Chapter 16: Summary

Diedrick Darrell Darmadi - 1103223031

Bab 16 membahas bagaimana membangun sistem pemrosesan bahasa alami (NLP) menggunakan **RNN, LSTM/GRU, Encoder–Decoder**, dan yang paling penting, **mekanisme Attention**. Bab ini melanjutkan kemampuan pemrosesan sekuens dari bab sebelumnya, namun fokus pada tugas NLP yang lebih kompleks seperti *machine translation*, *text summarization*, dan *sequence-to-sequence learning*. Masalah utama dalam NLP adalah bahwa banyak tugas membutuhkan pemahaman konteks jangka panjang, yang sulit ditangani oleh RNN biasa. Oleh karena itu, bab ini memperkenalkan arsitektur yang mampu memahami konteks kalimat secara lebih efektif.

Pembahasan dimulai dengan **Sequence-to-Sequence (Seq2Seq) Architecture**, yang terdiri dari dua jaringan: **encoder** dan **decoder**. Encoder membaca seluruh input sequence (misalnya sebuah kalimat) dan mengubahnya menjadi satu *context vector* atau *thought vector*. Decoder kemudian menggunakan vector ini untuk menghasilkan output sequence, seperti menerjemahkan kalimat dari bahasa Inggris ke Prancis. Namun, pendekatan klasik ini menghadapi keterbatasan besar: semua informasi harus dimasukkan dalam satu vektor tetap. Pada kalimat panjang, kualitas terjemahan atau pemahaman model menurun drastis karena model sulit mengingat detail informasi awal. Masalah inilah yang kemudian mendorong lahirnya mekanisme **Attention**.

Bagian inti bab ini menjelaskan secara mendalam konsep **Attention Mechanism**, yang memungkinkan decoder "melihat kembali" bagian-bagian tertentu dari input sequence sesuai kebutuhan. Alih-alih hanya mengandalkan satu context vector, attention menghitung *alignment score* antara state decoder saat ini dengan setiap output encoder. Dengan cara ini, pada setiap langkah generasi token, model dapat fokus pada kata atau frasa tertentu yang relevan. Mekanisme ini membuat model jauh lebih efektif dalam menangani kalimat panjang, struktur kompleks, dan terjemahan antar-bahasa. Pembaca diperkenalkan pada jenis attention seperti **Luong attention** dan **Bahdanau attention**, beserta langkah matematisnya seperti scoring function, softmax weighting, dan context combination.

Setelah memahami konsep dasarnya, bab ini menunjukkan implementasi Attention di TensorFlow dan Keras. Model encoder–decoder dengan attention digunakan untuk berbagai tugas seperti *translation*, *headline generation*, dan *text summarization*. Dalam implementasinya, encoder dapat berupa LSTM atau GRU, dan decoder mengombinasikan hidden state dengan *attention context vector* pada setiap langkah time-step. Penggunaan attention tidak hanya meningkatkan akurasi, tetapi juga memberikan interpretabilitas: kita dapat melihat bagian mana dari input yang diperhatikan model, melalui *attention heatmaps*.

Bagian selanjutnya membahas **Word Embeddings**, termasuk penggunaan embedding yang dipelajari sendiri dan embedding yang sudah dilatih sebelumnya seperti **Word2Vec dan GloVe**. Representasi ini membuat model NLP lebih kaya secara semantik, karena kata-kata yang memiliki makna mirip akan memiliki vektor yang saling berdekatan dalam ruang embedding. Bab ini juga menjelaskan bagaimana melakukan *masking*, *padding*, dan *handling unknown tokens* dengan benar agar model bekerja stabil.

Bab ini juga menyentuh berbagai teknik penting dalam NLP seperti **beam search decoding** untuk menghasilkan output yang lebih baik dibanding greedy search, **teacher forcing** untuk mempercepat pelatihan decoder, serta strategi mengatasi overfitting pada model teks seperti recurrent dropout dan input dropout. Selain itu, dibahas bagaimana memuat dataset teks besar menggunakan tf.data dan bagaimana melakukan *tokenization* tingkat lanjut menggunakan subword units (misalnya Byte-Pair

Encoding).

Pada bagian akhir, bab ini memberikan perspektif bahwa meskipun RNN dengan Attention sangat kuat, pendekatan ini masih memiliki batasan dalam hal paralelisasi dan efisiensi. Hal ini membuka jalan bagi munculnya model berbasis **Transformer**, yang menghilangkan RNN sepenuhnya dan menggantinya dengan **self-attention** murni. Transformer kemudian menjadi standar industri dalam NLP modern. Bab ini menjadi transisi penting sebelum masuk ke topik model generatif dan Transformer pada bab-bab berikutnya.