

## Bagian IV

---

### Aplikasi dan Topik Tambahan



## Penerapan Pembelajaran Mesin

“Leading is not the same as being the leader. Being the leader means you hold the highest rank, either by earning it, good fortune or navigating internal politics. Leading, however, means that others willingly follow you – not because they have to, not because they are paid to, but because they want to.”

---

Simon Sinek

Bab ini memuat contoh penggunaan *machine learning* untuk dua permasalahan praktis yaitu: (1) sistem rekomendasi dan (2) sistem peringkasan dokumen. Dua domain ini dipilih karena tidak asing (*familiar*) bagi penulis. Seperti yang sudah dideskripsikan pada bab-bab sebelumnya, penerapan *machine learning* pada suatu domain membutuhkan pengetahuan/keahlian pada domain tersebut. Bab ini tidak akan membahas domain secara detail, tetapi secara abstrak (bertujuan memberikan gambaran/pengenalan). Untuk mengerti domain yang dibahas secara mendalam, silakan membaca sumber lainnya. Bab ini akan memuat secara sangat singkat, apa guna *machine learning* dan pada contoh kasus seperti apa teknik *machine learning* diterapkan pada suatu domain permasalahan. Tujuan bab ini adalah untuk memberikan gambaran, bahwa mengetahui *machine learning* saja mungkin tidak cukup. Sekali lagi kami ingin menekankan, pembaca harus mengerti domain aplikasi.

## 12.1 Sistem Rekomendasi

Bagian sistem rekomendasi ditulis oleh **Candy Olivia Mawalim** (*Japan Advanced Institute of Science and Technology*). Kami ingin berterima kasih atas sumbangsih yang diberikan.

Salah satu penerapan pembelajaran mesin adalah sistem rekomendasi. Sistem rekomendasi dimotivasi oleh keinginan pemilik usaha untuk meningkatkan penjualan dengan cara mengerti pola pembelian pengguna. Aplikasi yang memanfaatkan sistem rekomendasi dapat kita temukan dalam kehidupan sehari-hari, misalnya *Youtube* yang memberikan rekomendasi video berdasarkan riwayat video yang telah kita lihat sebelumnya dan *Amazon* yang merekomendasikan produknya dengan cara menawarkan produk yang sering dibeli pengguna lain yang memiliki karakteristik yang “mirip” dengan kita. Ada dua komponen penting pada sistem rekomendasi yaitu: pengguna dan *item*. Pengguna adalah sang pengguna sistem, sementara *item* dapat berupa video, buku, dan lain sebagainya (produk/layanan yang ditawarkan sistem).

Secara umum, terdapat dua teknik untuk membangun sistem rekomendasi yaitu: (1) *content-based filtering* dan (2) *collaborative filtering*. Teknik pertama berfokus pada karakteristik pengguna secara spesifik. Teknik kedua berfokus pada selera terhadap suatu *item*. Dalam sistem rekomendasi, teknik *machine learning* dapat digunakan untuk memprediksi *item* yang disukai pengguna. Dua subbab berikutnya memuat aplikasi teknik *machine learning* untuk kedua teknik sistem rekomendasi (12.1.1 dan 12.1.2).

### 12.1.1 Content-based Filtering

Teknik membangun sistem rekomendasi berdasarkan *content-based filtering* memanfaatkan informasi mengenai profil seorang pengguna beserta uraian item yang sangat menarik bagi pengguna tersebut [83]. Profil pengguna diartikan sebagai karakteristik (atribut dan *behavior*) yang dimiliki pengguna. Atribut pengguna misalnya *gender*, kewarganegaraan, umur, dan lain-lain. Informasi mengenai *behavior* mencakup karakteristik *item* yang seorang pengguna sukai. Misalkan *item* adalah film, karakteristik film dapat ditentukan dari aktor yang terlibat dalam film, pembuat film, tahun pembuatan film, dan *genre* dari film (misalnya *action*, *horror*, *comedy*). Dengan karakteristik ini, kita dapat menentukan kemiripan antar-film.

Kedua informasi ini dapat direpresentasikan sebagai vektor (ekuivalen dengan *feature vector*) agar mudah untuk dilakukan operasi aritmatika (dikenal dengan istilah ***user embedding*** dan ***item embedding***). Cara melakukan *embedding* untuk profil pengguna dan uraian item mirip dengan cara *word embedding* yang telah dijelaskan pada subbab 10.4.3).

Rekomendasi *item* yang diberikan pada pengguna adalah *item* yang paling mungkin disukai pengguna berdasarkan karakteristiknya. Agar mendapat gambaran lebih jelas, penulis mengajak pembaca untuk mengikuti tutorial

pembuatan sistem rekomendasi film sangat sederhana menggunakan dataset MovieLens <sup>1</sup> berukuran kecil (100K). Dataset ini memuat berbagai informasi sebagai berikut:

1. Informasi *rating* pengguna untuk masing-masing film.
2. Informasi film, misalkan berupa *genre* dan tanggal *release*.
3. Demografik pengguna, misal usia, *gender*, pekerjaan, dan lain lain.

Untuk menyederhanakan tutorial, kami hanya akan menjelaskan contoh penggunaan *genre* film untuk membuat sistem rekomendasi. Pertama-tama, kita bangun representasi *item embedding*, yaitu *genre* untuk masing-masing film (Tabel 12.1). Setiap baris pada tabel tersebut merepresentasikan *item embedding* untuk suatu film. Perhatikan! Setiap sel pada tabel diisi nilai “1” apabila film memiliki *genre* yang tertera pada kolom bersesuaian, “0” apabila tidak.

MovieId	Adventure	Animation	Children	Comedy
6	1	1	1	1
22	1	0	1	0
50	0	0	0	1

**Tabel 12.1.** Representasi *item embedding* untuk film berdasarkan *genre*.

Kedua, kita bangun representasi *user embedding*, yaitu apakah pengguna menyukai suatu film atau tidak (biner). Suka atau tidaknya pengguna terhadap suatu film dapat dilihat berdasarkan *rating* yang ia berikan. Sebagai contoh sederhana, kita anggap apabila pengguna menyukai suatu film apabila memberi nilai *rating* lebih dari atau sama dengan 4. Sebagai contoh, perhatikan Tabel 12.2. Apabila user menyukai suatu film, nilai “1” diisi pada kolom *Like or Not*, dan “0” apabila sebaliknya.

UserId	MovieId	Like or Not
1	6	0
1	22	0
1	50	1

**Tabel 12.2.** Representasi *user embedding* berdasarkan *rating* yang diberikan pengguna.

Berdasarkan *item embedding* dan *user embedding* yang kita punya, kita ganti kolom *MovieId* pada Tabel 12.2 menggunakan baris pada *item embedding*. Sekarang, kita memiliki dataset *behavior* pengguna dengan *UserId*

<sup>1</sup> <https://grouplens.org/datasets/movielens/>

= 1 (Tabel 12.3). Perhatikan! Tabel tersebut seperti dataset *machine learning* yang sudah kamu pelajari pada bab-bab sebelumnya. Diberikan *feature vector* dan kelas (Like or Not) yang berkorespondensi. Menggunakan data

MovieId	Adventure	Animation	Children	Comedy	Like or Not
6	1	1	1	1	0
22	1	0	1	0	0
50	0	0	0	1	1

**Tabel 12.3.** Dataset *behavior* pengguna dengan `UserId = 1`.

seperti pada Tabel 12.3, kita dapat menggunakan teknik *machine learning* untuk memprediksi apakah suatu pengguna akan menyukai film tertentu, berdasarkan *genre* yang dimuat oleh film tersebut.

Teknik *content-based filtering* memiliki keunggulan dimana kita tidak memerlukan banyak informasi tentang pengguna lain. Kita hanya memerlukan informasi uraian *item* dan informasi karakteristik suatu pengguna. Hal ini mengakibatkan rekomendasi yang diberikan sangat bergantung pada kepribadian pengguna. Apabila pengguna tidak konsisten, sistem rekomendasi juga bingung.

### 12.1.2 Collaborative Filtering

Teknik ini diperkenalkan oleh Paul Resnick dan Hal Varian pada 1997 [84]. Prinsip *collaborative filtering* adalah asumsi bahwa selera penggunaan terhadap suatu *item* cenderung sama dari waktu ke waktu [85]. Pada contoh kasus sederhana untuk sistem rekomendasi film, teknik ini memanfaatkan informasi *rating* dari banyak pengguna. Kita dapat merepresentasikan tingkah laku (*behaviour*) semua pengguna menggunakan matriks utilitas dimana baris merepresentasikan profil pengguna dan kolom merepresentasikan *item*. Sebagai contoh, perhatikanlah Tabel 12.4.

	$item_1$	$item_2$	...	$item_N$
$user_1$	2	3	...	4
$user_2$	5	3	...	1
...				
$user_3$	4	1	...	2

**Tabel 12.4.** Matriks Utilitas.

Ada dua metode varian *collaborative filtering* yaitu: (1) *neighborhood-based collaborative filtering* (*memory-based method*) dan (2) *model-based collabora-*

*tive filtering*. Method *neighborhood-based collaborative filtering* bekerja dengan fakta bahwa pengguna yang “mirip” memiliki pola yang “mirip” dalam memberikan *rating* untuk *item* [86] (pada Tabel 12.4, pengguna yang mirip memiliki baris yang mirip). Selain itu, *item* yang memiliki kemiripan, akan memiliki pola *rating* yang mirip (pada Tabel 12.4, *item* yang mirip memiliki kolom yang mirip). Dengan itu, kita dapat menggunakan perhitungan kemiripan vektor untuk menghitung pengguna mana yang mirip dengan suatu pengguna  $p$ . Saat memberikan rekomendasi film bagi pengguna  $p$ , kita tunjukkan film-film yang pernah ditonton pengguna yang mirip dengannya, atau kita tunjukkan film-film yang mirip dengan film-film yang pernah ditonton oleh pengguna  $p$ .

Untuk *model-based collaborative filtering*, prediksi dibangun dengan menggunakan teknik *machine learning* [86]. Untuk suatu sistem dengan data yang *sparse*, matriks utilitas seperti Tabel 12.4 tidaklah efisien secara memori. Kita dapat memanfaatkan teknik-teknik seperti *matrix factorization/principal component analysis* dan *autoencoder* untuk mengurangi ukuran matriks. Silakan membaca lebih lanjut materi-materi tersebut (ingat kembali materi bab 10).

## 12.2 Peringkasan Dokumen

Meringkas dokumen berarti mengerti keseluruhan isi teks/dokumen, kemudian mampu menyampaikan kembali **sebanyak/seakurat mungkin** maksud dokumen asli, ke dalam bentuk yang lebih singkat [87, 88, 49]. Suatu ringkasan harus lebih pendek dibanding dokumen asli. Dengan demikian, sulit untuk dikatakan bahwa suatu ringkasan dapat memuat keseluruhan isi dokumen. Karena itu, ringkasan hanya memuat **sebanyak/seakurat mungkin** maksud dokumen asli. Ada beberapa jenis peringkasan dokumen dilihat dari berbagai sudut pandang, misal:

1. Jumlah dokumen yang diringkas, meringkas satu dokumen (*single-document summarization*) [89] atau banyak dokumen menjadi satu ringkasan (*multi-document summarization*) [90].
2. Indikatif atau Informatif. Indikatif berarti menyediakan *pointer* ke bagian dokumen (misal membuat daftar isi). Informatif berarti menyampaikan sebanyak/seakurat mungkin maksud dokumen asli ke dalam bentuk lebih singkat. Pada masa sekarang, hampir seluruh riset peringkasan dokumen mengacu untuk menyediakan ringkasan yang informatif.
3. Domain. Hasil ringkasan bergantung pada domain dokumen, misalkan berita atau novel. Pada domain berita, hal yang penting untuk dimuat dalam ringkasan adalah 5W1H (*what, who, when, whom, where, how*) [91], sementara pada novel mungkin kita ingin tahu kejadian-kejadian yang ada (beserta urutannya).
4. Generik, *query-based*, atau *tailored*. Generik berarti menghasilkan ringkasan untuk umum, dalam artian tidak ada *target user* secara spesifik (contohnya adalah *review* buku) [87]. *Query-based* artinya menghasilkan ringkasan

dokumen berdasarkan *query* yang diberikan *user* (ringkasan adalah informasi yang dibutuhkan oleh *user* yang ditemukan di dokumen) [92]. *Tailored* berarti menyajikan informasi dengan tipe spesifik. Misalkan pada berita terdapat informasi 5W1H, kita hanya ingin mencari tahu informasi *how*.

5. Ekstraktif [93] atau abstraktif [49]. Ekstraktif berarti ringkasan hanya memuat unit informasi yang ada di dokumen asli. Analoginya seperti memilih potongan dari dokumen asli sebagai ringkasan (*copy*). Abstraktif berarti ringkasan dapat memuat unit informasi yang mungkin tidak ada di dokumen asli. Analoginya seperti mengerti dokumen kemudian menulis ulang (parafrase).

Hal terpenting untuk diingat adalah peringkasan dokumen (apa yang diringkaskan, ringkasan harus memuat apa, dsb) sangat bergantung pada **tujuan ringkasan** (siapa pembaca ringkasan, untuk apa ringkasan dihasilkan). Diktat ini akan membahas *framework* (kerangka kerja/berpikir) peringkasan dokumen dan bagaimana *machine learning* membantu peringkasan dokumen.

Secara umum, ada beberapa tahap pada proses peringkasan dokumen secara otomatis, terlepas dari tipe peringkasan dokumen yang dijabarkan [87, 94, 95, 88, 96, 91, 90, 89, 49, 97]:

1. *Representation* – Melakukan pemisahan dan representasi unit informasi; pada umumnya adalah kalimat, kata, atau frase. Dalam artian, kita menganggap dokumen tersusun atas sekuens kalimat, kata, atau frase. Kita potong-potong unit informasi pada dokumen lalu, unit informasi dapat direpresentasikan sebagai vektor untuk tiap unitnya [98]. Kita pun dapat menggali hubungan antara unit informasi (misal direpresentasikan sebagai graf) untuk kemudian mencari unit mana yang sentral [91, 99].
2. *Filtering* – Menyaring informasi. Unit manakah yang penting/tidak penting. Unit mana yang dibutuhkan/tidak dibutuhkan. Unit mana yang relevan/tidak relevan. Unit mana yang representatif/tidak representatif. Teknik menyaring informasi sangat bergantung pada representasi unit pada tahap sebelumnya (e.g., vektor atau graf). Secara singkat, tahap ini memilih unit informasi berdasarkan objektif.
3. *Generation* – Menghasilkan (*generate*) ringkasan. Bagian ini membutuhkan pengetahuan mengenai bidang pemrosesan bahasa alami (*natural language processing*) (NLP). Pada diktat ini, tidak akan dibahas.

Teknik *machine learning* berperan penting pada tahap 1 dan 2. Kami akan memberikan studi kasus pada subbab 12.2.1 agar lebih jelas. Sistem peringkasan dokumen yang membagi-bagi proses menjadi tahapan-tahapan tersebut disebut *pipelined approach*. Artinya kita melakukan proses diatas sebagai tahap-tahap berbeda (independen satu sama lain). Selain *pipelined approach*, kita juga dapat memandang peringkasan dokumen dengan *single-view approach* (subbab 12.2.2), artinya keseluruhan proses tersebut terjadi bersamaan.



### 12.2.1 Pipelined Approach

Subbab ini akan memuat cerita singkat tentang peringkasan *paper* [94, 97]. Berdasarkan teori *argumentative zoning* [94], *paper* terdiri dari zona-zona dengan tujuan komunikatif yang berbeda. Dengan bahasa yang lebih mudah, tiap kalimat (unit informasi) pada *paper* memiliki tujuan komunikasi yang berbeda. Misalnya ada kalimat yang menyatakan tujuan penelitian, latar belakang penelitian, atau hasil penelitian. Tujuan komunikasi kalimat disebut *rhetorical categories* [100].

Ringkasan, dalam bentuk abstrak atau judul *paper* memiliki pola [101, 97]. Dalam artian, suatu ringkasan bisa jadi hanya memuat informasi dengan *rhetorical categories* tertentu (*tailored summary*). *Machine learning* dapat digunakan untuk mengklasifikasikan kalimat (unit informasi) ke kelas masing-masing [100, 97]. Pertama-tama setiap kalimat direpresentasikan menjadi *feature vector* (ingat kembali materi bab 3 dan bab 4). Sebagai contoh, *paper* [97] merepresentasikan kalimat pada *paper* sebagai *feature vector* berdasarkan fitur-fitur sebagai berikut:

1. Posisi. Lokasi kalimat dibagi dengan panjang teks (numerik).
2. Kata kunci (*lexicon*). Apakah kalimat memuat kata kunci yang eksklusif untuk *rhetorical category* tertentu (biner – ya atau tidak).
3. Bobot. Jumlah bobot TF-IDF (ingat bab 10) kata pada kalimat (numerik).
4. Kategori sebelumnya. *Rhetorical category* kalimat sebelumnya (nominal).

Setelah diubah menjadi *feature vector*, teknik *machine learning* digunakan untuk mengklasifikasikan kalimat menjadi kelas *rhetorical categories* yang sesuai. Dengan demikian, kita dapat menyaring kalimat berdasarkan tipe informasi mana yang kita inginkan, berdasarkan tujuan peringkasan. Selanjutnya, teknik NLP digunakan untuk memproses informasi yang disaring untuk menghasilkan ringkasan. Materi tersebut diluar bahasan diktat ini.

### 12.2.2 Single-view Approach

Pada *pipelined approach*, setiap tahapan peringkasan dokumen (*representation*, *filtering*, dan *generation*) dianggap sebagai tahap yang independen satu sama lain. Permasalahannya adalah kesalahan pada suatu tahap akan mempengaruhi tahap berikutnya. Opsi lain adalah dengan melakukan keseluruhan proses sebagai satu tahapan yang utuh (*end-to-end*). Metode ini memiliki kaitan yang erat dengan teknik *machine translation*, yang pada masa sekarang populer didekati dengan teknik *deep learning* (*sequence to sequence encoder-decoder*) [49, 73, 72, 81, 102, 103]. Dokumen tersusun atas sejumlah  $N$  sekuens unit informasi  $\mathbf{u} = u_1, u_2, \dots, u_N$ . Sebuah ringkasan adalah  $M$  buah sekuens unit informasi  $\mathbf{r} = r_1, r_2, \dots, r_M$  dimana  $M < N$ . Tujuan peringkasan adalah untuk mencari  $\mathbf{r}$  terbaik, sedemikian sehingga dapat memenuhi persamaan 12.1.

$$\arg \max_{\mathbf{r}} P(\mathbf{r} \mid \mathbf{u}) \quad (12.1)$$

Pada umumnya, terdapat suatu variabel tambahan  $\theta$  yang mengendalikan (*govern*) probabilitas tersebut. Sehingga secara lebih tepat, persamaan 12.1 diubah menjadi persamaan 12.2.

$$\arg \max_{\mathbf{r}} P(\mathbf{r} \mid \mathbf{u}, \theta) \quad (12.2)$$

Perhatikan, persamaan 12.2 adalah bentuk yang dapat dimodelkan menggunakan *sequence to sequence encoder-decoder*. Selain peringkasan dokumen, banyak permasalahan pada bidang *natural language processing* memiliki bentuk yang “mirip”, seperti: *part-of-speech tagging* [104], *named entity recognition* [105], mesin translasi [73], dan rekonstruksi paragraf [106].

### 12.3 Konklusi

Teknik *machine learning* sangatlah berguna untuk berbagai macam permasalahan. Tetapi perlu dipahami bahwa teknik yang ada belum mampu memodelkan proses berpikir manusia dengan benar. Proses berpikir manusia sangatlah kompleks, dan model yang ada sekarang ini adalah bentuk simplifikasi. Sebagai contoh, seorang bayi sekali melihat anjing mungkin akan mengetahui anjing lainnya walaupun tidak memiliki rupa yang sama persis. Sementara itu, model *machine learning* harus diberikan banyak sampel. Manusia memiliki otak yang luar biasa hebat karena mampu belajar dengan sedikit contoh.

Persoalan-persoalan yang diberikan pada buku ini secara umum mencakup *supervised* dan *unsupervised learning* saja. Terdapat satu pemodelan masalah penting lainnya yaitu *reinforcement learning* dimana kita ingin memaksimalkan hasil untuk sekuens aksi (sekuens keputusan). Setiap keputusan dapat diberikan bobot yang berbeda. Kemudian, agen memaksimalkan nilai untuk sekuens keputusan (*cummulative reward*)<sup>2</sup>. Pada *supervised learning*, kita hanya perlu membuat satu keputusan saja (menggolongkan data ke kelas mana). Pada konferensi-konferensi, para *master* banyak menyebutkan masa depan *machine learning* berada pada *unsupervised* atau *reinforcement learning*. Aplikasi mutakhir yang ada sekarang ini didominasi oleh *supervised learning*. Pada kenyataannya, sebagian besar (hampir semua) data yang ada di dunia ini tidak berlabel. Karena itu, representasi data secara *unsupervised* menjadi topik riset hangat (*representation learning*).

Buku ini telah menjelaskan berbagai macam teknik, tetapi perlu diketahui bahwa materi yang diberikan adalah simplifikasi agar mampu dipahami secara intuitif. Selain itu, buku ini di desain sebagai materi pengantar saja. Untuk memahami teknik *machine learning* secara lebih jauh dan sempurna, penulis menyarankan untuk membaca buku referensi [8, 11].

<sup>2</sup> [https://en.wikipedia.org/wiki/Reinforcement\\_learning](https://en.wikipedia.org/wiki/Reinforcement_learning)

Materi yang disajikan pada buku ini adalah persoalan-persoalan *machine learning* dari sudut pandang optimisasi. Yaitu mencari himpunan (*set of*) parameter agar model pembelajaran yang dibangun mampu memberi keputusan yang optimal. Kami menyarankan pembaca untuk mencari referensi lebih jauh tentang *machine learning* dari sudut pandang eksplorasi, yaitu mencari *concept space* dimana model dapat bekerja dengan “baik”. Sebagai contoh, kita memiliki sebuah mesin (fisik) dengan berbagai macam konfigurasi (parameter). Apabila kita menjalankan mesin dengan konfigurasi parameter tertentu, mesin akan rusak. Tugas kita adalah mencari *parameters space* dimana mesin dapat berjalan. Sedangkan, pada persoalan optimisasi, kita mencari satu konfigurasi terbaik.

*Machine learning* adalah bidang yang sangat luas (lebih luas dari apa yang diceritakan pada buku ini) dan berkembang pesat. Penulis menyarankan pembaca untuk membaca makalah dari konferensi<sup>3</sup> *top-tier* untuk mengetahui perkembangan terkini. Sebagai contoh (diurutkan berdasarkan abjad)<sup>4</sup>:

- AAAI. AAAI Conference on Artificial Intelligence
- ACL. Annual Meeting of Association for Computational Linguistics
- CVPR. IEEE Conference on Computer Vision and Pattern Recognition
- EMNLP. Empirical Methods in Natural Language Processing
- ICCV. IEEE International Conference on Computer Vision
- ICLR. International Conference on Learning Representation
- ICML. International Conference on Machine Learning
- IJCAI. International Conference on Artificial Intelligence
- INTERSPEECH. Conference of the International Speech Association
- NIPS. Neural Information Processing System
- SIGIR. ACM Special Interest Group in Information Retrieval
- SIGKDD. ACM Special Interest Group in Knowledge Discovery and Data Mining

## 12.4 Saran Buku Lanjutan

Penulis ingin memberikan beberapa rekomendasi bacaan pembelajaran mesin selanjutnya (teoritis). Buku-buku berikut (diurutkan secara subjektif berdasarkan kualitas konten dan kemudahan dimengerti):

1. *Deep Learning* oleh Ian Goodfellow et al. [11]. Banyak yang menganggap buku ini sebagai “kitab” *deep learning* modern. Buku ini juga men-

<sup>3</sup> Kamu lebih mudah mengetahui informasi terkini dari konferensi dibanding jurnal karena proses *review* yang lebih cepat. Pada bidang keilmuan tertentu, konferensi lebih penting dibanding jurnal.

<sup>4</sup> Konferensi penting untuk bidang pemrosesan bahasa alami terdaftar pada <http://aclweb.org/anthology/> atau <https://aclanthology.coli.uni-saarland.de/>

cakup materi matematika dasar (e.g., aljabar linier) yang dibutuhkan untuk mengerti *deep learning*.

2. *Neural Network Methods in Natural Language Processing* oleh Goldberg [1]. Buku ini ditujukan sebagai bahan transisi bagi peneliti bidang pemrosesan bahasa alami untuk menggunakan metode *neural network*. Apabila kamu tidak tertarik dengan pemrosesan bahasa alami, kamu bisa membaca bab 1-5 pada buku ini sebagai pengenalan *neural network*.
3. *Pattern Recognition and Machine Learning* oleh Christopher M. Bishop [8]. Menurut penulis, banyak yang tahu buku ini karena dianggap sebagai “kitab”. Penjelasan buku ini sangat matematis dan relatif berat untuk dimengerti. Tetapi, kamu dapat menjadi *master* apabila memahami seluruh materi pada buku ini. Algoritma *machine learning* yang disajikan juga relatif lebih “klasik” dibanding rekomendasi pertama.
4. *Machine Learning* oleh Tom Mitchel [4]. Buku ini memuat materi *machine learning* yang cukup “klasik”. Buku ini kurang lebih cocok sebagai pengenalan, tapi relatif kurang dalam.

Sedikit tambahan pesan sponsor karena penulis berlatar belakang dari bidang pemrosesan bahasa alami, penulis menyarankan membaca buku-buku (teoritis) berikut sebagai dasar pengetahuan pemrosesan bahasa alami (diurutkan dari konten paling dasar):

1. *Foundations of Statistical Natural Language Processing* oleh Christopher D. Manning dan Hinrich Schutze [56]. Buku ini dapat dideskripsikan dengan satu kata, **TOP**. Buku ini memuat materi pemrosesan bahasa alami dengan sudut pandang matematis, dapat dianggap sebagai *framework* berpikir yang modern.
2. *Speech and Language Processing* oleh Daniel Jurafsky dan James H. Martin [12]. Buku ini jauh lebih tebal dari buku pertama dan memuat materi yang lebih luas. Penulis merekomendasikan buku ini untuk mengetahui permasalahan-permasalahan pemrosesan bahasa alami.
3. *An introduction to Information Retrieval* oleh Manning et al. [57]. Walaupun berjudul *information retrieval*, penulis pertama kali mengenal konsep *embedding* dari buku ini. Buku ini ditulis dengan apik.
4. *Neural Network Methods in Natural Language Processing* oleh Goldberg [1]. Apabila kamu telah membaca buku-buku yang disebutkan sebelumnya, kamu dapat membaca buku ini untuk transisi ke metode *neural network*.

Pada saat menulis buku ini, penulis berharap bisa menulis pada level di antara buku Tom Mitchel [4] dan Bishop [8] yaitu cukup matematis, lumayan mudah dipahami, dan lumayan dalam. Mudah-mudahan pembaca merasa tujuan ini tercapai dan menikmati membaca buku ini. Sebagai kalimat penutup, terimakasih sudah membaca sampai tahap ini.

## Soal Latihan

### 12.1. Cold Start Problem

- (a) Jelaskan apa itu *cold start problem* pada sistem rekomendasi, serta bagaimana cara menangani permasalahan tersebut!
- (b) Pada teknik sistem rekomendasi manakah (*content-based filtering* atau *collaborative filtering*) *cold start problem* mungkin muncul? Mengapa?

### 12.2. Eksplorasi Sistem Rekomendasi

Bangunlah suatu sistem rekomendasi dengan teknik *collaborative filtering* pada dataset MovieLens dengan memanfaatkan library *recommenderlab* <sup>5</sup>!

### 12.3. Peringkasan Dokumen

- (a) Presentasikanlah di kelasmu, *paper* sistem peringkasan dokumen otomatis oleh Kupiec et al. (1995) [98] <sup>6</sup> yang menggunakan *pipelined approach*!
- (b) Presentasikanlah di kelasmu, *paper* sistem peringkasan dokumen otomatis oleh Cheng and Lapata [103] <sup>7</sup> yang menggunakan *single-view approach*!
- (c) Jelaskan perbedaan, kelebihan, dan kelemahan pendekatan-pendekatan sistem peringkasan dokumen!

---

<sup>5</sup> <https://cran.r-project.org/web/packages/recommenderlab/index.html>

<sup>6</sup> <https://dl.acm.org/citation.cfm?id=215333>

<sup>7</sup> <http://www.aclweb.org/anthology/P16-1046>



---

## Referensi

1. Yoav Goldberg. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017.
2. Peter Linz. *An Introduction to Formal Language and Automata*. Jones and Bartlett Publishers, Inc., USA, 2006.
3. Ronald Brachman and Hector Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
4. Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.
5. Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.
6. Jonathan Gratch and Stacell Marsella. Computationally modelling human emotion. *Communications of the ACM*, 57(12):71–99, 2014.
7. Masayu Leylia Khodra and Dessi Puji Lestari. Odd semester lecture on machine learning. Lecture of Institute Teknologi Bandung, 2015.
8. Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
9. Sumio Watanabe and Hidetoshi Nishimori. Fall lecture note on statistical learning theory. Lecture note for Tokyo Institute of Technology, 2016.
10. Brian Caffo. *Statistical Inference for Data Science*. Lean Publishing, 2015.
11. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
12. Daniel Jurafsky and James H. Martin. *Speech and Language Processing Second Edition*. Prentice Hall, 2009.
13. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 1991.
14. Hidetoshi Nishimori. *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Clarendon Press, 2001.
15. Sharon L. Myres Ronald E. Walpole, Raymond H. Myers and Keying Ya. *Probability and Statistics for Engineers and Scientists*. Prentice Hall, 2012.
16. Gilbert Strang. *Linear algebra and its applications*. Thomson, Brooks/Cole, Belmont, CA, 2006.
17. Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.
18. Jeff Leek. *The Elements of Data Analytic Style*. Leanpub, 2015.

19. Takao Terano and Tsuyoshi Murata. Spring lecture on machine learning. Lecture of Tokyo Institute of Technology, 2017.
20. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
21. Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10, 2004.
22. Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.
23. J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108, 1979.
24. T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, September 2006.
25. Marti A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, July 1998.
26. John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization, 2010.
27. J. R. Quilan. *Discovering rules by induction from large collections of examples*. Edinburgh University Press, 1979.
28. J.R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986.
29. C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
30. Takao Terano and Tsuyoshi Murata. Spring lecture on machine learning. Lecture of Tokyo Institute of Technology, 2017.
31. L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSp Magazine*, 1986.
32. James Allen. *Natural Language Understanding*. Benjamin-Cummings Publishing Co., Inc., 1995.
33. George Karypis Michael Steinbach and Vipin Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, pages 525 – 526, 2000.
34. Omer I. E. Mohamed Fathi H. Saad and Rafa E. Al-Qutaish. Comparison of hierarchical agglomerative algorithms for clustering medical documents. *International Journal of Software Engineering and Applications (IJSEA)*, 3(3), 2012.
35. Rajeev Rastogi Sudipto Guha and Kyuseok Shim. Cure: An efficient clustering algorithm for large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 73 – 84, 1998.
36. Jack D. Cowan. Neural networks: The early days. In *Proceedings of Advances in Neural Information Processing Systems 2*, 1989.
37. Amir Atiya. *Learning Algorithms for Neural Network*. PhD thesis, California Institute of Technology, 1994.
38. Al. Cripps. Using artificial neural nets to predict academic performance. In *Proceedings of the 1996 ACM Symposium on Applied Computing*, pages 33–37, 1996.



39. Thomas Mikolov. *Statistical Language Models Based on Neural Networks*. PhD thesis, Brno University of Technology, 2012.
40. Kai Chen Greg Corrado Thomas Mikolov, Ilya Sutskever and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of CoRR*, 2013.
41. Gred Corrado Thomas Mikolov, Kai Chen and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of CoRR*, 2013.
42. Kai Yu. Large-scale deep learning at baidu. In *Proceedings of the 22<sup>nd</sup> ACM International Conference on Information and Knowledge Management*, pages 2211–2212, 2013.
43. M. A. Aizerman, E. A. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control*, number 25, pages 821–837, 1964.
44. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.
45. Marvin L. Minsky and Seymour A. Papert. *Perceptrons: Expanded Edition*. MIT Press, Cambridge, MA, USA, 1988.
46. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.
47. Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas, November 2016. Association for Computational Linguistics.
48. David Alvarez-Melis and Tommi Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
49. Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning and Representation (ICLR)*, 2015.
50. Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
51. Finale Doshi-Velez and Been Kim. A roadmap for a rigorous science of interpretability. In *ArXiv e-prints*.
52. Jeffrey L. Elman. Learning and development in neural networks: The importance of starting small. *Journal of Cognition*, (48):71–99, 1993.
53. Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, 2009. ACM.
54. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

55. Dzmitry Bahdanau Kyunghyun Cho, Bart van Merriënboer and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.
56. Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
57. Prabhakar Raghavan Christopher D. Manning and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge UP, 2009.
58. Andrew Y. Ng Richard Socher, Cliff Chiung-Yu Lin and Christopher D. Manning. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28<sup>th</sup> International Conference on Machine Learning*, 2011.
59. Jean Y. Wu Jason Chuang Richard Socher, Alex Perelygin and Christopher D. Manning. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Empirical Methods in Natural Language Processing*, 2013.
60. Erhc H. Huang Andrew Y. Ng Richard Socher, Jeffrey Pennington and Christoper D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Empirical Methods in Natural Language Processing*, 2011.
61. Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31<sup>st</sup> International Conference on Machine Learning*, 2014.
62. Richard Socher Jeffrey Pennington and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the Empirical Methods in Natural Language Processing*, pages 1532 – 1543, 2014.
63. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003.
64. Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.
65. Guoqiang Zhong, Li-Na Wang, Xiao Ling, and Junyu Dong. An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science*, 2(4):265 – 278, 2016.
66. Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, (37):141–188, 2010.
67. Jan Wira Gotama Putra and Takenobu Tokunaga. Evaluating text coherence based on semantic similarity graph. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 76–85, Vancouver, Canada, August 2017. Association for Computational Linguistics.
68. Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.
69. Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

70. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
71. Paul J. Werbos. Backpropagation through time: what does it do and how to do it. In *Proceedings of IEEE*, volume 78, pages 1550–1560, 1990.
72. Caglar Gulcehre Dzmitry Bahdanau Fethi Bougares HolgerSchwenk Kyunghyun Cho, Bart van Merrienboer and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.
73. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS’14*, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press.
74. Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. Character-based joint segmentation and pos tagging for chinese using bidirectional rnn-crf. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 173–183, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.
75. Tobias Horsmann and Torsten Zesch. Do lstms really work so well for pos tagging? – a replication study. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 727–736, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
76. Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016.
77. Ryohei Sasano Hiroya Takamura Yuta Kikuchi, Graham Neubig and Manabu Okumura. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas, November 2016. Association for Computational Linguistics.
78. Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, and aglar Gülehre and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*, 2016.
79. Yan-Kai Lin Cun-Chao Tu Yu Zhao Zhi-Yuan Liu Ayana, Shi-Qi Shen and Mao-Song Sun. Recent advances on neural headline generation. *Journal of Computer Science and Technology*, 32(4):768–784, Jul 2017.
80. Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas, November 2016. Association for Computational Linguistics.
81. Xiaojun Wan Jiwei Tan and Jianguo Xiao. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada, July 2017. Association for Computational Linguistics.

82. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.
83. Daniel Billsus and Michael J. Pazzani. The adaptive web. chapter Adaptive News Access, pages 550–570. Springer-Verlag, Berlin, Heidelberg, 2007.
84. Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, March 1997.
85. Daniar Asanov. Algorithms and methods in recommender systems. Berlin Institute of Technology, 2011.
86. Charu C. Aggrawal. *Recommender Systems: The Textbook*. Springer International Publishing Switzerland, 2016.
87. Eduard Hovy and Chin-Yew Lin. Automated text summarization and the summarist system. In *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, TIPSTER '98, pages 197–214, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.
88. Liang Zhou and Eduard Hovy. Template-filtered headline summarization. In *In the Proceedings of the ACL workshop, Text Summarization Branches Out*, pages 56–60, 2004.
89. Amin Mantrach Carlos A. Colmenares, Marina Litvak and Fabrizio Silvestri. Heads: Headline generation as sequence prediction using an abstract feature-rich space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 133–142, Denver, Colorado, May–June 2015. Association for Computational Linguistics.
90. Daniele Pighin Enrique Alfonseca and Guillermo Garrido. Heady: News headline abstraction through event pattern clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1243–1253, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.
91. Pierre-Etienne Genest and Guy Lapalme. Framework for abstractive summarization using text-to-text generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 64–73, Portland, Oregon, June 2011. Association for Computational Linguistics.
92. Shufeng Xiong and Donghong Ji. Query-focused multi-document summarization using hypergraph-based ranking. *Inf. Process. Manage.*, 52(4):670–681, July 2016.
93. David Zajic, Bonnie J. Dorr, and Richard Schwartz. Bbn/umd at duc-2004: Topiary. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Workshop on Document Understanding*, pages 112–119, 2004.
94. Simone Teufel and Marc Moens. Argumentative classification of extracted sentences as a first step towards flexible abstracting. In *Advances in automatic Text Summarization*, pages 155–171. MIT Press, 1999.
95. Bonnie J. Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: A parse-and-trim approach to headline generation. In Dragomir Radev and Simone Teufel, editors, *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 1–8, 2003.
96. Jurij Leskovec, Natasa Milic-Frayling, and Marko Grobelnik. Extracting summary sentences based on the document semantic graph. *Microsoft Research*, 2005.

97. Jan Wira Gotama Putra. Rhetorical sentence classification for automatic title generation in scientific article. *TELKOMNIKA*, 15(2):656–664, 2017.
98. Jan Pedersen Julian Kupiec and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 68–73, New York, NY, USA, 1995. ACM.
99. Hans-Martin Ramsel Daraksha Parveen and Michael Strube. Topical coherence for graph-based extractive summarization. In *Conference on Empirical Methods in Natural Language Processing*, pages 1949–1954. The Association for Computational Linguistics, 2015.
100. Simone Teufel and Marc Moens. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Comput. Linguist.*, 28(4):409–445, December 2002.
101. Diarmuid Ó Séaghdha and Simone Teufel. Unsupervised learning of rhetorical structure with un-topic models. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2–13, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.
102. Vibhu O. Mittal Michele Banko and Michael J. Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 318–325, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
103. Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. *CoRR*, abs/1603.07252, 2016.
104. Christopher D. Manning. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*, CICLing'11, pages 171–189, Berlin, Heidelberg, 2011. Springer-Verlag.
105. Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
106. Jiwei Li and Dan Jurafsky. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.



---

## Biografi Penulis

Jan **Wira** Gotama Putra adalah mahasiswa pascasarjana di Computational Linguistics/Natural Language Processing group (Tokunaga-Fujii-lab), Artificial Intelligence Major, Department of Computer Science, Tokyo Institute of Technology. Sebelumnya, penulis mendapatkan gelar sarjana di jurusan Teknik Informatika, Institut Teknologi Bandung. Penulis memiliki pengalaman menulis makalah ilmiah pada bidang pemrosesan bahasa alami (termasuk yang menggunakan teknik pembelajaran mesin) dan menerapkan teknik pembelajaran mesin untuk perusahaan IT. Penulis pernah mendapat *best paper award* untuk publikasi ilmiahnya, menjuarai *data mining hackathon*, serta memiliki prestasi akademik lainnya. Penulis masih terus mempelajari teknik *machine learning*, buku ini adalah catatan yang ingin ia bagikan.

<https://wiragotama.github.io/>