Ringkasan Buku 1

Nama Buku: Tensorflow in Actions
Team: Wilhelmina Arlene, Luthfiah Maulidya, Fadhilah Dwi Istiani

Chapter 10 — *Natural Language Processing with TensorFlow: Language Modeling*

🎯 **Tujuan Utama Chapter 10**
Chapter 10 berfokus pada **language modeling**, yaitu tugas inti di balik:

- text generation
- machine translation
- summarization
- dan semua model Transformer modern

Jika Chapter 9 menjawab:
"Apa sentimen dari teks?"
maka Chapter 10 menjawab:
"Kata apa yang paling mungkin muncul berikutnya?"

GAMBARAN BESAR LANGUAGE MODELING
Apa itu Language Modeling?
Language modeling adalah tugas:

- memodelkan **probabilitas urutan kata**

Secara formal:

$$P(w_1, w_2, \ldots, w_n)$$

atau dalam bentuk prediksi:

$$P(w_t \mid w_1, w_2, \ldots, w_{t-1})$$

Artinya:
Model belajar **pola bahasa**, bukan sekadar label.

STRUKTUR BESAR CHAPTER 10
Chapter ini dibagi menjadi **6 bagian utama**:

1. Pemrosesan data language modeling
2. Konsep dasar language modeling
3. Text generation dengan GRU
4. Evaluasi kualitas teks
5. Training & evaluasi model

6. Decoding strategies (Greedy & Beam Search)

10.1 Processing the Data

10.1.1 Apa Bedanya dengan Sentiment Analysis?

Pada sentiment analysis:

- satu teks → satu label

Pada language modeling:

- satu teks → **banyak pasangan input–target**

Contoh:

Kalimat:

"I love machine learning"

Dipecah menjadi:

- input: "I"
  - target: "love"
- input: "I love"
  - target: "machine"
- input: "I love machine"
  - target: "learning"

➡️ Model belajar **prediksi kata berikutnya**.

10.1.2 Masalah Vocabulary Besar

Tantangan besar:

- vocabulary terlalu besar
- training lambat
- memory boros

10.1.3 N-grams to the Rescue

Apa itu N-gram?

N-gram:

- potongan urutan kata sepanjang N

Contoh:

- unigram: "I"
- bigram: "I love"
- trigram: "I love ML"

Fungsi:

- membatasi konteks
- menurunkan kompleksitas

Menggunakan n-gram:

- untuk mengontrol ukuran input
- sebelum masuk ke neural network

10.1.4 Tokenizing Text

Proses:

1. lowercase
2. tokenization
3. mapping kata → integer
4. padding / truncation

Menggunakan:

- **TextVectorization**
- konsisten dengan Chapter 9

10.1.5 tf.data Pipeline

Language modeling membutuhkan:

- data sangat besar
- streaming efisien

Pipeline:

- generate (input, target) pairs
- batch
- shuffle
- prefetch

👉 Tanpa pipeline efisien, GPU akan idle.

10.2 GRUs in Wonderland: Generating Text

Kenapa GRU?

GRU (Gated Recurrent Unit):

- lebih sederhana dari LSTM
- lebih cepat
- performa sebanding

Dipilih karena:

- cocok untuk text generation
- efisien untuk eksperimen

Arsitektur Model Language Model

Struktur umum:

Input sequence
→ Embedding
→ GRU
→ Dense (softmax over vocabulary)

Output:

- probabilitas setiap kata dalam vocabulary

Cara Model Menghasilkan Teks

1. mulai dengan seed text
2. prediksi kata berikutnya
3. tambahkan ke input
4. ulangi

➡️ Proses autoregressive.

10.3 Measuring the Quality of Generated Text
Masalah Evaluasi NLP Generatif
Tidak ada "jawaban benar" tunggal.
Evaluasi kuantitatif sulit.

Perplexity
Perplexity adalah:
- metric standar language modeling
- mengukur seberapa "bingung" model

Interpretasi:
- perplexity rendah → model lebih baik
- perplexity tinggi → prediksi buruk

⚠️ Tapi:
- perplexity ≠ kualitas teks bagi manusia

10.4 Training and Evaluating the Language Model
Training Strategy
- loss: categorical cross-entropy
- optimizer: Adam
- training lama & berat

Evaluasi
Dilakukan dengan:
- perplexity
- sampling teks
- inspeksi manual

Human evaluation tetap penting.

10.5 Generating Text: Greedy Decoding
Apa itu Greedy Decoding?
Strategi paling sederhana:
- selalu pilih kata dengan probabilitas tertinggi

Kelebihan:
- cepat
- mudah

Kekurangan:

- teks monoton
- sering repetitif
- tidak kreatif

10.6 Beam Search: Better Decoding Strategy

Apa itu Beam Search?

Alih-alih satu pilihan:

- simpan **K kandidat terbaik**
- eksplor beberapa jalur

Proses:

1. mulai dengan K sequence
2. kembangkan semuanya
3. pilih K terbaik
4. ulangi

Trade-off Beam Search

- K besar → hasil lebih baik
- K besar → lebih lambat

Beam search:

- lebih natural
- lebih koheren
- **standar untuk NLP generatif klasik**