

Laporan Tugas Individu: Machine Translation dengan PyTorch

*Note: Sub-titles are not captured in Xplore and should not be used

Ahmad Mukhlis Farhan
Informatics Engineering
(of Affiliation)
University Of Darussalam Gontor
(of Affiliation)
Ponorogo, Indonesia
mukhfar02@gmail.com

Abstract—Tugas ini bertujuan untuk membangun dan membandingkan dua sistem penerjemah otomatis (Machine Translation), yaitu RNN dengan Mekanisme Attention dan Transformer, menggunakan framework PyTorch. Kedua model dilatih untuk menerjemahkan teks dari bahasa Inggris (EN) ke bahasa Indonesia (ID). Evaluasi performa dilakukan dengan menganalisis metrik loss pelatihan dan validasi.

Keywords—Machine Translation, Deep Learning, PyTorch, RNN, Attention, Transformer, Natural Language Processing, Ablation Study

I. INTRODUCTION (HEADING 1)

Machine Translation (MT) adalah salah satu aplikasi terpenting dalam Natural Language Processing (NLP) yang memungkinkan penerjemahan otomatis antarbahasa. Dalam tugas ini, kami mengeksplorasi dua arsitektur deep learning yang dominan untuk MT: Recurrent Neural Network (RNN) dengan mekanisme attention dan arsitektur Transformer. Tujuan utama adalah memahami implementasi kedua model, melatihnya pada pasangan bahasa Inggris-Indonesia, dan membandingkan kinerja serta karakteristiknya.

II. PERSIAPAN DATA

Data yang digunakan adalah pasangan kalimat bahasa Inggris-Indonesia. Tahapan persiapan data meliputi:

- **Pembersihan Teks:** Kalimat diubah menjadi huruf kecil, dinormalisasi dari Unicode ke ASCII, dan tanda baca dipisahkan. Karakter non-alfabet dibersihkan.
- **Pembagian Data:** Dataset dibagi menjadi data pelatihan (80%), validasi (10%), dan pengujian (10%).
- **Tokenisasi Subword:** Menggunakan SentencePiece dengan algoritma BPE (Byte Pair Encoding) untuk tokenisasi subword pada kedua bahasa. Ukuran vocabulary untuk bahasa Inggris dan Indonesia ditetapkan 8000. Token khusus seperti <pad>, <unk>, <bos>, dan <eos> ditambahkan dan dipetakan.
- **DataLoader PyTorch:** Data dikemas ke dalam objek TranslationDataset dan DataLoader untuk memfasilitasi pelatihan batch dan padding sekuens.

Jumlah pasangan kalimat yang dimuat adalah 14881, dengan 11904 untuk pelatihan, 1488 untuk validasi, dan 1489 untuk pengujian. Ukuran vocabulary yang dihasilkan untuk bahasa Inggris dan Indonesia masing-masing adalah 8000.

III. IMPLEMENTASI MODEL

A. Baseline: RNN dengan Mekanisme Attention

Model ini terdiri dari:

- **EncoderRNN:** Menggunakan GRU bidireksional untuk memproses sekuens input. Hidden state maju dan mundur terakhir digabungkan melalui lapisan linear untuk menghasilkan hidden state awal dekoder.
- **Attention:** Mekanisme attention lugh-style yang menghitung bobot relevansi antara hidden state dekoder saat ini dengan semua output encoder.
- **DecoderRNN:** Menggunakan GRU untuk menghasilkan sekuens output. Input ke GRU adalah kombinasi embedding token input saat ini dan weighted sum dari output encoder yang dihitung oleh mekanisme attention.

Model RNN memiliki total 31,173,952 parameter yang dapat dilatih.

B. Transformer

Arsitektur Transformer dibangun berdasarkan mekanisme self-attention dan tidak menggunakan RNN rekuren. Model ini terdiri dari:

- **TransformerEncoder:** Terdiri dari beberapa EncoderLayer. Setiap EncoderLayer memiliki lapisan multi-head self-attention dan position-wise feedforward network, diikuti oleh normalisasi lapisan dan dropout.
- **TransformerDecoder:** Terdiri dari beberapa DecoderLayer. Setiap DecoderLayer memiliki dua lapisan multi-head attention (satu untuk self-attention pada target dan satu untuk attention encoder-decoder), serta position-wise feedforward network.
- **Masking:** Masker dibuat untuk padding dan untuk mencegah decoder melihat token masa depan selama pelatihan.

Model Transformer memiliki total 10,156,864 parameter yang dapat dilatih.

IV. PELATIHAN DAN EVALUASI

Kedua model dilatih selama 10 epoch menggunakan optimizer Adam dan fungsi loss CrossEntropyLoss, mengabaikan indeks padding. Gradient clipping diterapkan untuk mencegah exploding gradients.

A. Hasil Pelatihan RNN+Attention

Epoch	Waktu (m:s)	Loss Latihan	PPL Latihan	Loss Validasi	PPL Validasi
01	0m 27s	2.946	19.030	3.272	26.376
02	0m 27s	2.052	7.783	3.193	24.364
03	0m 27s	1.583	4.869	3.256	25.940
04	0m 27s	1.341	3.821	3.393	29.743
05	0m 27s	1.258	3.518	3.492	32.859
06	0m 27s	1.222	3.393	3.535	34.296
07	0m 27s	1.189	3.284	3.568	35.457
08	0m 27s	1.168	3.216	3.585	36.069
09	0m 27s	1.157	3.181	3.595	36.425
10	0m 27s	1.154	3.170	3.604	36.753

Setelah 10 *epoch*, model RNN mencapai *training loss* 1.154 dan *validation loss* 3.604. Perplexity (PPL) validasi adalah 36.753.

B. Hasil Pelatihan Transformer (Ukuran Vocab 8000)

Epoch	Waktu (m:s)	Loss Latihan	PPL Latihan	Loss Validasi	PPL Validasi
01	0m 5s	8.248	3820.928	6.861	954.720
02	0m 5s	5.176	177.009	5.151	172.583
03	0m 5s	4.103	60.528	4.474	87.729
04	0m 5s	3.619	37.300	4.090	59.728
05	0m 5s	3.447	31.408	3.882	48.528
06	0m 5s	3.328	27.876	3.791	44.295
07	0m 5s	3.303	27.198	3.743	42.228
08	0m 5s	3.299	27.080	3.728	41.590
09	0m 5s	3.315	27.525	3.722	41.381
10	0m 5s	3.322	27.708	3.722	41.360

Setelah 10 *epoch*, model Transformer (dengan *vocab size* 8000) mencapai *training loss* 3.322 dan *validation loss* 3.722. Perplexity (PPL) validasi adalah 41.360.

V. ABLATION STUDY: PENGARUH UKURAN VOCABULARY PADA TRANSFORMER

A. Transformer (Ukuran Vocab 4000)

Epoch	Waktu (m:s)	Loss Latihan	PPL Latihan	Loss Validasi	PPL Validasi
01	0m 4s	5.426	227.240	4.908	135.297
02	0m 4s	4.299	73.613	4.549	94.577
03	0m 4s	3.901	49.444	4.394	80.970
04	0m 4s	3.746	42.365	4.298	73.541

05 0m 4s 3.708 40.767 4.288 72.824

Model ini dilatih selama 5 *epoch* dengan *vocab size* 4000. *Validation loss* akhir adalah 4.288 dengan PPL validasi 72.824.

B. Transformer (Ukuran Vocab 14000)

Epoch	Waktu (m:s)	Loss Latihan	PPL Latihan	Loss Validasi	PPL Validasi
01	0m 5s	8.528	5057.494	7.151	1275.523
02	0m 5s	5.250	190.589	4.892	133.178
03	0m 5s	4.598	99.318	4.507	90.681
04	0m 5s	4.241	69.483	4.285	72.632
05	0m 6s	4.241	69.483	4.285	72.632

Model ini juga dilatih selama 5 *epoch* dengan *vocab size* 14000. *Validation loss* akhir adalah 4.285 dengan PPL validasi 72.632.

VI. KESIMPULAN DAN DISKUSI

Dari hasil pelatihan dan ablation study, dapat dilihat bahwa:

- Efisiensi Transformer: Model Transformer secara konsisten menunjukkan waktu pelatihan per epoch yang jauh lebih cepat (sekitar 4-6 detik) dibandingkan RNN+Attention (sekitar 27 detik), meskipun dengan jumlah parameter yang lebih sedikit. Ini menyoroti efisiensi komputasi dari arsitektur Transformer.
- Dampak Ukuran Vocabulary:
- Vocab 8000 (10 epoch): PPL validasi akhir 41.360.
- Vocab 4000 (5 epoch): PPL validasi akhir 72.824.
- Vocab 14000 (5 epoch): PPL validasi akhir 72.632.
- Dari ablation study, tampaknya model Transformer pada vocab size 4000 dan 14000 (setelah 5 epoch) memiliki PPL validasi yang jauh lebih tinggi dibandingkan dengan Transformer pada vocab size 8000 (setelah 10 epoch). Ini menunjukkan bahwa ukuran vocabulary 8000 memberikan keseimbangan yang lebih baik atau bahwa pelatihan 10 epoch untuk vocab size 8000 lebih efektif. Peningkatan vocab size menjadi 14000 tanpa fine-tuning atau pelatihan lebih lama mungkin tidak langsung menghasilkan performa yang lebih baik dalam jumlah epoch yang sama. Model dengan vocabulary yang lebih kecil (4000) menunjukkan performa yang sebanding dengan vocabulary yang lebih besar (14000) dalam 5 epoch, yang mungkin mengindikasikan bahwa vocabulary yang terlalu besar atau terlalu kecil bisa menjadi tantangan tanpa penyesuaian hyperparameter yang tepat.
- Performa Awal (PPL): Model Transformer cenderung memiliki PPL yang sangat tinggi di awal pelatihan, menunjukkan bahwa ia mungkin memerlukan strategi learning rate schedule (seperti

warm-up) atau epoch pelatihan yang lebih banyak untuk konvergensi optimal, terutama dengan vocabulary yang lebih besar.

Secara keseluruhan, meskipun Transformer menunjukkan keunggulan dalam kecepatan dan efisiensi, pemilihan hyperparameter seperti ukuran vocabulary dan durasi pelatihan sangat krusial untuk mencapai performa terbaik.

VII. SARAN UNTUK PENELITIAN LANJUTAN

- **Hyperparameter Tuning Lanjutan:** Lakukan tuning hyperparameter yang lebih sistematis untuk setiap ukuran vocabulary (misalnya, learning rate, jumlah layer, head attention, dropout) untuk menemukan konfigurasi optimal.
- **Evaluasi dengan Metrik Standard:** Hitung metrik evaluasi standar seperti SacreBLEU dan chrF pada test set untuk semua konfigurasi model (termasuk hasil ablation study) untuk memberikan perbandingan performa terjemahan yang lebih objektif dan detail.
- **Analisis Kualitatif Lanjutan:** Periksa lebih banyak contoh terjemahan dari setiap model dan konfigurasi vocabulary untuk memahami jenis kesalahan yang terjadi dan bagaimana ukuran vocabulary memengaruhi kualitas terjemahan.
- **Durasi Pelatihan yang Seragam:** Pastikan semua model dilatih untuk jumlah epoch yang sama atau

sampai terjadi konvergensi untuk memastikan perbandingan yang adil.

- **Integrasi Teknik Decoding Canggih:** Terapkan teknik decoding seperti beam search untuk meningkatkan kualitas terjemahan yang dihasilkan oleh model.

ACKNOWLEDGMENT (Heading 5)

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak/Ibu Dosen Pembelajaran Mesin 2 atas bimbingan, arahan, dan ilmu yang telah diberikan selama pengerjaan tugas ini. Dukungan dan masukan yang konstruktif sangat membantu dalam penyelesaian laporan ini.

REFERENCES

- [1] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, arXiv:1312.6114. [Online]. Available: <https://arxiv.org/abs/1312.6114>
- [2] K. Eves and J. Valasek, "Adaptive control for singularly perturbed systems examples," Code Ocean, Aug. 2023. [Online]. Available: <https://codeocean.com/capsule/4989235/tree>
- [3] S. Liu, "Wi-Fi Energy Detection Testbed (12MTC)," 2023, gitHub repository. [Online]. Available: <https://github.com/liustone99/Wi-Fi-Energy-Detection-Testbed-12MTC>
- [4] "Treatment episode data set: discharges (TEDS-D): concatenated, 2006 to 2009." U.S. Department of Health and Human Services, Substance Abuse and Mental Health Administration, Office of Applied Studies, August, 2013, DOI:10.3886/ICPSR30122.v2
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems* 30 (NIPS 2017).