BAB V

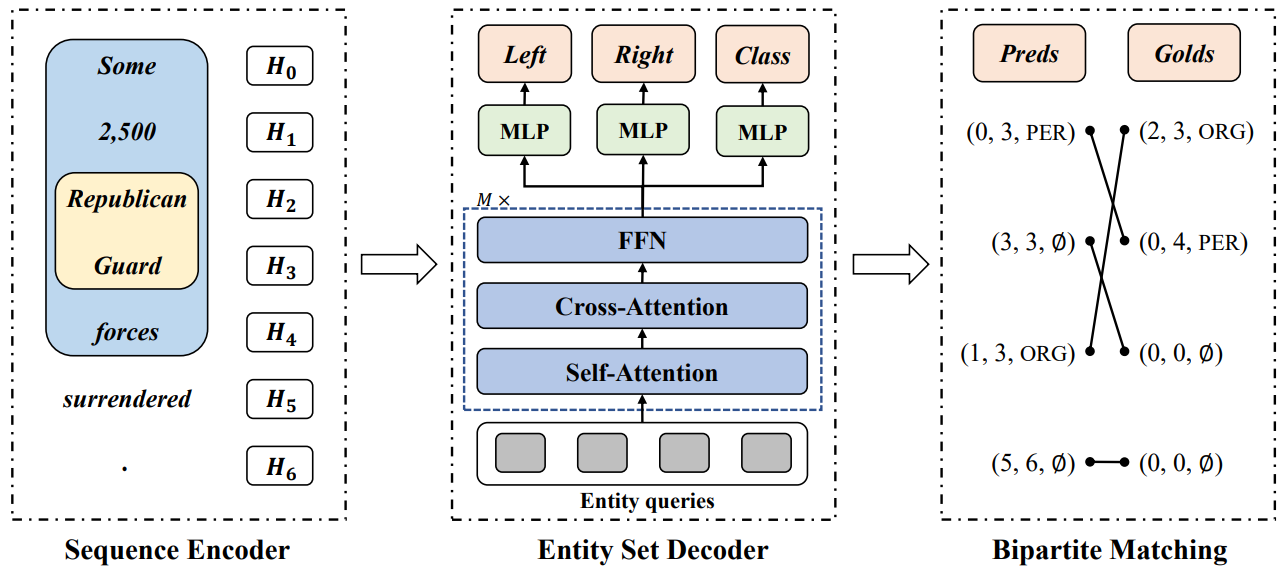
# SEQUENCE TO SET DALAM BAHASA INDONESIA

Sebelum memasuki uji coba tugas akhir ini, perlu diketahui seluruh isi dari arsitektur Sequence-To-Set Network secara detail. Sequence-To-Set Network memiliki tiga bagian besar dari arsitekturnya yaitu *encoder layer*, *decoder layer* dan *bipartite matching.* Bab ini akan dibagi menjadi tiga subbab juga, menjelaskan masing-masing tiga bagian besar itu, juga tiap subbab akan dibagi menjadi subabb kecil lagi sesuai dengan komponen/metode/fungsi yang digunakan untuk mencapai struktur akhir dari tiap subbab. Sebagai contoh, untuk encoder layer memiliki lima subbab kecil karena penggunaan empat jenis *word embedding* dimana tiap embedding memiliki cara tersendiri dan diperlukan penjelasan untuk tiap cara embedding tersebut. Namun, sebelum memasuki penjelasan tiap layer akan dijelaskan secara keseluruhan alur dari arsitektur sistem Sequence-To-Set Network.

## Sequence To Set Network

Sesuai pada gambar 5.1, alur dari Sequence-To-Set Network terdiri dari sequence encoder layer, entity set decoder layer dan bipartite matching dalam urutan yang telah disebut. Alur dari sistem ini adalah pertama kalimat input diterima oleh sequence encoder layer dalam bentuk struktur data yang telah dibahas sebelumnya mengenai pra proses dataset. Dimana output dari encoder ini memiliki tujuan untuk mendapatkan representasi dari kalimat tersebut untuk tiap kata dari kalimat tersebut dengan informasi dan konteks yang didapatkan.

Cara encoder mendapatkan informasi ini adalah dari beberapa metode word embedding yang telah ditentukan. Setiap dari embedding tersebut, seperti yang telah dijelaskan, akan dikenal lebih dalam pada subbab-subbab nya sendiri. Terdapat lima jenis word embedding yang disediakan, namun pada tugas akhir ini hanya digunakan empat (BERT, Word2Vec, Char LSTM, BiLSTM, POS Tag). Seluruh hasil embedding dari tiap metode akan kemudian digabung menjadi satu variabel dalam bentuk *vector*. Representasi kata terakhir ini kemudian akan dikenal sebagai *entity queries* dan menjadi output terakhir dari sequence encoder, juga menjadi input untuk bagian berikut yaitu set decoder layer.



Gambar 5.1  
Arsitektur Sistem Sequence-To-Set Network[[1]](#footnote-1)

Entity set decoder layer, mengambil entity queries dari sequence encoder yang telah memberikan nilai-nilai konteks dan informasi yang penting untuk tiap kata agar decoder mengerti kata apa yang perlu diperhatikan untuk diprediksikan. Cara untuk memperhatikan dan melakukan proses decoding ini adalah dengan bantuan attention. Pada bab teori penunjang telah menjelaskan sekilas mengenai self-attention dan cross-attention masing-masing. Untuk subbab decoder dibab ini akan menjelaskan lebih rinci terhadap alur, input, output tiap layer mengingat bahwa beberapa dari layer decoder memiliki cara kerja dan peran berbeda.

Dan bagian terakhir ada bagian yang menghitung nilai error dari keseluruhan training yang telah dilakukan encoder dan decoder Sequence-To-Set Network ini. Cara untuk menghitung nilai error dari perbandingan set cukup menyusahkan, karena itu Sequence-To-Set Network memilih bipartite matching sebagai metode penghitungan loss function untuk menghituing pencocokan optimal dari prediksi dan *golden entity*. Nilai optimal dari pencocokan set prediksi dengan golden entity didapatkan dengan bantuan dari Algoritma Hungarian.

### Sequence Encoder

Sequence encoder adalah proses/bagian/layer pertama yang berada di Sequence-To-Set Network. Sama dengan penjelasan sebelumnya, tujuan dari proses ini adalah untuk mendapatkan informasi dan konteks kata-kata dari kalimat input yang didapatkan. Seluruh subbab dari bagian ini adalah semua word embedding yang digunakan pada tugas akhir ini. Penjelasan untuk word embedding yang tidak digunakan (POS tag) tidak akan dijelaskan dalam bab ini. Pada akhir subbab ini akan dijelaskan entity queries, bentuk akhir yang dihasilkan dari proses encoder nanti.

#### BERT

Bidirectional Encoder Representations from Transformers (BERT) adalah metode yang telah diteliti dan *publish* oleh Google AI Language[[2]](#footnote-2). Metode BERT ini memiliki tujuan yaitu untuk menunjukkan pentingnya *bidirectional* pre-training dalam sebuah language model, karena pada saat itu metode Long Short Term Memory (LSTM) tidak dapat dinyatakan sebagai bidirectional (meskipun untuk BiLSTM juga tidak disebut mengambil informasi secara bidirectional. Karena ini, BERT dibuat untuk mendemonstrasikan hal tersebut. Secara struktur, BERT mengambil semua bagian dari encoder Transformers tanpa decodernya sekali pun. Memang Transformers mempunyai guna pada task yang paling cocok yaitu translasi bahasa. Namun untuk BERT, atau tumpukkan dari layer encoder dari Transformers, dapat memberikan solusi untuk beberapa pekerjaan seperti translasi, tanya jawaban, juga bisa analisa sentiment. Hal-hal ini hanya dapat dilakukan dengan BERT memahami informasi mengenai cara bahasa tersebut bekerja.

BERT akan dibagi menjadi dua tahap training, pretrain BERT untuk mengenal bahasa tersebut terlebih dahulu. Kemudian fine tune BERT untuk mengarahkan pemikiran solusi BERT tersebut kepada task yang ditujukan. Penjelasan berikut adalah mengenai proses yang dilewatkan untuk BERT.

Bagian pertama adalah pretraining, bagian ini mengajar model untuk mengenal bahasa apa yang dipelajari, apa itu bahasa dan apakah bentuk konteks dalam bahasa tersebut. Cara untuk mencapai pengertian tentang bahasa dan konteks dengan baik adalah dengan dua cara, Masked Model Language (MLM) dan Next Sentence Prediction (NSP). MLM adalah cara komputer belajar bentuk dari kalimat sebuah bahasa menggunakan token *mask* yang akan dimenggantikan beberapa kata dari suatu kalimat. Dari penutupan kata ini dan juga menebak kata apa yang sesuai, BERT dan memahami apa yang wajar untuk mengisi dengan cara melihat konteks juga dalam satu kalimat. Contoh yang dapat dimengerti untuk metode MLM adalah jika sebuah kalimat awalnya “The [MASK1] brown fox [MASK1] over the lazy dog. Kemudian cara satunya adalah NSP yang dapat dimengerti dari namanya sendiri, yaitu cara belajar komputer untuk mengerti kalimat yang dapat dihubungkan dengan kalimat lain. Hal ini juga dapat mengajar BERT untuk melihat konteks tiap kalimat untuk menyatukan mereka yang berhubungan. Contohnya, kalimat pertama adalah “Kampus ISTTS sangat ramai”. Kalimat keduanya, “Gedungnya sangat tinggi”. Dan kalau komputer yang sedang memelajari, artinya jawaban dari NSP yang benar seharusnya “benar kalimat pertama berhubungan dengan kalimat kedua”.

Setelah mengenal bentuk dari bahasa yang akan dipelajari, BERT perlu di fine tune agar sesuai dengan tujuan dari pembuatan model tersebut. Contoh dari penjelasan ini adalah untuk membuat model yang dapat melakukan tanya jawab dengan baik. Karena tujuan dari model harus menjawab dari sebuah pertanyaan, maka harus diberikan *supervised* dataset yang berisikan pertanyaan dan juga jawaban benar terhadap pertanyaan tersebut. Keunggulan BERT muncul dibagian ini. BERT yang telah pretrained telah mengenal bahasa tersebut, sehingga yang perlu diubah adalah beberapa parameter didalam BERT tersebut, dan juga jenis output dari BERT harus diubah sesuai dengan yang diinginkan (yaitu output kata-kata). Untuk menyimpulkan, BERT adalah metode yang dapat digunakan untuk memahami bahasa dan dengan cepat di fine tune untuk mendapatkan model yang bertujuan untuk suatu task dengan fine tune yang hanya mengubah beberapa dari isi BERT.

Bagian dari BERT masih belum selesai pass 2 dan pass 3

#### Word2Vec

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

#### Char LSTM

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

#### BiLSTM

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

#### Entity Queries

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

### Entity Set Decoder

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

#### Decoder Layer

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

##### Self – Attention

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

##### Cross – Attention

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

##### Feed Forward Network

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

#### Classification Layer

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

### Bipartite Matching

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

## Modifikasi Metode Sequence To Set Network

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity.

## Contoh Kasus Penggunaan Sequence To Set

Sebelum mencoba mengerti apa itu tugas pengenalan named entity (atau NER) secara mendalam, perlu diketahui terlebih dahulu apa yang dapat disebut sebagai sebuah named entity

1. Zeqi Tan, dkk, A Sequence-to-Set Network for Nested Named Entity Recognition, (2021). [↑](#footnote-ref-1)
2. Jacob Devlin, dkk, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, In Proceedings of NAACL 2019, pages 4171–4186, Minneapolis, Minnesota, June 2019. [↑](#footnote-ref-2)