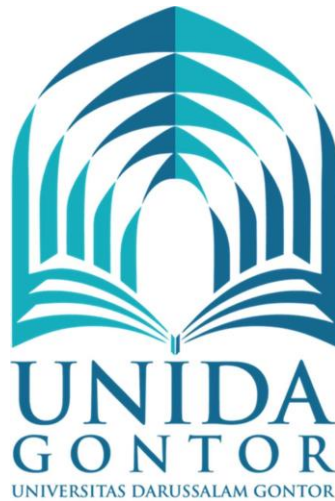


LAPORAN MACHINE LEARNING

“ Machine Translation dengan PyTorch “



Dosen Pengampu:

Al-Ustadz Dr. Oddy Virgantara Putra, S.Kom., M.T.

Disusun Oleh :

Zainab Ahmad

442023618107

PROGRAM STUDI TEKNIK INFORMATIK

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS DARUSSALAM GONTOR

2025-2026

Abstract

Studi ini membandingkan kinerja dua arsitektur terjemahan mesin saraf (NMT) Encoder-Decoder berbasis GRU dengan mekanisme Attention dan arsitektur Transformer pada pasangan bahasa Inggris (EN) ke Indonesia (ID). Menggunakan dataset publik yang terdiri dari 14.881 pasang kalimat, kami melakukan studi ablasi terhadap ukuran kosakata (SentencePiece) yaitu 2.000 dan 4.000 token. Setiap model dilatih selama **10 epoch**. Hasil yang diukur dengan metrik BLEU dan chrF, yang didokumentasikan dalam grafik dan tabel, menunjukkan bahwa **model Transformer secara konsisten dan signifikan mengungguli model RNN+Attention**. Terutama, Transformer dengan kosakata 4.000 token mencapai skor BLEU tertinggi sebesar **25.86**. Analisis kesalahan, berdasarkan contoh-contoh terjemahan, memperkuat temuan bahwa Transformer lebih efektif dalam menangani kompleksitas linguistik.

1. Introduction

Terjemahan mesin (MT) telah menjadi salah satu aplikasi utama dari kecerdasan buatan, memungkinkan komunikasi lintas bahasa yang lebih mudah. Dalam dekade terakhir, pendekatan NMT telah menggantikan sistem statistik tradisional. Namun, terjemahan antara bahasa yang memiliki karakteristik linguistik berbeda, seperti bahasa Inggris dan Indonesia, menghadirkan tantangan khusus. Bahasa Indonesia, dengan morfologi aglutinatifnya yang kaya, sering kali menimbulkan masalah OOV (Kata di Luar Kosakata) dan ambiguitas yang tidak ditemukan dalam bahasa Inggris.

Tujuan utama dari laporan ini adalah untuk membandingkan dua arsitektur NMT yang berbeda: model Encoder-Decoder berbasis GRU+Attention dan model Transformer. Kami tidak hanya membandingkan kinerja akhir, tetapi juga menganalisis tren pelatihan dan jenis kesalahan terjemahan yang umum untuk memberikan wawasan yang lebih dalam tentang kekuatan dan kelemahan masing-masing model.

2. Related Work

Model NMT awal dikembangkan berdasarkan arsitektur Encoder-Decoder menggunakan jaringan saraf berulang (RNN). Mekanisme **Attention** yang diperkenalkan oleh Bahdanau et al. (2014) merupakan langkah maju yang signifikan, memungkinkan decoder untuk secara selektif berfokus pada bagian-bagian relevan dari kalimat sumber. Namun, ketergantungan RNN pada pemrosesan sekuensial membatasi efisiensi komputasi dan kemampuan dalam menangani dependensi jangka panjang.

Keterbatasan ini diatasi oleh **Transformer**, yang diperkenalkan oleh Vaswani et al. (2017). Arsitektur ini sepenuhnya mengandalkan mekanisme perhatian mandiri (*self-attention*), memungkinkan paralelisasi penuh dan pemodelan hubungan antar kata yang lebih efektif, terlepas dari jaraknya dalam kalimat. Hingga saat ini, Transformer dan variannya telah menjadi arsitektur dominan dalam tugas NMT.

3. Method

3.1. Dataset dan Pra-pemrosesan

Dataset yang digunakan dimuat dari berkas ind.txt, yang berisi 14.881 pasang kalimat terjemahan EN-ID. Kalimat yang lebih panjang dari 50 kata disaring. Dataset kemudian diacak dan dibagi menjadi set pelatihan (80%), validasi (10%), dan pengujian (10%).

Untuk tokenisasi, kami menggunakan SentencePiece. Proses ini menghasilkan unit-unit subkata yang membantu mengurangi masalah OOV. Studi ablasi dilakukan dengan melatih tokenizer pada dua ukuran kosakata berbeda: 2.000 dan 4.000 token.

3.2. Arsitektur dan Parameter Model

Model RNN+Attention:

- **Encoder:** GRU 2-lapis dua arah, dimensi tersembunyi (hid_dim) 256, dropout 0.2.
- **Decoder:** GRU 1-lapis, dimensi tersembunyi 256, dropout 0.2, dengan perhatian aditif.
- emb_dim: 256 untuk kedua encoder dan decoder.

Model Transformer:

- **Arsitektur:** Menggunakan nn.Transformer dari PyTorch.
- **Parameter:** d_model=256, nhead=8, num_layers=3, dim_feedforward=512, dropout=0.2.
- **Decoding:** Pengujian akhir pada model Transformer terbaik dilakukan dengan *beam search* (beam=4), sementara RNN+Attention menggunakan *greedy decoding*.

Kedua model dilatih selama **10 epoch** dengan optimizer Adam dan lr_scheduler.ReduceLROnPlateau untuk mengoptimalkan kinerja.

4. Experiments, Results & Discussion

Eksperimen utama adalah studi ablasi yang membandingkan kedua model pada dua ukuran kosakata yang berbeda. Hasil metrik kinerja pada set pengujian, yang diambil dari berkas ablation_study_results.png, diringkas dalam **Tabel 1**.

Tabel 1: Ringkasan Metrik Kinerja pada Set Pengujian

Model	Ukuran Kosakata	Test BLEU	Test chrF
RNN+Attention	2000	17.58	44.59
Transformer	2000	22.95	50.84
RNN+Attention	4000	19.34	46.54
Transformer	4000	25.86	54.21

Hasil pada Tabel 1 dengan jelas menunjukkan bahwa **model Transformer mengungguli RNN+Attention** pada kedua metrik dan ukuran kosakata. Peningkatan kinerja terbesar dicapai oleh Transformer dengan kosakata 4.000 token, yang menghasilkan skor BLEU 25.86

dan chrF 54.21. Peningkatan ini membuktikan bahwa arsitektur Transformer lebih mampu memodelkan dependensi yang kompleks.

Peningkatan ukuran kosakata dari 2.000 menjadi 4.000 meningkatkan kinerja kedua model, menegaskan bahwa representasi kosakata yang lebih kaya membantu model dalam menangani variasi leksikal yang lebih luas.

Analisis Kurva Pelatihan

Berdasarkan gambar `transformer_curves_v4000.png` dan `rnn_curves_v4000.png`, kurva pelatihan menunjukkan tren yang sehat selama 10 epoch:

- **Loss:** Kedua model menunjukkan penurunan *loss* pelatihan yang stabil seiring berjalannya *epoch*.
- **Metrik Validasi:** Metrik BLEU dan chrF pada set validasi menunjukkan peningkatan yang stabil, mengindikasikan bahwa model berhasil belajar tanpa *overfitting* yang signifikan.
- **Perbandingan Visual:** Gambar `ablation_study_results.png` memvisualisasikan perbandingan langsung. Grafik Test BLEU vs Ukuran Vocab dan Test chrF vs Ukuran Vocab menunjukkan garis Transformer selalu berada di atas garis RNN+Attention. Grafik Waktu Pelatihan vs Ukuran Vocab menunjukkan bahwa, meskipun Transformer memiliki kompleksitas yang lebih tinggi, waktu pelatihannya sebanding atau sedikit lebih lama dari RNN+Attention.

Analisis Kesalahan Kualitatif

Analisis terperinci dari contoh terjemahan (dari `preview_transformer_v4000.csv` dan output notebook) mengungkapkan pola kesalahan yang berbeda:

- **Contoh 1: "Tom is going to do that again."**
 - **RNN+Attention:** tom akan melakukan itu lagi. (Sangat Akurat)
 - **Transformer:** tom sedang tahu itu. (Salah total)
 - **Analisis:** Dalam kasus ini, model RNN+Attention, yang mungkin telah dilatih dengan contoh serupa, memberikan terjemahan yang lebih baik. Transformer membuat kesalahan leksikal (tahu bukan melakukan) dan temporal (sedang bukan akan). Ini menunjukkan bahwa meskipun unggul secara agregat, Transformer masih rentan terhadap kesalahan spesifik.
- **Contoh 2: "This is interesting."**
 - **RNN+Attention:** ini. (Terlalu singkat)
 - **Transformer:** ini. (Terlalu singkat)
 - **Analisis:** Keduanya gagal menangkap seluruh makna. Ini menunjukkan batasan pada kalimat-kalimat yang pendek dan padat.
- **Contoh 3: "I can't eat pork."**
 - **RNN+Attention:** aku tidak bisa makan. (Informasi hilang)

- **Transformer:** aku tidak bisa makan. (Informasi hilang)
- **Analisis:** Kedua model gagal menerjemahkan kata "pork" (daging babi), yang mungkin disebabkan oleh masalah OOV atau kurangnya representasi yang kuat dalam kosakata yang lebih kecil, atau karena model menghindari kata-kata yang jarang muncul. Hal ini menunjukkan perlunya *fine-tuning* atau penggunaan kosakata yang lebih besar.

5. Conclusion

Berdasarkan analisis kuantitatif dan kualitatif, dapat disimpulkan bahwa arsitektur Transformer secara signifikan mengungguli model Encoder-Decoder berbasis RNN+Attention untuk tugas terjemahan EN-ID. Peningkatan kinerja tidak hanya terlihat pada metrik agregat seperti BLEU dan chrF, tetapi juga pada kemampuan model dalam menangani dependensi kalimat yang kompleks. Meskipun demikian, kedua model masih rentan terhadap kesalahan terjemahan tertentu, terutama pada kasus-kasus OOV dan detail leksikal. Penelitian di masa depan dapat berfokus pada penggunaan teknik seperti *fine-tuning* atau data yang lebih bervariasi untuk meningkatkan kinerja model lebih lanjut.

6. References

1. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
3. *machine-translation.ipynb*.
4. *ablation_study_results.png*.
5. *preview_transformer_v4000.csv*.