Game Dialogue Text Generation With Bi-Directional LSTM & GRU

I Gusti Agung Premananda
21/473829/PA/20432
i.gusti0403@mail.ugm.ac.id

Muhammad Linggar Ryanidha
21/475209/PA/20548
muhammad.linggar2402@mail.ugm.ac.id

Shahran Kurnia Ramadhan
21/476650/PA/20592
shahran.kurnia1202@mail.ugm.ac.id

Daniel Ardi Chandra
21/479046/PA/20780
daniel.ardi.chandra@mail.ugm.ac.id

Abstract

Paper ini menjelaskan pendekatan yang digunakan untuk menghasilkan teks dialog yang realistis dan koheren dengan menggunakan RNN (Reccurant Neural Network) yaitu dengan model LSTM (Long Short-Term Memory) dengan arsitektur bi-directional. Fokus utama penelitian ini adalah pada penggunaan dataset dialog dari berbagai jenis game dan media hiburan untuk melatih model. Hasil dari eksperimen ini berhasil membentuk suatu model yang dapat melengkapi teks dialog yang dimasukkan oleh pengguna, namun tingkat koheren dari teks akan mengalami penurunan apabila diminta untuk melengkapi teks dialog yang panjang. Pelatihan model terhadap dataset yang lebih luas menjadi salah satu pertimbangan lanjutan dari eksperimen ini.

1 Text Generation

Teks generation adalah suatu bidang dalam pemrosesan bahasa alami (Natural Language Processing/NLP) yang berkaitan dengan kemampuan sistem komputer untuk menghasilkan teks secara otomatis. Problem ini melibatkan pembuatan model atau algoritma yang dapat menghasilkan urutan kata atau frasa yang memenuhi kriteria tertentu. Tantangan utama dalam problem ini adalah pemahaman dan penerapan struktur gramatikal, konteks, serta gaya bahasa yang konsisten dan sesuai dengan tujuan yang diinginkan. *Teks generation* memiliki aplikasi luas, mulai dari pembuatan konten otomatis, sistem chatbot, hingga pengembangan narasi dan dialog dalam *video game*. Peningkatan kualitas teks yang dihasilkan memerlukan pemahaman mendalam tentang representasi bahasa dan kemampuan model untuk mengantisipasi konteks dan maksud pengguna dengan akurat.

2 Dataset

Dalam eksperimen ini, kami memanfaatkan sebuah dataset yang terdiri dari dialog-dialog yang berasal dari beberapa *video game*. Data dialog ini telah diekstrak oleh anggota komunitas dari *video game* yang bersangkutan. Lebih spesifiknya, dataset yang kami gunakan adalah dialog dari *video game* "Portal 2," yang dikembangkan dan dirilis oleh Valve Software.

Rasio training-validasi-testing dataset kami secara berturut turut adalah 60-20-20 karena merupakan rasio yang umum digunakan.

Contoh sampel 3 baris dari dataset kami: "glados says this if you place a portal on the wall under a camera. to ensure the safe performance of all authorized activities, do not destroy vital testing apparatus. for your own safety, do not destroy vital testing apparatus."

3 Implementation

3.1 Pre-processing

3.1.1 Import dataset

Pertama, kami mengimport dataset dengan fungsi open() dalam bentuk .txt ke dalam program dengan encoding 'ISO-8859-1' ke dalam variable data.

3.1.2 Regex

Kemudian, kami melalukan data cleaning udengan bantuan REGEX dari library re Python untuk menghapus simbol pada data seperti @, [,",',dan].Hasilnya disimpan kedalam list bernama corpus.

3.1.3 Tokenization

Lalu, kami lakukan Tokenisasi untuk mengubah corpus menjadi angka yang bisa dipahami oleh model. Prosesnya dimulai dengan indexing setiap kata dalam corpus dalam sebuah dictionary index yang kemudian berfungsi untuk menerjemahkan setiap kalimat dalam corpus menjadi serangkaian angka dalam proses berikutnya, yaitu n-gram.

3.1.4 N-gram

Kami mendefiniskan sebuah fungsi dari scratch untuk menghasilkan array token dari data teks. Untuk

setiap baris dalam korpus, ini menghasilkan semua kemungkinan n-gram di mana n berkisar dari 2 hingga total kata dalam baris.

Tujuan kami menerapkan n-gram adalah untuk melatih model yang dapat memprediksi kata berikutnya secara berurutan, yang merupakan tugas umum dalam pemodelan bahasa dan dapat digunakan untuk tugas seperti text generation.

3.1.5 Padding

Terakhir, kami melakukan proses padding untuk memastikan bahwa jumlah kata per baris dalam suatu batch memiliki panjang yang sama. Banyak model pembelajaran mesin memerlukan data masukan yang memiliki bentuk yang konsisten, dan padding adalah cara untuk memenuhi persyaratan ini saat bekerja dengan data urutan, yang panjangnya dapat bervariasi.

Jika kita tidak menggunakan padding dan urutan kita memiliki panjang yang berbeda, kita mungkin mengalami kesalahan saat mencoba melatih model kita. Meskipun model kita dapat menangani rangkaian dengan panjang berbeda, melatihnya dengan cara ini mungkin akan kurang efisien, karena model harus memproses setiap rangkaian satu per satu, bukan memprosesnya secara batch. (1)

3.2 Model

Dalam waktu beberapa hari kami hanya dapat bereksperimen dengan 2 jenis model yaitu LSTM dan GRU.

Kami menggunakan library Tensorflow dengan model sequential. Secara lebih detail, layer yang kami gunakan adalah sebagai berikut:

1. Layer embedding dengan embedding_dim 256.

2. Layer Bidirectional LSTM atau GRU dengan jumlah *rnn_layer* dan *dropout* yang dispesifikasikan oleh hyperparameter

3. Layer Dense dengan neuron sebanyak *vocab size* atau jumlah kata unik.

3.2.1 Hyperparameter tuning

Kami melakukan hyperparameter tuning pada layer Bi-Directional LSTM untuk jumlah rnn_layer (128, 256, dan 512) dan juga persentasi dropoutnya (0.2, 0.3, dan 0.4). Dikarenakan masalah teknis dan waktu, kami memilih untuk melakukan hyperparameter tuning tanpa library yaitu dengan fungsi for-loop sederhana dan 5 epoch untuk satu kombinasi hyperparameter. Untuk hyperparameter tuning ini kami mendapat akurasi validasi terbaik sebesar 11%. dengan yaitu dengan hidden layer sebanyak 256, dan dropout 0.2.

3.2.2 Output & Kesimpulan

Setelah mendapatkan parameter terbaik tadi, kami menggunakannya untuk melatih model layer LSTM dan GRU dengan jumlah epoch yang jauh lebih besar, yaitu 50.

Kami menggunakan categorical cross entropy untuk mengukur loss dan metrik classification untuk mengukur akurasi. Model pertama, yaitu LSTM mendapatkan akurasi 11,6% baik pada test dataset yang di-merged dan atau yang hanya Portal 2. Sedangkan, jika menggunakan model GRU, kami mendapatkan akurasi sebesar 11,9% pada test dataset yang di-merged dan 11.7% pada test dataset yang hanya Portal 2. Dapat dilihat bahwa, meski hanya sepersekian persen, model GRU memiliki akurasi yang relatif lebih tinggi dibanding model LSTM.

Kesimpulannya, dalam evaluasi kami, kedua model LSTM dan GRU memiliki loss yang besar dan akurasi yang kecil pada set testing, sehingga kemampuan model kami terbilang masih lemah. Investigasi lebih lanjut dan penyempurnaan model mungkin diperlukan untuk meningkatkan kinerja tugas yang diberikan.

4 Pengembangan

Dikarenakan kurangnya kemampuan teknis tentang implementasi kode mengenai model transformer kami belum berhasil mengimplementasikan model Transformer. Kami sadar bahwa model kami jauh dari kata sempurna, karena masih banyak hal yang belum kami coba seperti menggunakan tokenizer GLOVE dan Word2Vec, dan jenis layer lainnya. Meski begitu, kami tetap mencoba semampu kami untuk menggunakan kemampuan yang kami miliki untuk menghasilkan output yang terbaik.

5 Ucapan Terima Kasih

Diucapkan terima kasih sebesar-besarnya kepada Ibu Yunita Sari karena selaku dosen mata kuliah Deep Learning, beliau telah memberikan feedback yang konstruktif dalam pengembangan pembuatan paper ini. Juga kepada semua pihak yang ikut berperan dalam proses pembuatan final project ini.

References

[1] Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing (3rd ed.). https://web.stanford.edu/ jurafsky/slp3/

[2] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press. https://nlp.stanford.edu/IR-book/

[3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org/

[4] Chollet, F. (2017). Deep Learning with Python. Manning Publications.

[5] Diederik P. Kingma & Jimmy Lei Ba (2015). Adam: a Method for Stochastic Optimization. https://arxiv.org/pdf/1412.6980.pdf

[6] Michael Phi (2015). Illustrated Guide to LSTM's and GRU's: A step by step explanation. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21