JNATIA Volume 1, Nomor 1, November 2022
Jurnal Nasional Teknologi Informasi dan Aplikasinya

Penerapan Long Short-Term Memory dalam Mengklasifikasi Jenis Ujaran Kebencian pada Tweet Bahasa Indonesia

Ni Putu Sintia Wati[a1], Cokorda Pramartha[b2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]putu.sintia.wati@student.unud.ac.id
[2]cokorda@unud.ac.id

Abstract

Tweets are messages posted to Twitter and contain photos, videos, links, and text. Twitter is a social media service that allows everyone to communicate and stay connected through the rapid and frequent exchange of messages. However, as many communities have sprung up, user is getting less control while communicating on Twitter. One of them, more and more hate speech is being hurled either through retweets or replies to each other in one of the threads belonging to a particular community. To minimize this impact, a classification is needed to find out whether the tweet contains hate speech or not before being uploaded to Twitter. Due to the rapid increase in the current data usage, it is necessary to review it with other methods to classify the data more deeply. Based on these problems, the method that can be used is Long Short-Term Memory (LSTM). This study succeeded in providing predictions with different hyperparameter accuracy values for the LSTM application reaching 74%.

Keywords: *classification, LSTM, Tweet, hate speech*

1. Pendahuluan

Tweet adalah pesan yang diposting di Twitter yang dapat berisi teks, foto, video, atau tautan. Twitter memungkinkan pengguna untuk berkomunikasi cepat melalui pertukaran pesan, dan pada akhir 2021, rata-rata pengguna aktif harian mencapai 217 juta. Namun, semakin banyak komunitas yang bermunculan di Twitter, menyebabkan komunikasi yang kurang terkendali, termasuk peningkatan ujaran kebencian (hate speech). Ujaran kebencian sering melibatkan provokasi, hasutan, atau hinaan terhadap kelompok atau individu berdasarkan ras, agama, gender, dan sebagainya. Untuk meminimalkan dampak negatif ini, penting untuk mengklasifikasikan tweet yang berpotensi mengandung ujaran kebencian sebelum diposting. Dengan pesatnya peningkatan data, dibutuhkan metode klasifikasi yang lebih mendalam, salah satunya menggunakan Long Short-Term Memory (LSTM). LSTM, modifikasi dari Recurrent Neural Network (RNN), memiliki kemampuan untuk menyimpan informasi lebih lama dan mengatasi masalah vanishing gradient, di mana gradien pada input layer lebih kecil daripada output layer.

2. Metode Penelitian

2.1. Pengumpulan Data

Data penelitian ini bersumber dari Github, yaitu *multi-label hate speech and abusive language detection* [3], dengan 13.169 baris dan 13 atribut, bernilai 0 untuk 'tidak' dan 1 untuk 'ya'. Terdapat juga 15.167 baris kata yang akan dinormalisasi, termasuk kesalahan penulisan dan stopwords dalam bahasa Indonesia.

2.2. Preprocessing

Setelah pengumpulan data, tahap preprocessing mencakup penghapusan nilai null, karakter khusus, case folding, text normalization, stopword removal, dan tokenization.

2.3. Word Embedding

Word Embedding mengonversi kata menjadi vektor, merepresentasikan kata sebagai titik dalam ruang berdimensi tertentu. Pada Keras, lapisan Embedding digunakan dalam Neural Network untuk data teks dengan tiga argumen utama: (a) *input_dim* untuk ukuran kata, (b) *output_dim* untuk ukuran vektor, dan (c) *input_length* untuk panjang input.

2.4. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) adalah pengembangan dari RNN yang menambahkan sel LSTM untuk menangani masalah seperti pengenalan tulisan tangan dan ucapan [4]. LSTM dapat menyimpan bobot lebih lama berkat node self-recurrent dan lebih efektif pada data sekuens panjang. Cell gates mengatur informasi ke cell state, dan pada forget gate, informasi yang tidak relevan dihapus dengan fungsi sigmoid [4].

$$ft = \sigma\,(Wf\,[h_{t-1}, x_t] + b_f) \quad (1)$$

Pada input gate, informasi dipilih dan ditentukan sebelum dibawa ke cell state menggunakan fungsi aktivasi sigmoid, yang dihitung dengan persamaan 2. Selain itu, kandidat vektor baru ditentukan menggunakan fungsi aktivasi tanh, yang kemudian dibawa ke cell state menggunakan persamaan 3 [4].

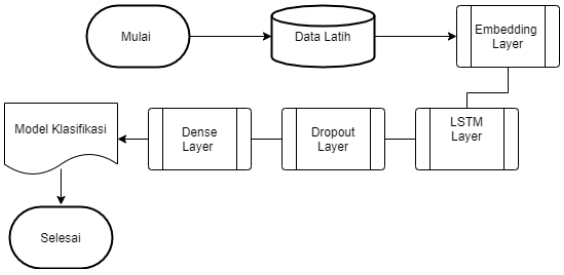
$$i_t = \sigma\,(W_f\,[h_{t-1}, x_t] + b_i) \quad (2)$$
$$\tilde{c}_t = tanh\ tanh\,(W_c\,[h_{t-1,}\, xt] + b_c) \quad (3)$$

Nilai pada cell state lama C_{t-1} akan diperbarui ke nilai cell state baru C_t melalui persamaan 4 [4].

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (4)$$

Fungsi sigmoid menghasilkan output pada hidden state, sementara cell state diproses dengan fungsi tanh. Kedua output ini dikalikan sebelum diproses lebih lanjut, dengan perhitungan dilakukan pada persamaan 5 dan 6 [4].

$$o_t = \sigma\,(W_o * [h_{t-1}, x_t] + b_o) \quad (5)$$
$$h_t = o_c * tanh(c_t) \quad (6)$$



Gambar 1. Diagram Alir Proses Pelatihan dengan Model LSTM

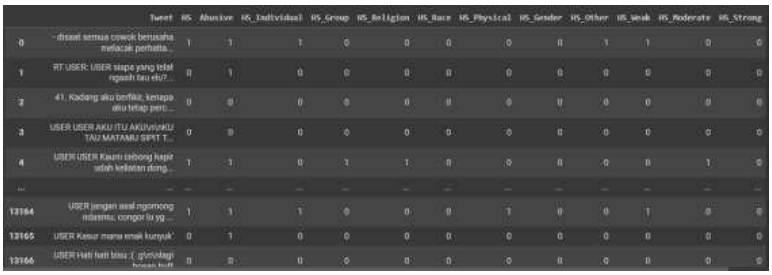
Arsitektur LSTM menggunakan fungsi aktivasi Sigmoid dan Tanh untuk transformasi nilai, dengan model yang terdiri dari tujuh layer, termasuk Embedding, LSTM, Dropout, dan Dense. Lapisan Embedding mengubah kata menjadi vektor berdasarkan kesamaan semantik, sementara LSTM mengatur parameter seperti memory unit dan dropout. Dense layer menambah fully connected layer

sesuai jumlah kelas.

3. Hasil dan Pembahasan

3.1. Data Understanding

Data yang digunakan dalam penelitian diperoleh melalui github, yaitu multi-label hate speech and abusive language detection [3] dan kaggle yaitu Indonesian Stoplist. Dataset tersebut memiliki 13169 data dengan 13 labels. Hasil pada tahap pengumpulan data ditampilkan pada Gambar 2.



	Tweet	HS	Abusive	HS_Individual	HS_Group	HS_Religion	HS_Race	HS_Physical	HS_Gender	HS_Other	HS_Weak	HS_Moderate	HS_Strong
0	- disaat semua cowok berusaha melacak perhatia...	1	1	1	0	0	0	0	0	1	1	0	0
1	RT USER: USER siapa yang telah ngasih tau elu?...	0	1	0	0	0	0	0	0	0	0	0	0
2	41. Kadang aku berfikir, kenapa aku tetap perc...	0	0	0	0	0	0	0	0	0	0	0	0
3	USER USER AKU ITU AKU/ANKU TAU MATAMU SIPIT T...	0	0	0	0	0	0	0	0	0	0	0	0
4	USER USER Kaum cebong kapir udah kelelatan dong...	1	1	0	1	1	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
13164	USER jangan awal ngomong ndasmu, congor lu yg ...	1	1	1	0	0	0	1	0	0	1	0	0
13165	USER Kasur mana enak kunyuk'	0	1	0	0	0	0	0	0	0	0	0	0
13166	USER Hati hati bisa :(gitvidagi	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 2. Hasil Pengumpulan Data

3.2. Data Preprocessing and Preparation

Data diatas selanjutnya akan dilakukan preprocessing data untuk meminimalisir terjadinya bias pada proses selanjutnya. rincian pembagian data ditunjukkan pada Tabel 1:

Table 1. Hasil Preprocessing

Process	Process Title	Result
1	**Data Awal**	USER USER USER USER BANCI KALENG MALU GA BISA JAWAB PERTANYAAN KAMI DARI 2 HARI LALU. NYUNGSEP KOE USER URL
2	**remove character**	BANCI KALENG MALU GA BISA JAWAB PERTANYAAN KAMI DARI 2 HARI LALU.... NYUNGSEP KOE
3	**Case Folding**	banci kaleng malu ga bisa jawab pertanyaan kami dari dua hari lalu…. nyungsep koe
4	**remove punctuation**	banci kaleng malu ga bisa jawab pertanyaan kami dari dua hari lalu nyungsep koe
5	**text normalization**	banci kaleng malu tidak bisa jawab pertanyaan kami dari dua hari lalu nyungsep kau
6	**Tokenization**	['banci', 'kaleng', 'malu', 'tidak', 'bisa', 'pertanyaan', 'kami', 'dari', 'dua', 'hari', 'lalu','nyungsep','kau']
Pada Tahapan	tokenizer, kamus data	atau num_word ditentukan sebanyak 1000. Tahap

Selanjutnya, data dibagi menjadi 80% data latih dan 20% data uji. Rincian pembagian data dapat dilihat pada tabel berikut:

Kelompok Data Jumlah	Kelompok Data Jumlah
Data Latih 10.535	Data Latih 10.535
Data Uji 2.634	Data Uji 2.634

Jumlah 13.169	**Jumlah 13.169**

Fitur yang digunakan untuk klasifikasi jenis hate speech mencakup tweet, hs_religion, hs_race, hs_physical, hs_gender, dan hs_other. Fitur selain dari ini dapat dihapus untuk memfokuskan analisis pada elemen-elemen yang relevan dengan klasifikasi tersebut.

3.3. Pemodelan menggunakan LSTM

Vektor dengan *input_length* sesuai *num_word* diproses melalui embedding, lalu dilanjutkan dengan pemodelan menggunakan parameter berikut:

Layer	Jumlah Neuron	Addition
Embedding	-	input_dim=1000, output_dim=32
LSTM	64	-
LSTM	64	return_sequences=True
Dense	16	-
Dropout	-	0.2
Dense	6	activation=softmax

3.4. Evaluasi Model

Berikut adalah hasil evaluasi pengujian model dengan menggunakan LSTM dan Adam optimizer (Gambar 3).

	precision	recall	f1-score	support
0	0.60	0.54	0.57	167
1	0.67	0.60	0.64	91
2	0.00	0.00	0.00	60
3	0.60	0.06	0.11	50
4	0.73	0.68	0.70	727
5	0.80	0.89	0.84	1539
accuracy			0.77	2634
macro avg	0.57	0.46	0.48	2634
weighted avg	0.74	0.77	0.75	2634

Gambar 3. Hasil Evaluasi

4. Kesimpulan

Berdasarkan hasil penelitian, nilai presisi yang didapat sebesar 74%, recall 77% dan f1-score sebesar 75%. Nilai tersebut dapat berubah jika dilakukan hyperparameter tuning untuk meningkatkan hasil akurasi.

References

[1] T. Spangler, "Variety," 2 February 2022. [Online]. Available: https://variety.com/2022/digital/news/twitter-q4-2021-earnings-users-growth-1235176882/.
[2] D. K. Teologi, “Studia Sosial Religia,” vol. 3, no. 1, pp. 70–82, 2020, [Online]. Available: http://jurnal.uinsu.ac.id/index.php/ssr
[3] M. O. Ibrohim and I. Budi, “Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter,” pp. 46–57, 2019, doi: 10.18653/v1/w19-3506.
[4] J. Brownlee, “How to Prepare Text Data for Machine Learning with Scikit-Learn,” 2019. [Online]. Available: https://machinelearningmastery.com/prepare-text-data-machine- learning-scikit-learn/.