh Pennington dkk. (Pennington, Socher and Manning, 2014) yaitu GloVe yang menggunakan rasio co-occurrence probability antarkata

## FastText

(iii) Kemudian pada tahun 2017, Bojaniwski dkk. (Bojanowski et al., 2017) mengembangkan model word2vec dan memperkenalkan FastText yang mempelajari informasi subword dari kata.

# Word2Vec

- Word2vec meng-capture hubungan dalam teks; akibatnya, kata-kata yang mirip memiliki representasi yang serupa.
- Beberapa contoh untuk memahami apa hubungan dan analogi yang ditangkap oleh model Word2vec. Contoh yang sangat sering digunakan berkaitan dengan embedding dari King, Man, Queen, dan Woman.

$$vector\ (Man) - vector\ (King) + vector\ (Queen) = vector\ (Woman)$$

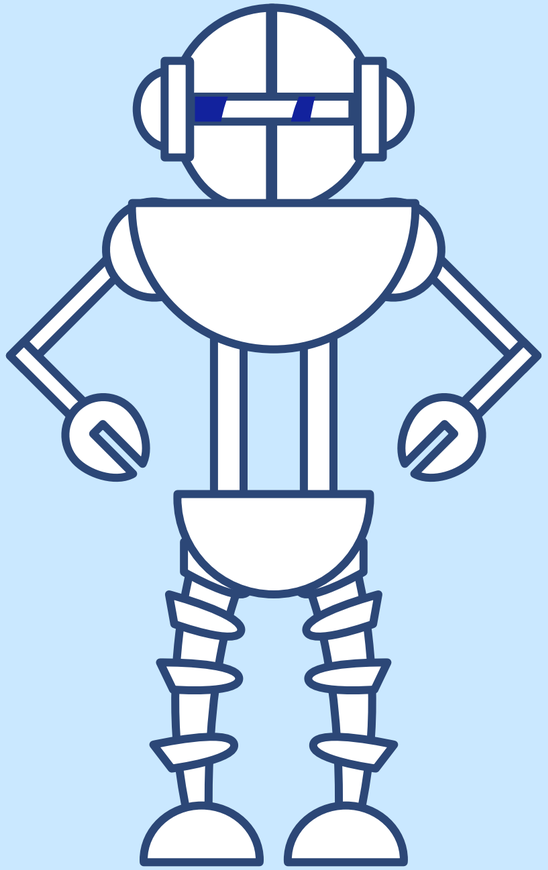
$$vector\ (Man) + vector\ (Queen) = vector\ (King) + vector\ (Woman)$$

# Unsupervised of Word2Vec

- Meskipun kata-kata diprediksi, komponen prediksi atau atribut kelas itu sendiri berasal dari teks atau corpus.
- Oleh karena itu, tidak ada atribut kelas khusus yang tersedia, tidak seperti halnya dalam skenario pembelajaran terawasi.
- Karena ini, Word2vec berada di bawah kelas dari algoritma yang tidak diawasi.
- Semua pembelajaran berasal dari data yang tidak terstruktur dengan cara tanpa pengawasan.

# Parameter of Word2Vec

- **Learning Models / Arsitektur**
- **Metode Evaluasi**
- **Dimensi**

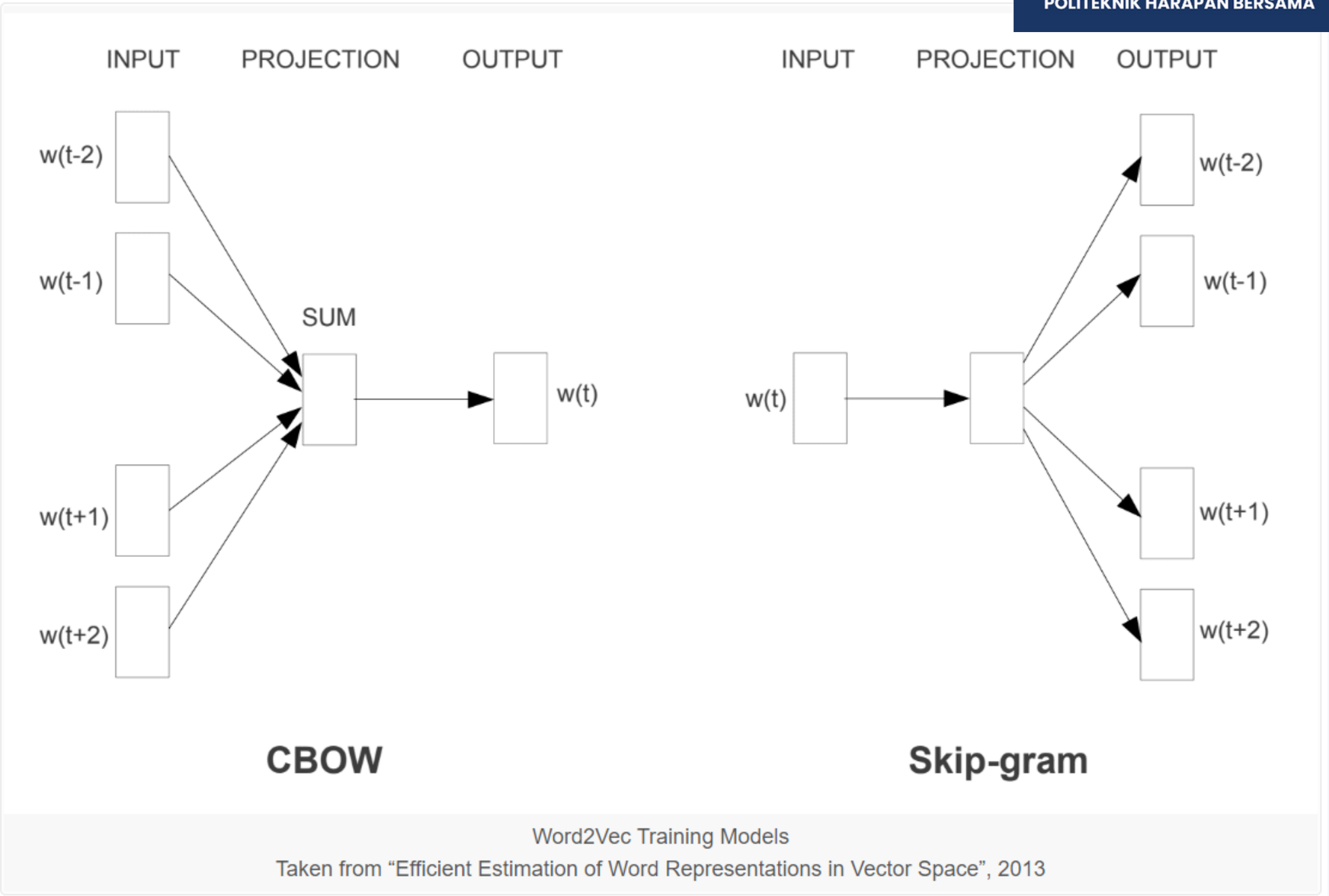


# Learning Models of Word2Vec

Dalam arsitektur Word2vec, upaya dilakukan untuk melakukan salah satu dari hal berikut:

- Memprediksi kata target berdasarkan kata konteks (Continuous Bag-of-Words, or CBOW model).
- Memprediksi kata konteks berdasarkan kata target (Continuous Skip-Gram Model)

# Learning Models of Word2Vec




Word2Vec Training Models

Taken from “Efficient Estimation of Word Representations in Vector Space”, 2013

# Dimension of Word2Vec

- Keluaran dari algoritma Word2vec adalah |V| * Matriks D, di mana |V| adalah ukuran kosakata yang kita inginkan untuk representasi vektor dan D adalah jumlah dimensi yang digunakan untuk mewakili setiap vektor kata.
- Setiap baris dalam matriks ini membawa embedding untuk kata dalam kumpulan kosa kata.
- Nilai D dapat diubah dan tergantung pada beberapa faktor, seperti ukuran korpus teks dan berbagai hubungan yang perlu ditangkap.
- Umumnya, D mengambil nilai antara 50 dan 300 dalam kasus penggunaan kehidupan nyata.

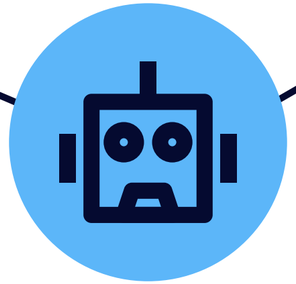
# Evaluation Method of Word2Vec

## Negative Sampling

Negative Sampling memungkinkan pelatihan untuk hanya memodifikasi sebagian kecil dari bobot, daripada semuanya untuk setiap sampel pelatihan.

ebagian kecil kata negatif atau kata-kata yang tidak diharapkan muncul dalam konteks dari kata target, dipilih dan hanya bobot negatif ini saja bobotnya diperbaharui.

Akibatnya hanya sebagian kecil saja matriks bobot yang diperbarui



## Hierarchical Softmax

Hierarchical Softmax menggunakan representasi pohon biner dari lapisan keluaran dengan kata-kata sebagai daunnya dan, untuk setiap simpul, secara eksplisit mewakili probabilitas relatif dari simpul turunannya.

Model binary tree ini menjadikan kata yang jarang muncul pasti akan mewarisi representasi vektor di atasnya

# Programming of Word2Vec

Kampus Merdeka INDONESIA JAYA

Poltek Harber
POLITEKNIK HARAPAN BERSAMA

# Membaca Corpus Wikipedia

```
from __future__ import print_function

# Ignore warnings dari gensim
import warnings
warnings.filterwarnings(action='ignore', category=UserWarning, module='gensim')

import logging
import os.path
import sys

from gensim.corpora import WikiCorpus

program = os.path.basename(sys.argv[0])
logger = logging.getLogger(program)

logging.basicConfig(format='%(asctime)s: %(levelname)s: %(message)s')
logging.root.setLevel(level=logging.INFO)
logger.info("running %s" % ' '.join(sys.argv))

namaFileInput = "idwiki-latest-pages-articles.xml.bz2"
namaFileOutput = "wiki.id.text"

space = " "
i = 0

# Write file ke variabel namaFileOutput encoder utf-8  '
output = open(namaFileOutput, 'w', encoding='utf-8')

# lower=False: huruf kecil dan besar dibedakan
wiki = WikiCorpus(namaFileInput, lemmatize=False, dictionary={}, lower=False)
for text in wiki.get_texts():
    output.write(' '.join(text) + '\n')
    i = i + 1
    if i % 10000 == 0:
        logger.info("Saved " + str(i) + " articles")

output.close()
logger.info("Finished Saved " + str(i) + " articles")
```

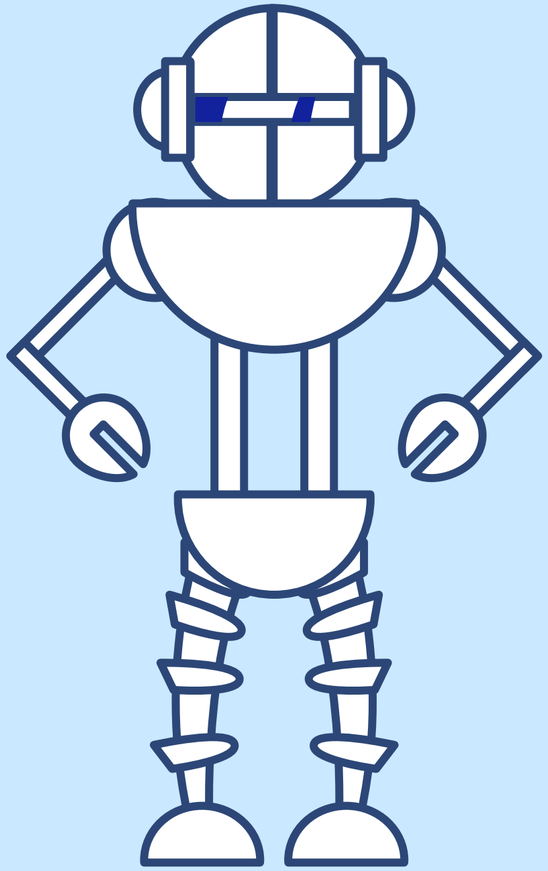
# Pembentukan Model Word2Vec

```
import multiprocessing
import logging
import os.path
import sys
import multiprocessing
from gensim.models import Word2Vec
from gensim.models.word2vec import LineSentence

program = os.path.basename(sys.argv[0])
logger = logging.getLogger(program)

logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s')
logging.root.setLevel(level=logging.INFO)
logger.info("running %s" % ' '.join(sys.argv))

namaFileInput = "wiki.id.text"
namaFileOutput = "w2v_wiki_sg.model"


# size 300 = 300 dimensi vektor, window 10 = 10 pengaruh kata disekitarnya,
#min_count 5 = kata-kata yang muncul < 5 kali akan dikeluarkan dari kosakata dan diabaikan selama pelatihan
#sg 0 = cbow / sg 1 = skip gram, hs 1 = hiarachical softmax,
model = Word2Vec(LineSentence(namaFileInput), size=300, window=10, min_count=5, sg=1, hs=1, workers=multiprocessing.cpu_count())

# trim unneeded model memory = use (much) less RAM
model.init_sims(replace=True)
model.wv.save_word2vec_format(namaFileOutput)
```

# Load Model Word2Vec

```
1 import gensim
2 from gensim.models import Word2Vec
3 from gensim.utils import simple_preprocess
4 import pickle
5
6 from gensim.models.keyedvectors import KeyedVectors
7
8 word_vectors = KeyedVectors.load_word2vec_format('w2v_wiki_sg.model')
9
```

# Reference




- Kedia, A., dan Rasu, M., 2020, Hands-On Python Natural Language Processing, Packt Publishing Ltd., Brimingham, UK
- Nawangsari, R.P., Kusumaningrum, R., dan Wibowo, A., 2019, Word2vec for Indonesian Sentimen Analysis Towards Hotel Reviews: An Evaluation Study, Procedia Computer Science 157, 360–366.
- Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies, 37

# Thank You

credit: canva.com