

# Natural Language Processing

---

Ali Akbar Septiandri

December 29, 2020

Universitas Al Azhar Indonesia

1. Natural Language Processing
2. NLTK
3. Referensi

# Natural Language Processing

---

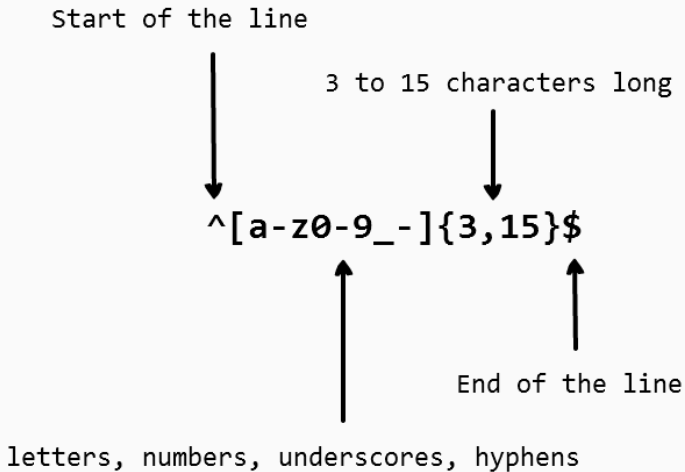
# Apa Itu NLP?

Salah satu ilmu multidisiplin yang berfokus pada interaksi manusia dan komputer melalui bahasa alami manusia. Beberapa hal yang dibahas di dalamnya antara lain:

- Part-of-Speech (POS) tagging
- Parsing
- Stemming
- Machine translation
- Named entity recognition (NER)
- Question answering
- Sentiment analysis
- Automatic summarisation
- Speech recognition
- Text-to-speech

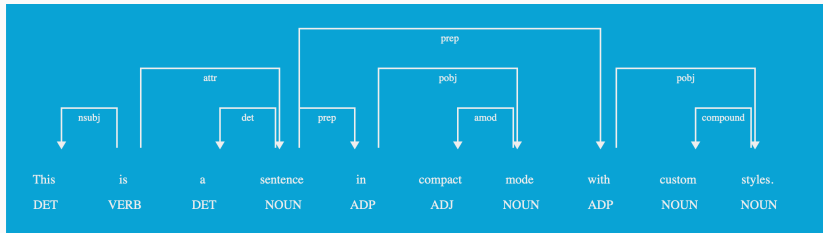
# Kategori Tugas-tugas NLP

- Syntax
  - Part-of-Speech (POS) tagging
  - Parsing
  - Stemming
- Semantics
  - Machine translation
  - Named entity recognition (NER)
  - Question answering
  - Sentiment analysis
- Discourse
  - Automatic summarisation
- Speech
  - Speech recognition
  - Text-to-speech



**Figure 1:** Contoh regular expression. Sumber: tajawal

# Part-of-Speech (POS) Tagging



**Figure 2:** Kategori kata (Sumber: spaCy)

# Named Entity Recognition (NER)

When Sebastian Thrun **PERSON** started working on self-driving cars at Google **ORG** in 2007 **DATE**, few people outside of the company took him seriously.

**Figure 3:** Menemukan orang, organisasi, dan tanggal dalam teks  
(Sumber: spaCy)



# SENTIMENT ANALYSIS



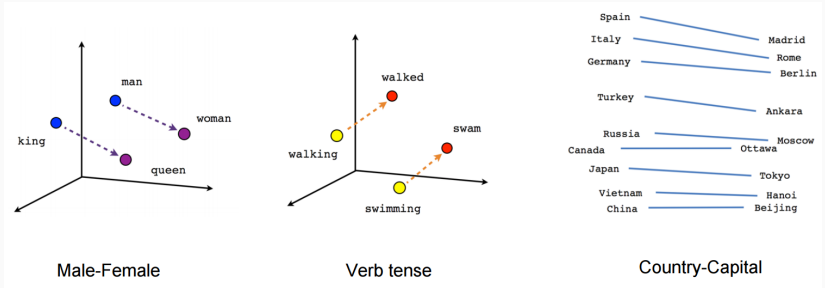
**Figure 4:** "Apa yang menjadi sentimen dari ulasan ini?" Sumber: KDNuggets

"[A] fascinating read from beginning to end." —TYLER COWEN,  
professor of economics, George Mason University, author of *Average Is Over*

THE  
LANGUAGE  
OF  
FOOD

A LINGUIST  
READS THE MENU

DAN  
JURAFSKY

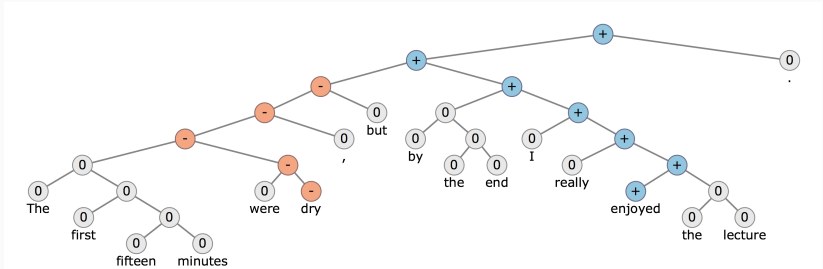


**Figure 5:** Representasi kata dalam vektor (Mikolov et al., 2013).

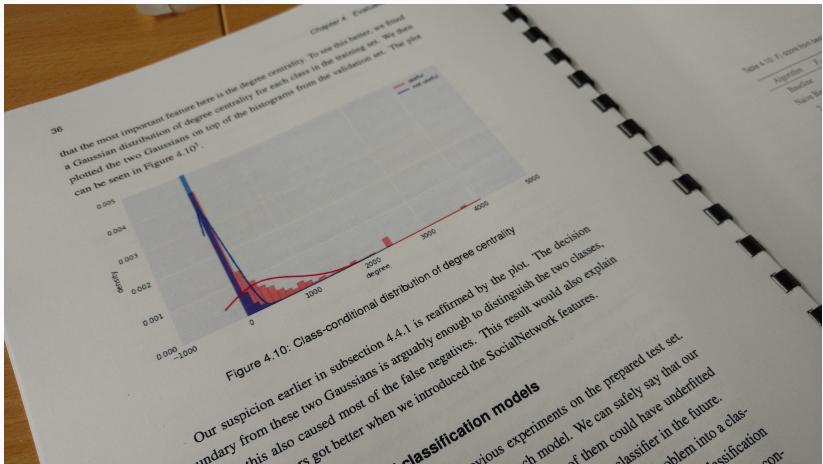
Sumber: TensorFlow

# Demo word2vec

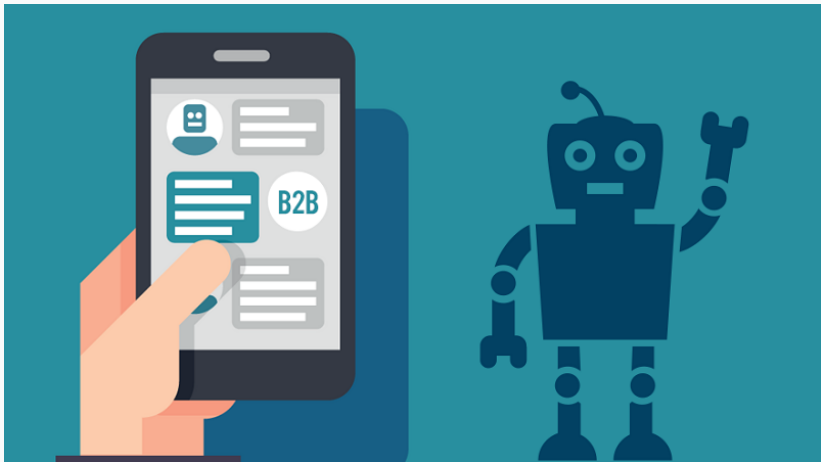
# Sentiment Analysis



**Figure 6:** Hasil analisis sentimen dengan *deep learning* [Socher, 2017]



**Figure 7:** Deteksi plagiarisme dari makalah



**Figure 8:** Penggunaan chatbot untuk bisnis. Sumber: Acquire

- NLP juga dikenal dengan nama *computational linguistics*, karena mencoba merepresentasikan makna dari kata, frasa, kalimat, dan dokumen melalui **distribusinya**



- NLP juga dikenal dengan nama *computational linguistics*, karena mencoba merepresentasikan makna dari kata, frasa, kalimat, dan dokumen melalui **distribusinya**
- Distribusi tersebut direpresentasikan dalam **vektor konteks**

- NLP juga dikenal dengan nama *computational linguistics*, karena mencoba merepresentasikan makna dari kata, frasa, kalimat, dan dokumen melalui **distribusinya**
- Distribusi tersebut direpresentasikan dalam **vektor konteks**
- “Dalam suatu dokumen, kata apa saja yang muncul bersamaan?”

- NLP juga dikenal dengan nama *computational linguistics*, karena mencoba merepresentasikan makna dari kata, frasa, kalimat, dan dokumen melalui **distribusinya**
- Distribusi tersebut direpresentasikan dalam **vektor konteks**
- “Dalam suatu dokumen, kata apa saja yang muncul bersamaan?”
- Begitu pula di level semantik → **Bag-of-Words (BoW) model**

- NLP juga dikenal dengan nama *computational linguistics*, karena mencoba merepresentasikan makna dari kata, frasa, kalimat, dan dokumen melalui **distribusinya**
- Distribusi tersebut direpresentasikan dalam **vektor konteks**
- “Dalam suatu dokumen, kata apa saja yang muncul bersamaan?”
- Begitu pula di level semantik → **Bag-of-Words (BoW) model**
- Bahkan, bisa sampai ke level **karakter!**

Dalam representasi ini, urutan atau letak dari kata tersebut tidak relevan

- D1 “send us your password”
- D2 “send us your review”
- D3 “review your password”
- D4 “review us”
- D5 “send your password”
- D6 “send us your account”

## Binary Bag-of-Words

Dalam representasi ini, urutan atau letak dari kata tersebut tidak relevan

dokumen	account	password	review	send	us	your
D1	0	1	0	1	1	1
D2	0	0	1	1	1	1
D3	0	1	1	0	0	1
D4	0	0	1	0	1	0
D5	0	1	0	1	0	1
D6	1	0	0	1	1	1

Apa yang menjadi masalah di sini?

- Matriksnya jarang
- Tidak ada informasi urutan
- Ada kata-kata yang sangat sering muncul



$$w_{t,d} = (1 + \log(tf_{t,d})) \log\left(\frac{N}{df_t}\right)$$

- $tf_{t,d}$  ... frekuensi kata  $t$  dalam dokumen  $d$ ,  $N$  ... jumlah dokumen,  $df_t$  ... jumlah dokumen yang mempunyai kata  $t$
- Kata yang sering muncul mungkin tidak penting, e.g. kata *hubung*
- Kata yang langka akan bernilai lebih – lihat posisi  $df_t$ !

# Menemukan Dokumen yang Mirip

- **Euclidean distance** adalah metode pengukuran jarak yang umum
- Untuk dokumen, jumlah kemunculan kata  *mungkin tidak begitu penting*
- Yang penting adalah keberadaan katanya → **cosine similarity**

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$$

- Dengan ide yang serupa dan beberapa tambahan algoritma lainnya, e.g. *Latent Semantic Analysis* (LSA), kita bisa menggunakan kakas ini untuk **tes seperti TOEFL**

- Dengan ide yang serupa dan beberapa tambahan algoritma lainnya, e.g. *Latent Semantic Analysis* (LSA), kita bisa menggunakan kakas ini untuk **tes seperti TOEFL**
- LSA berhasil menjawab 64.4% soal

- Dengan ide yang serupa dan beberapa tambahan algoritma lainnya, e.g. *Latent Semantic Analysis* (LSA), kita bisa menggunakan kakas ini untuk **tes seperti TOEFL**
- LSA berhasil menjawab 64.4% soal
- Pengguna bahasa Inggris non-natif rata-rata berhasil menjawab 64.5% soal

- Dengan ide yang serupa dan beberapa tambahan algoritma lainnya, e.g. *Latent Semantic Analysis* (LSA), kita bisa menggunakan kakas ini untuk **tes seperti TOEFL**
- LSA berhasil menjawab 64.4% soal
- Pengguna bahasa Inggris non-natif rata-rata berhasil menjawab 64.5% soal
- *Cukup untuk masuk banyak universitas di US!*

# Beberapa Tantangan

- Homograf, kata yang tulisannya sama tetapi maknanya berbeda, e.g. “beruang”

# Beberapa Tantangan

- Homograf, kata yang tulisannya sama tetapi maknanya berbeda, e.g. “**beruang**”
- Token yang tidak dikenali, e.g. **salah tik** (*typo*), neologisme, **slang**



# Beberapa Tantangan

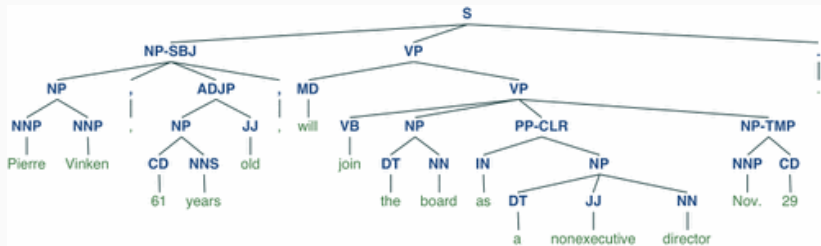
- Homograf, kata yang tulisannya sama tetapi maknanya berbeda, e.g. “**beruang**”
- Token yang tidak dikenali, e.g. **salah tik** (*typo*), neologisme, **slang**
- Kata dapat berubah makna dalam frasa, e.g. “**mahasiswa**” itu **netral**, tetapi “**harga mahasiswa**” itu positif

**NLTK**

---

“NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, **tokenization**, **stemming**, **tagging**, **parsing**, and **semantic reasoning**...”

# NER Tagging



**Figure 9:** Hasil NER tagging dengan NLTK [NLTK Project, 2017]

Beberapa korpus dan model yang terkenal dari NLTK:

- Project Gutenberg Selections
- Penn Treebank
- SentiWordNet
- Stopwords Corpus
- Porter Stemmer

## **Everything Data**

Document Similarity using NLTK and Scikit-Learn

Beberapa alternatif untuk tugas-tugas spesifik:

- **spaCy**: Industrial-Strength Natural Language Processing in Python
- **gensim**: topic modelling for humans

	SPACY	NLTK	CORENLP
Programming language	Python	Python	Java / Python
Neural network models	✓	✗	✓
Integrated word vectors	✓	✗	✗
Multi-language support	✓	✓	✓
Tokenization	✓	✓	✓
Part-of-speech tagging	✓	✓	✓
Sentence segmentation	✓	✓	✓
Dependency parsing	✓	✗	✓
Entity recognition	✓	✓	✓
Entity linking	✗	✗	✗
Coreference resolution	✗	✗	✓

**Figure 10:** Perbandingan fitur (Sumber: spaCy)



	SPACY	NLTK	ALLEN- NLP	STANFORD- NLP	TENSOR- FLOW
I'm a beginner and just getting started with NLP.	✓	✓	✗	✓	✗
I want to build an end-to-end production application.	✓	✗	✗	✗	✓
I want to try out different neural network architectures for NLP.	✗	✗	✓	✗	✓
I want to try the latest models with state-of-the-art accuracy.	✗	✗	✓	✓	✓
I want to train models from my own data.	✓	✓	✓	✓	✓
I want my application to be efficient on CPU.	✓	✓	✗	✗	✗

**Figure 11:** Mana yang harus dipakai? (Sumber: spaCy)

## Referensi

---

1. Bird, S., Edward L. & Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.
2. Jurafsky, D. & Martin, J. H. (2018). *Speech and Language Processing (Vol. 3)*. Pearson.
3. Manning, C., Raghavan, P., & Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

1. Stanford CS124: From Languages to Information
2. University of Edinburgh: Text Technologies for Data Science
3. Stanford CS276: Information Retrieval and Web Search  
(advanced)
4. Stanford CS224n: Natural Language Processing with Deep Learning (advanced)

- **Stanford NLP**: Chris Manning, Dan Jurafsky, Percy Liang
- **EdinburghNLP**: Sharon Goldwater, Mirella Lapata, Ivan Titov, Mark Steedman, Shay Cohen, Walid Magdy, etc.
- **UniMelb CIS School**: Tim Baldwin, Trevor Cohn, Karin Verspoor
- **UWNLP**: Noah Smith, Luke Zettlemoyer
- **QCRI**
- Chris Dyer (DeepMind), Sebastian Ruder (DeepMind), Hal Daumé III (UMaryland), Graham Neubig (CMU), Kyunghyun Cho (NYU), Phil Blunsom (Oxford), Richard Socher (Salesforce), Isabelle Augenstein (Copenhagen)

- ITB: Ayu Purwarianti, Masayu Leylia Khodra, Dessi Puji Lestari (speech)
- UI: Mirna Adriani, Rahmad Mahendra
- UGM: Yunita Sari
- Adhiguna Kuncoro (DeepMind), Dani Yogatama (DeepMind), Ruli Manurung (Google Japan), Clara Vania (NYU), Genta Indra Winata (HKUST), Samuel Cahyawijaya (HKUST), Kemal Kurniawan (UniMelb), Yudi Wibisono (UPI)



NLTK Project (2 Januari 2017)

## **Natural Language Toolkit**

<http://www.nltk.org/>



Richard Socher (diakses 15 Mei 2017)

## **CS224d: Deep Learning for Natural Language Processing**

<http://cs224d.stanford.edu/>

Terima kasih