

Analisis Perbandingan Kinerja Model-Model Deep Learning dalam Identifikasi Sarkasme pada Judul Berita

Diki Wahyudi^{a,1,*}

^aProgram Studi Statistika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia

¹diki.wahyudi11@ui.ac.id*

*Corresponding author

ARTICLE INFO

NIM 2106709131

Keywords:

Judul Berita

Sarkasme

Recurrent Neural Network (RNN)

Gated Recurrent Unit (GRU)

Long Short-Term Memory (LSTM)

Convolutional Neural Network (CNN)

ABSTRACT

Di era konsumsi digital yang tinggi, kemampuan mendeteksi sarkasme dalam judul berita menjadi hal penting untuk masyarakat kuasai agar mampu membedakan berita yang salah dengan maksimal. Oleh karena itu, dibutuhkan suatu model yang dapat mempelajari pola-pola judul berita yang sarkas agar dapat membantu memudahkan pekerjaan manusia dalam meng-klasifikasikan judul berita tersebut. Dengan berkembangnya teknologi dan ilmu pengetahuan, berbagai model-model telah dikembangkan untuk melakukan klasifikasi pada data teks, salah satunya model deep learning. Deep learning (DL) telah muncul sebagai bagian dari machine learning (ML) dan artificial intelligence (AI), mengungguli metode ML tradisional, terutama dalam menangani dataset yang tidak terstruktur dan besar. Pada penelitian ini, digunakan dataset sarkasme yang dikumpulkan dari dua situs berita, yaitu The Onion dan HuffPost. Enam model deep learning, seperti Simple Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Bidirectional LSTM, Gated Recurrent Unit (GRU), Bidirectional GRU, dan Convolutional Neural Network (CNN), digunakan dalam mendeteksi sarkasme dengan evaluasi kinerja *accuracy*, *precision*, *recall*, dan *F1-score*. Dataset dibagi menjadi data *train* dan data *test* dengan rasio 0,8 banding 0,2. Dari hasil *fitting* model, didapatkan bahwa model Bidirectional LSTM dan CNN menghasilkan *accuracy* dan *precision* yang tinggi dibandingkan model-model lainnya. Model Bidirectional LSTM dan CNN menjadi model yang terbaik untuk identifikasi sarkasme pada data ini karena kinerjanya yang bagus.

Copyright © 2024.

All rights reserved.

I. Introduction

Judul berita memiliki peran penting dalam menarik perhatian pembaca dan menyampaikan pesan secara singkat dan padat. Di era di mana konsumsi media digital berada pada titik tertingginya, kemampuan membedakan sarkasme dalam judul berita menjadi semakin penting. Pendeteksian sarkasme dalam judul berita tidak hanya penting untuk memahami sentimen yang mendasarinya, tetapi juga untuk mencegah penyebaran informasi yang salah dan memastikan keakuratan penyebaran berita. Untuk mengatasi tantangan ini, penelitian mengenai identifikasi judul berita yang memuat sarkasme menjadi semakin relevan, mengingat perannya yang sangat krusial dalam penyebaran informasi dan persepsi masyarakat terhadap suatu topik.

Saat ini, para peneliti telah banyak beralih ke model artificial intelligence, machine learning, atau deep learning untuk memanfaatkan kemampuan algoritma model tersebut dalam mengekstrak pola dan nuansa yang kompleks dari data teks. Makalah ini menyajikan analisis perbandingan model deep learning untuk mengidentifikasi sarkasme dalam judul berita. Dengan mengevaluasi kinerja berbagai arsitektur deep learning, seperti Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), dan Convolutional Neural Network (CNN), penelitian ini bertujuan untuk memberikan wawasan tentang keefektifan model dengan pendekatan yang berbeda dalam mendeteksi sarkasme pada judul berita.

Penelitian sebelumnya di bidang deteksi sarkasme banyak berfokus pada platform media sosial dan forum *online*, khususnya Twitter (sekarang X), dengan fokus yang terbatas terhadap judul berita. Karakteristik judul berita yang berbeda, seperti narasi yang singkat dan gaya jurnalistik, menghadirkan tantangan unik bagi model deteksi sarkasme. Oleh karena itu, penelitian ini mengisi kesenjangan dalam literatur yang ada dengan secara khusus menargetkan identifikasi sarkasme pada judul berita.

Analisis perbandingan model deep learning akan dilakukan terhadap data judul berita [1] yang diberi keterangan sarkasme atau bukan merupakan sarkasme. Metrik kinerja klasifikasi, seperti *accuracy*, *precision*, *recall*, dan *F1-score* akan digunakan untuk mengevaluasi efektivitas setiap model deep learning. Dengan membandingkan kinerja berbagai arsitektur model deep learning secara sistematis, penelitian ini bertujuan untuk memberikan wawasan tentang teknik “otomatis” dalam mendeteksi sarkasme pada judul berita. Penelitian ini diharapkan dapat berkontribusi terhadap kemajuan teknologi Natural Language Processing (NLP) dan memiliki implikasi terhadap peningkatan keandalan dan kredibilitas media berita di era digital.

II. Literature Review

Penelitian terbaru dalam deteksi dan klasifikasi sarkasme telah mendapatkan perhatian yang masif karena penerapan praktisnya dalam analisis sentimen, natural language processing, dan pemantauan media sosial. Berbagai pendekatan telah diusulkan untuk mengatasi tantangan yang ditimbulkan oleh nuansa sifat sarkasme dalam data teks. Misalnya, A. Rajadesingan *et al.* [2] memperkenalkan metode berbasis machine learning yang memanfaatkan fitur linguistik dan analisis sentimen untuk mendeteksi sarkasme dalam data Twitter. Studi mereka menunjukkan hasil yang bagus dalam mengidentifikasi *tweet* sarkastis secara akurat. Selain itu, model FastText dengan CNN dan LSTM [3] dan model LSTM berbasis perhatian/*attention* [4] juga telah menunjukkan efektivitas dalam menangkap isyarat kontekstual yang penting untuk identifikasi sarkasme.

Penelitian sebelumnya mengenai deteksi sarkasme berfokus pada penggunaan dataset Twitter. Berikut ini merupakan rangkuman beberapa penelitian yang telah dilakukan untuk identifikasi sarkasme pada judul berita.

Table 1. Identifikasi Sarkasme pada Judul Berita dalam Penelitian Sebelumnya

Penulis	Dataset	Classifiers	Akurasi
P. Shrikhande, V. Setty, dan A. Sahan [5]	26,7 ribu judul dengan 11,7 ribu judul sarkastis dan 14,9 ribu judul nonsarkastis	CNN-LSTM	86%
		Bi-LSTM	86,13%
R. Misra dan P. Arora [6]	26.709 judul diambil, di mana 11.725 judul sarkastis dan 14.984 judul nonsarkastis	CNN-LSTM	89,70%
R. Ali <i>et al.</i> [7]	26.709 judul berita, mencakup 14.985 judul berita netral dari HuffPost dan 11.724 judul berita sarkastis dari The Onion	Decision Tree	74,65%
		Random Forest	80,82%
		Multinomial	83,52%
		Bernoulli	84,25%
		Support Vector Machine	84,22%
		LSTM	64,88%
		Proposed GMP-LSTM	92,5486%

Studi-studi ini secara kolektif berkontribusi pada upaya berkelanjutan dalam mengembangkan metode yang kuat dan efektif untuk deteksi dan klasifikasi sarkasme, serta menawarkan wawasan dan metodologi yang berharga untuk penelitian lebih lanjut dalam domain ini.

III. Data and Methods

A. Data

Dataset yang digunakan pada penelitian ini merupakan dataset judul berita untuk deteksi sarkasme yang dikumpulkan dari dua situs berita, yaitu The Onion dan HuffPost. Situs The Onion digunakan untuk memproduksi judul berita versi sarkastis dari kategori berita singkat dan berita dalam foto (yang bersifat sarkastis). Selain itu, judul berita nyata (dan nonsarkastis) juga diambil dari situs HuffPost. Data diakses dari Kaggle dengan link sebagai berikut. <https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection>

B. Methods

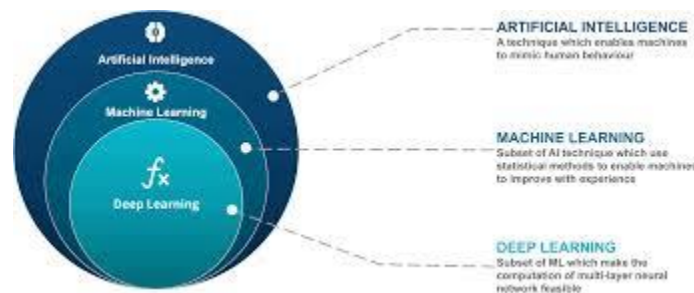


Fig. 1. Hubungan Antara Artificial Intelligence, Machine Learning, dan Deep Learning. Sumber: [8]

1. Deep Learning

Deep learning mewakili bagian dari teknik machine learning yang berfokus pada pembelajaran representasi data melalui penggunaan jaringan saraf dengan banyak lapisan. Meskipun deep learning berada di bawah payung artificial intelligence (AI), deep learning berbeda dari pendekatan machine learning tradisional dalam beberapa aspek utama.

Tidak seperti algoritma machine learning tradisional yang sering kali memerlukan fitur buatan untuk diekstrak dari data mentah, model deep learning dapat secara otomatis mempelajari representasi hierarki fitur langsung dari data. Kemampuan untuk mempelajari fitur secara otomatis telah menjadi faktor penting dalam keberhasilan deep learning, khususnya dalam domain seperti computer vision dan natural language processing.

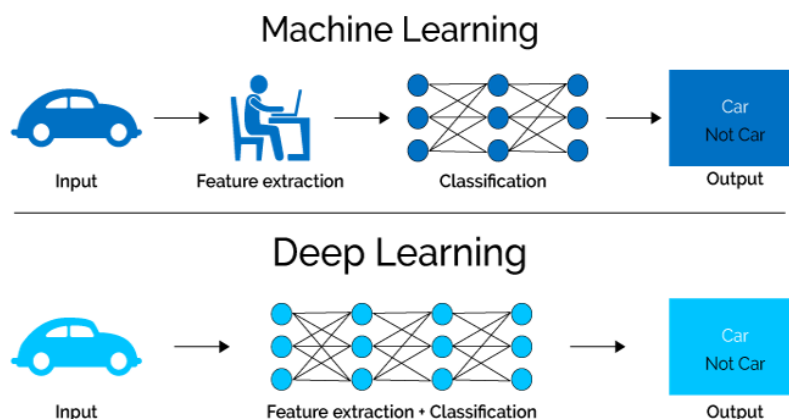


Fig. 2. Ilustrasi Visual Perbedaan antara Deep Learning dan Machine Learning Traditional dalam Hal Ekstraksi Fitur dan Pembelajarannya. Sumber: [9]

2. Simpel Recurrent Neural Network (RNN)

RNN adalah jenis jaringan saraf tiruan yang dirancang untuk bekerja dengan data urutan seperti teks atau runtun waktu. Setiap simpul dalam RNN memiliki siklus yang memungkinkan informasi untuk memiliki sifat memori atau “ingatan” terhadap informasi yang telah diproses sebelumnya. Namun, RNN sering mengalami masalah menghilangnya gradien saat memproses urutan yang panjang.

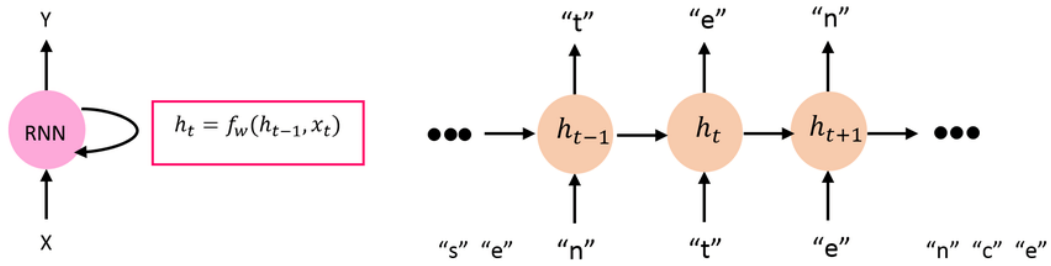


Fig. 3. Arsitektur Sempel RNN. Sumber: [10]

Fig. 3. menggambarkan simple RNN, di mana memori internal ($h(t)$) dihitung menggunakan persamaan

$$h_{t+1} = g(Wx_t + Uh_t + b) \quad (1)$$

di mana $g(\cdot)$ mewakili fungsi aktivas (biasanya garis singgung hiperbolik), U dan W dapat disesuaikan matriks bobot yang dapat disesuaikan untuk *hidde state* (h), b adalah suku bias, dan x menunjukkan vektor masukan.

RNN telah terbukti menjadi model pemrosesan data sekuensial *powerfull*, yang memanfaatkan kemampuannya untuk menangkap ketergantungan dari waktu ke waktu. Berbagai jenis model RNN, seperti LSTM, Bidirectional LSTM, GRU, dan Bidirectional GRU, telah dikembangkan untuk mengatasi tantangan spesifik di bidang ini pada aplikasi yang berbeda [11].

3. Long Short-Term Memory (LSTM)

LSTM adalah jenis RNN yang dirancang untuk mengatasi masalah menghilangnya gradien dalam RNN. Dengan menggunakan struktur gerbang, LSTM dapat mengontrol aliran informasi dalam jangka panjang dan jangka pendek, sehingga memungkinkan pemrosesan urutan yang lebih lama.

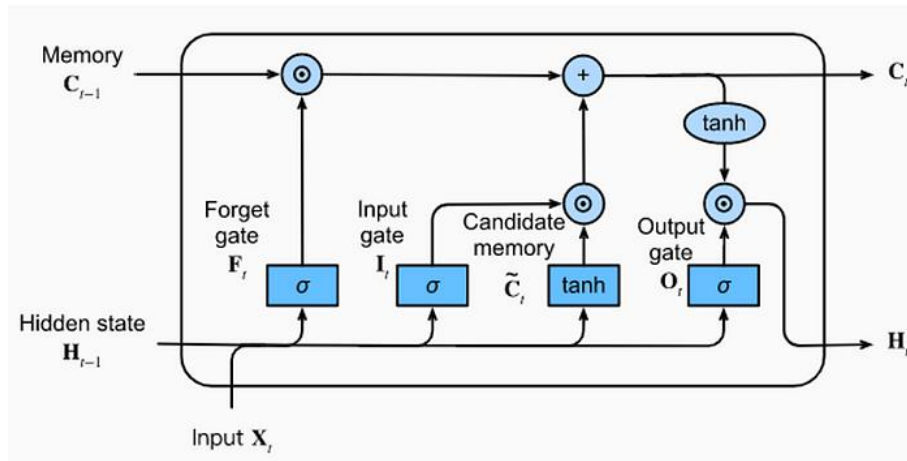


Fig. 4. Arsitektur LSTM. Sumber: [12]

Persamaan pembaruan untuk unit LSTM dinyatakan dengan persamaan berikut.

$$h^{(t)} = g_o^{(t)} f_h(s^{(t)})$$

$$s^{(t-1)} = g_f^{(t)} s^{(t-1)} + g_i^{(t)} f_s(wh^{(t-1)}) + uX^{(t)} + b$$

$$\begin{aligned} g_i^{(t)} &= \text{sigmoid}(w_i h^{(t-1)} + u_i X^{(t)} + b_i) \\ g_f^{(t)} &= \text{sigmoid}(w_f h^{(t-1)} + u_f X^{(t)} + b_f) \\ g_o^{(t)} &= \text{sigmoid}(w_o h^{(t-1)} + u_o X^{(t)} + b_o) \end{aligned} \quad (2)$$

di mana f_h dan f_s mewakili fungsi aktivasi dari keadaan sistem dan keadaan internal, biasanya memanfaatkan fungsi tangen hiperbolik. Operasi *gating*, dilambangkan sebagai g , adalah jaringan saraf *feedforward* dengan fungsi aktivasi sigmoid, memastikan nilai keluaran dalam rentang $[0, 1]$, yang diartikan sebagai sekumpulan bobot. Subskrip i, o , dan f masing-masing berhubungan dengan *input gate*, *output gate*, dan *forget gate* [11].

4. Bidirectional Long Short-Term Memory (Bi-LSTM)

Bi-LSTM adalah varian dari LSTM di mana dua arsitektur LSTM dihubungkan bersama secara sekuensial, satu berjalan maju dan yang lainnya berjalan mundur. Dengan demikian, Bi-LSTM dapat menangkap konteks dari kedua arah dalam data urutan/sekuensial.

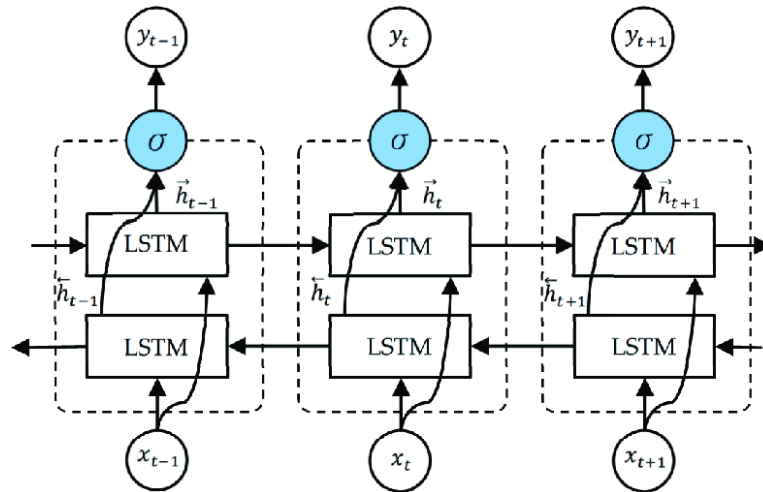


Fig. 5. Arsitektur Bi-LSTM. Sumber: [13]

5. Gated Recurrent Unit (GRU)

GRU adalah alternatif lain dari LSTM yang menggunakan struktur yang lebih sederhana. Dibandingkan dengan LSTM, GRU memiliki jumlah gerbang yang lebih sedikit, sehingga mengurangi jumlah parameter dan mempercepat proses pelatihan.

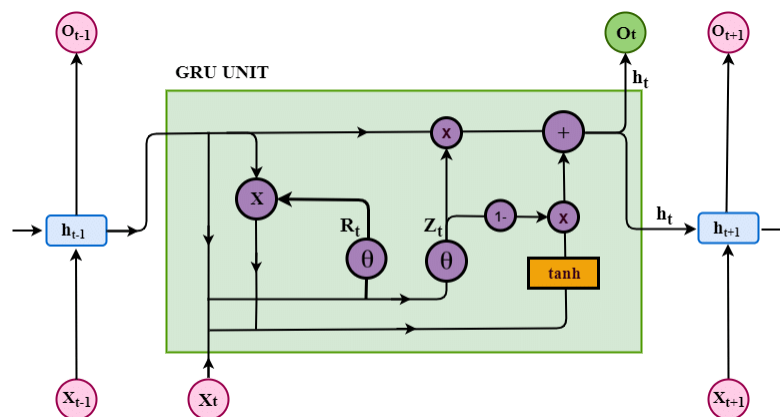


Fig. 6. Arsitektur GRU. Sumber: [14]

Update gate (persamaan 3) menentukan berapa banyak informasi masa lalu harus dipertahankan dan digabungkan dengan masukan saat ini pada langkah waktu tertentu. Informasi tersebut dihitung berdasarkan rangkaian *hidden state* sebelumnya h_{t-1} dan masukan saat ini x_t , diikuti dengan transformasi linier dan fungsi aktivasi sigmoid.

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \quad (3)$$

Reset gate (persamaan 4) menentukan seberapa banyak informasi masa lalu yang harus dilupakan. Informasi tersebut dihitung dengan cara yang serupa dengan *update gate* menggunakan rangkaian *hidden state* sebelumnya dan masukan saat ini.

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad (4)$$

Current memory content (persamaan 5) dihitung berdasarkan *reset gate* dan rangkaian transformasi *hidden state* sebelumnya dan masukan saat ini. Hasil tersebut melewati fungsi aktivasi tangen hiperbolik ke menghasilkan kandidat aktivasi.

$$\tilde{h}_t = \tanh(W_h[r_t h_{t-1}, x_t]) \quad (5)$$

Terakhir, *final memory state* h_t ditentukan oleh kombinasi *hidden state* sebelumnya dan kandidat *candidate activation* (persamaan 6). *Update gate* menentukan keseimbangan antara *hidden state* sebelumnya dan *e candidate activation*. Selain itu, *output gate* o_t dapat diperkenalkan untuk mengontrol aliran informasi dari memori saat ini konten ke output (persamaan 7). *Output gate* dihitung menggunakan *current memory state* h_t dan biasanya diikuti oleh fungsi aktivasi, seperti fungsi sigmoid.

$$h_t = (1 - z_t)h_{t-1} + z_t \tilde{h}_t \quad (6)$$

$$o_t = \sigma_o(W_o h_t + b_o) \quad (7)$$

di mana W_o merupakan matriks bobot lapisan keluaran dan b_o merupakan vektor bias lapisan keluaran [11].

6. Bidirectional Gated Recurrent Unit (Bi-GRU)

Bi-GRU menggabungkan konsep dari arsitektur GRU dan Bi-LSTM. Dengan menggunakan GRU yang beroperasi secara *bidirectional*, model ini dapat menangkap konteks teks dari kedua arah.

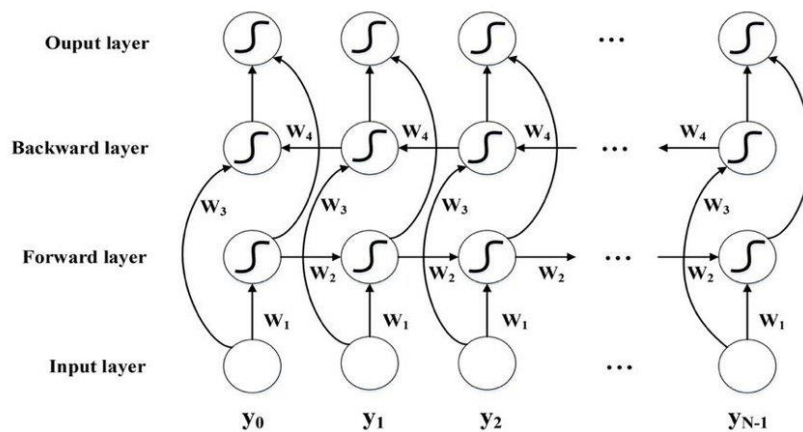


Fig. 7. Arsitektur Bi-GRU. Sumber: [15]

7. Convolutional Neural Network (CNN)

CNN adalah arsitektur jaringan saraf tiruan yang biasanya digunakan untuk pengolahan citra. Namun, dalam konteks pemrosesan teks, CNN dapat digunakan untuk mengekstrak fitur lokal dari teks dan mempelajari representasi yang berguna untuk klasifikasi.

Dalam klasifikasi teks, arsitektur Convolutional Neural Network (CNN) dapat digunakan untuk mengekstrak fitur-fitur lokal dari teks dan mempelajari representasi yang berguna untuk klasifikasi. Berikut adalah komponen *layer* CNN yang umumnya digunakan dalam klasifikasi teks.

- *Input Layer*: Input layer menerima representasi teks dalam bentuk vektor yang dapat diberikan dalam beberapa format, seperti *one-hot encoding*, *word embeddings*, atau karakter *embeddings*.
- *Embedding Layer*: *Embedding layer* adalah *layer* pertama dalam model CNN yang mengubah representasi teks menjadi representasi vektor dalam ruang fitur yang lebih padat. Hal tersebut dapat dilakukan dengan menggunakan *pretrained word embeddings*, seperti Word2Vec, GloVe, atau menggunakan *embedding layer* yang di-*train* bersamaan dengan model.
- *Convolutional Layer*: *Layer* konvolusi terdiri dari beberapa filter konvolusi yang bergerak melintasi representasi vektor teks. Filter ini memungkinkan model untuk mengekstrak fitur-fitur lokal dari teks. Setiap filter menghasilkan output konvolusi dengan menjalankan operasi konvolusi pada bagian-bagian kecil dari representasi teks.
- *Pooling Layer*: *Pooling layer* biasanya mengikuti *layer* konvolusi dan bertujuan untuk mengurangi dimensi representasi yang dihasilkan oleh *layer* konvolusi. *Max pooling* adalah jenis *pooling* yang umum digunakan dalam CNN untuk klasifikasi teks, di mana nilai maksimum dari setiap saluran (*channel*) diambil dari setiap area hasil konvolusi.
- *Flatten Layer* atau *Global Pooling*: Setelah *pooling layer*, hasil dari setiap filter atau kernel dikonkatenasi menjadi vektor tunggal, yang kemudian diberikan ke lapisan berikutnya. Alternatifnya, beberapa model menggunakan *global pooling* seperti *Global Max Pooling* atau *Global Average Pooling* untuk menghasilkan representasi agregat dari fitur-fitur yang diekstraksi.
- *Dense Layer (Fully Connected)*: *Dense layer* adalah *layer* berikutnya setelah hasil *pooling* yang bertanggung jawab untuk melakukan klasifikasi. *Layer* ini dapat memiliki beberapa lapisan dan menggunakan fungsi aktivasi seperti ReLU, sigmoid, atau softmax untuk menghasilkan output klasifikasi.
- *Output Layer*: *Output layer* menghasilkan prediksi klasifikasi akhir berdasarkan hasil dari *dense layer*. Untuk klasifikasi teks biner, *output layer* biasanya menggunakan satu neuron dengan fungsi aktivasi sigmoid, sedangkan untuk klasifikasi multikelas, fungsi aktivasi yang digunakan adalah softmax.

Setiap *layer* dalam arsitektur CNN untuk klasifikasi teks memiliki peran yang penting dalam mengekstraksi fitur dan menghasilkan prediksi akhir. Dengan melalui serangkaian *layer* ini, model CNN dapat mempelajari representasi yang berguna untuk membedakan dan mengklasifikasikan teks dengan akurasi yang tinggi.

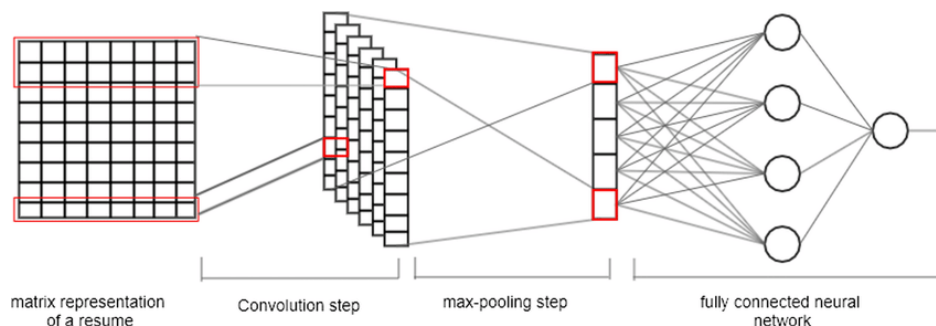


Fig. 8. Arsitektur CNN. Sumber: [16]

8. Performance Metrics

Untuk mengevaluasi kinerja dari model-model di atas, akan digunakan metrik-metrik penilaian kinerja, seperti *accuracy*, *precision*, *recall*, dan *F1-measure*. *Accuracy* mengukur keseluruhan kebenaran prediksi model, sementara *precision* mengevaluasi proporsi dari label positif yang diprediksi dengan benar. *Recall* menilai kemampuan model untuk mengidentifikasi label positif dengan benar, dan *F1-measure* menyediakan pengukuran seimbang dari *precision* dan *recall*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

di mana $TP = \text{True Positive}$, $TN = \text{True Negative}$, $FP = \text{False Positive}$, dan $FN = \text{False Negative}$.

C. Flow Chart

Berikut ini merupakan alur penelitian yang dilakukan untuk mengidentifikasi sarkasme pada judul berita menggunakan model-model deep learning.

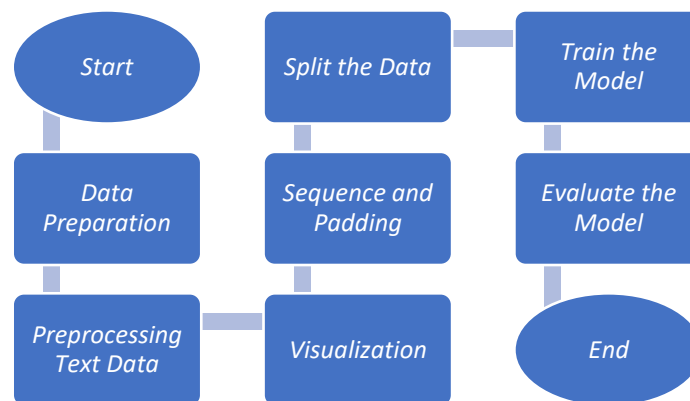


Fig. 9. Alur Penelitian

IV. Results and Discussions

Dataset yang digunakan pada penelitian ini merupakan dataset judul berita untuk deteksi sarkasme yang dikumpulkan dari dua situs berita, yaitu The Onion dan HuffPost. Dataset ini berisi 26.602, yang terdiri dari 14.951 judul berita dengan label nonsarkastis (label 0) dan 11.651 judul berita dengan label sarkastis (label 1). *Preprocessing* diawali dengan menghapus kolom yang tidak digunakan dan menghilangkan data duplikat. Judul berita ditulis oleh profesional secara formal. Oleh karena itu, untuk tidak menghapus apa pun yang dapat membantu konteks, hanya dilakukan beberapa *preprocessing* dasar, yaitu menghilangkan tanda baca, menghilangkan angka, dan menerapkan lemmatisasi. Visualisasi teks menunjukkan bahwa judul berita sarkastis seringkali mengandung kata-kata seperti “report”, “man”, dan “new”, sedangkan judul berita sering nonsarkastis sering kali diberikan mengandung kata-kata seperti “donald trump”, “trump”, “new”, dan “say”.

Dalam penelitian ini, digunakan enam arsitektur model deep learning yang berbeda untuk identifikasi sarkasme pada dataset sarkasme di atas. Dataset dibagi menjadi data *train* dan data *test* dengan perbandingan 0,8 dan 0,2. Dalam penelitian ini, parameter yang digunakan untuk semua model diatur ke *epochs* = 5 dan *batch-size* = 32. Semua model ditambahkan *drop layer*, *global max pooling* 1D, dan *dense layer* dengan fungsi aktivasi ReLU. Keenam model menggunakan dimensi *output* 32, dengan parameter *dropout* 0.2.

Untuk membandingkan kinerja model, digunakan visualisasi *training accuracy*, *validation accuracy*, *training loss*, dan *validation loss*. Visualisasi ini memberikan wawasan tentang seberapa baik model belajar dari data dan membantu dalam mengevaluasinya efektivitas untuk tugas identifikasi sarkasme.

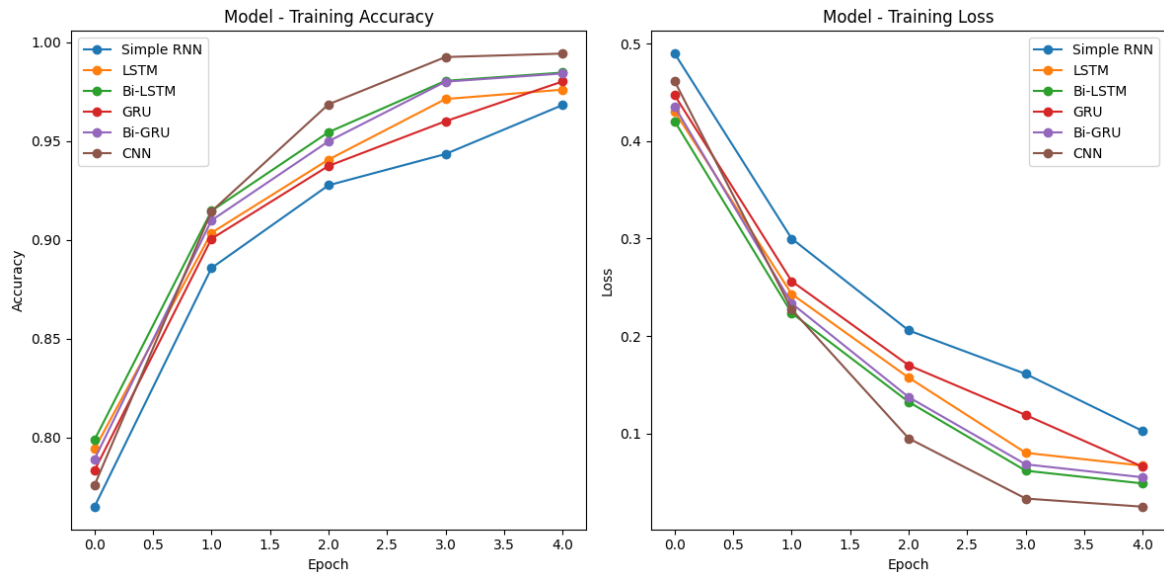


Fig. 10. Accuracy dan Loss Proses Training Model-Model Deep Learning pada Dataset Sarkasme

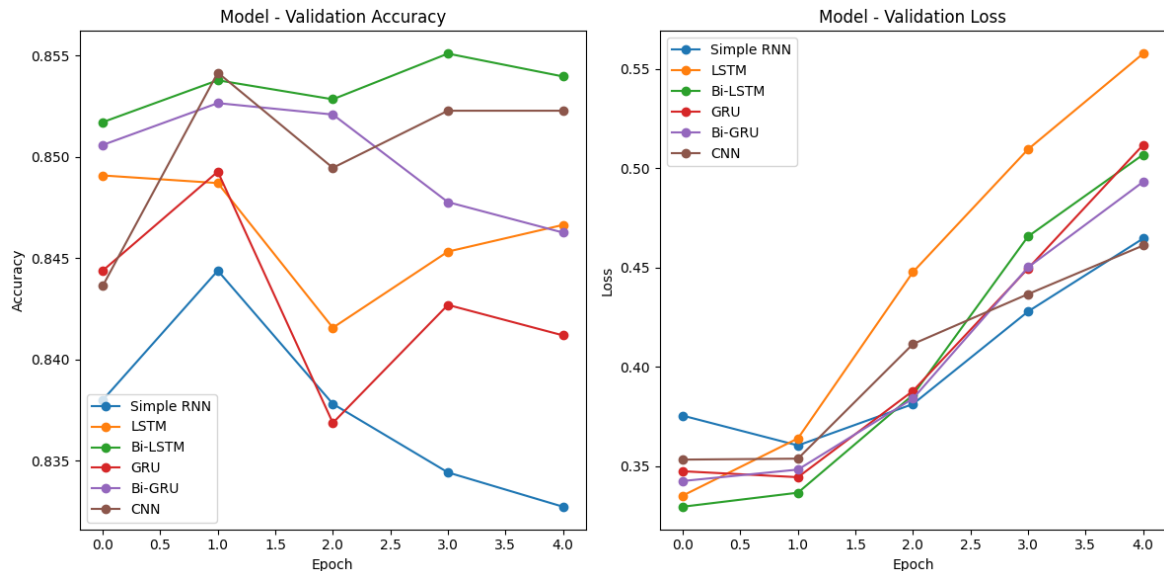


Fig. 11. Accuracy dan Loss Proses Validation Model-Model Deep Learning pada Dataset Sarkasme

Fig. 10 memberikan gambaran visual representasi bagaimana kinerja *accuracy* dan *loss* model-model deep learning yang berbeda selama proses *training*, sedangkan Fig. 11 kinerja selama proses *validation*. Fungsi kerugian (*loss*) mengukur perbedaan antara yang diprediksi label sarkasme dan label sebenarnya. Dari Fig. 10, didapatkan bahwa selama proses *training*, model CNN memberikan

performa yang baik secara keseluruhan, terlihat dari akurasi yang relatif tinggi secara keseluruhan dibandingkan model lainnya. Namun, akurasi yang tinggi dalam proses *training* tidak menjadi acuan dalam pemilihan model terbaik karena dapat mengindikasikan *overfitting*. Dari Fig. 11, terlihat bahwa model Bi-LSTM memiliki performa yang paling baik secara keseluruhan, disusul dengan model Bi-GRU dan CNN.

Table 2. Hasil Metrik Kinerja Model-Model Deep Learning

Model	Accuracy	Precision	Recall	F1-Score
RNN	0,83274	0,81915	0,79313	0,80593
LSTM	0,84665	0,83086	0,81588	0,8233
Bi-LSTM	0,85398	0,84557	0,81545	0,83024
GRU	0,8412	0,82073	0,81545	0,81808
Bi-GRU	0,84627	0,8354	0,80816	0,82155
CNN	0,85228	0,8465	0,80944	0,82756

Table 2 memberikan hasil metrik kinerja model-model deep learning dengan menggunakan data *test*. Berdasarkan hasil tersebut, dapat disimpulkan bahwa model Bi-LSTM dan CNN mencapai kinerja terbaik pada dataset ini untuk identifikasi sarkasme. Kedua model menunjukkan akurasi tinggi dalam mengidentifikasi sarkasme pada judul berita. Label 1 merupakan judul berita yang sarkas. Dalam kasus ini, *precision* paling penting dalam menentukan sarkasme karena jika terjadi kesalahan prediksi label 0 (nonsarkastis) padahal sebenarnya label 1 (sarkastis) maka akan dapat menyebabkan penyebaran kesalahan informasi. Berdasarkan Table 2, Bi-LSTM memiliki *precision* yang paling tinggi disusul dengan CNN. Hal tersebut sejalan dengan metrik akurasi, di mana Bi-LSTM dan CNN menjadi model yang baik.

V. Conclusions

Dalam mengevaluasi peran model deep learning pada identifikasi sarkasme, penelitian ini telah memberikan wawasan tentang kemajuan teknologi yang memungkinkan pengenalan dan pemahaman konteks sarkasme dalam teks dengan tingkat akurasi yang cukup tinggi. Penggunaan model deep learning untuk identifikasi sarkasme menunjukkan bahwa pendekatan Bi-LSTM dan CNN telah menunjukkan performa yang baik dalam identifikasi sarkasme pada judul berita. Model-model ini efektif dalam mempelajari pola-pola substansi dalam kalimat yang mencerminkan sarkasme, walaupun terdapat tantangan semantik dan kontekstual yang ada. Meskipun demikian, pengembangan lebih lanjut masih diperlukan untuk meningkatkan keakuratan, menangani kasus-kasus yang lebih kompleks, dan menyesuaikan model dengan perubahan bahasa dan budaya. Dengan penekanan pada penggunaan dataset yang besar dan bervariasi serta teknik-teknik yang canggih, penelitian di masa depan diharapkan dapat memperkuat kemampuan model dalam mengenali dan memahami sarkasme dengan lebih baik, khususnya pada judul berita.

References

- [1] R. Misra and P. Arora, "Sarcasm Detection using News Headlines Dataset," *AI Open*, vol. 4, pp. 13–18, 2023, doi: <https://doi.org/10.1016/j.aiopen.2023.01.001>.
- [2] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm Detection on Twitter: A Behavioral Modeling Approach," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, New York, NY, USA: ACM, Feb. 2015, pp. 97–106. doi: 10.1145/2684822.2685316.
- [3] M. F. Faiaz Isty, S. M. Mahedy Hasan, and M. S. Islam Shopnil, "Performance Analysis of Hybrid Deep Learning Models in Sarcasm Classification," in *2022 25th International Conference on Computer and Information Technology (ICCIT)*, 2022, pp. 1092–1097. doi: 10.1109/ICCIT57492.2022.10055499.
- [4] R. Pandey, A. Kumar, J. P. Singh, and S. Tripathi, "Hybrid attention-based Long Short-Term Memory network for sarcasm identification," *Appl Soft Comput*, vol. 106, p. 107348, Jul. 2021, doi: 10.1016/j.asoc.2021.107348.

- [5] P. Shrikhande, V. Setty, and Dr. A. Sahani, "Sarcasm Detection in Newspaper Headlines," in *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, IEEE, Nov. 2020, pp. 483–487. doi: 10.1109/ICIIS51140.2020.9342742.
- [6] R. Misra and P. Arora, "Sarcasm Detection using Hybrid Neural Network," *ArXiv*, vol. abs/1908.07414, 2019, [Online]. Available: <https://api.semanticscholar.org/CorpusID:201107123>
- [7] R. Ali *et al.*, "Deep Learning for Sarcasm Identification in News Headlines," *Applied Sciences*, vol. 13, no. 9, p. 5586, Apr. 2023, doi: 10.3390/app13095586.
- [8] A. Tondak, "Deep Learning Vs Machine Learning." Accessed: Mar. 30, 2024. [Online]. Available: <https://k21academy.com/datascience-blog/deep-learning/dl-vs-ml/>
- [9] J. Downson, "Machine Learning vs Deep Learning." Accessed: Mar. 30, 2024. [Online]. Available: <https://morioh.com/a/58904d3e463c/artificial-intelligence-vs-machine-learning-vs-deep-learning>
- [10] "A Survey on Understanding, Visualizations, and Explanation of Deep Neural Networks - Scientific Figure on ResearchGate." Accessed: Mar. 30, 2024. [Online]. Available: https://www.researchgate.net/figure/A-simple-RNN-architecture-unrolling_fig5_349025949
- [11] F. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU," *ArXiv*, vol. abs/2305.17473, 2023, [Online]. Available: <https://api.semanticscholar.org/CorpusID:258960275>
- [12] O. Calzone, "An Intuitive Explanation of LSTM." Accessed: Mar. 30, 2024. [Online]. Available: <https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c>
- [13] "Real-Time Cuffless Continuous Blood Pressure Estimation Using Deep Learning Model - Scientific Figure on ResearchGate." Accessed: Mar. 30, 2024. [Online]. Available: https://www.researchgate.net/figure/The-unfolded-architecture-of-Bidirectional-LSTM-BiLSTM-with-three-consecutive-steps_fig2_344751031
- [14] "A Dynamic DL-Driven Architecture to Combat Sophisticated Android Malware - Scientific Figure on ResearchGate." Accessed: Mar. 30, 2024. [Online]. Available: https://www.researchgate.net/figure/The-Architecture-of-basic-Gated-Recurrent-Unit-GRU_fig1_343002752
- [15] "Performance Evaluation of Channel Decoder based on Recurrent Neural Network - Scientific Figure on ResearchGate." Accessed: Mar. 30, 2024. [Online]. Available: https://www.researchgate.net/figure/The-architecture-of-Bi-GRU_fig3_338521450/actions#reference
- [16] "Skills prediction based on multi-label resume classification using CNN with model predictions explanation - Scientific Figure on ResearchGate." Accessed: Mar. 30, 2024. [Online]. Available: https://www.researchgate.net/figure/CNN-architecture-for-text-classification_fig7_343947158