

Deteksi Sarkasme dan Ironi pada Twitter dengan Menggunakan Metode CNN

Arvanchrist Charlie Wijaya¹, I Gede Arta Wibawa²

^{1,2}Informatika, Universitas Udayana
Kampus Bukit, Jl. Raya Kampus Unud Jimbaran, Kec. Kuta Sel., Kabupaten Badung, Bali,
Indonesia
¹arvanwijaya01@gmail.com
²gede.arta@unud.ac.id

Abstract

Sarcasm refers to the use of words with meaning opposite of what is written or said. With this ambiguity, it could be difficult to tell if the message is sarcastic or not. This paper aim to give some method to detect sarcasm in message, especially in a tweets from Twitter. In this paper we will use a convolutional neural network method to classify the tweets. In this paper we got the accuracy of 73,8% training and 74,6% for validation.

Keywords: Classification, Sentiment Analysis, Sarcasm Detection, Convolutional Neural Network, Natural Language Processing

1. Pendahuluan

Dalam kalimat yang mengandung sarkasme dan ironi umumnya memiliki makna yang berkebalikan dari kata-kata yang digunakan. Apa yang dimaksud dari sarkasme adalah penggunaan kata yang memiliki arti yang berlawanan dengan apa yang ingin disampaikan yang biasanya digunakan dalam menghina atau mengejek orang. Ironi juga menyerupai sarkasme dimana ironi adalah penggunaan kata yang berlainan dengan apa yang terjadi. Tetapi, terdapat pula perbedaan utama dari sarkasme dan ironi, dimana sarkasme memiliki konotasi negatif sedangkan ironi tidak.

Kalimat yang mengandung sarkasme dan ironi seringkali sulit untuk dibedakan dengan kalimat biasa[1]. Permasalahan tersebut seringkali menyebabkan salah komunikasi. Oleh karena itulah digunakan teknik-teknik tertentu dalam membedakan kalimat tersebut. Deteksi sarkasme dalam ilmu komputer adalah termasuk dalam bagian analisis sentiment dimana dalam pemrosesan bahasa alami masuk ke dalam jenis permasalahan klasifikasi[2]. Dalam deteksi sarkasme umumnya data tekstual akan diklasifikasikan apakah termasuk sarkasme atau bukan.

Pada tulisan ini akan dilakukan deteksi sarkasme pada tweet dari media sosial Twitter. Pesan-pesan tweet dari Twitter ini akan dipisahkan menjadi empat buah kelas yaitu untuk pesan yang mengandung sarkasme, pesan yang mengandung ironi, pesan yang mengandung sarkasme dan ironi, dan pesan yang tidak mengandung sarkasme maupun ironi. Metode klasifikasi yang akan digunakan adalah metode *convolutional neural network* atau CNN. CNN umumnya digunakan dalam proses klasifikasi untuk data citra, namun penggunaan metode CNN tidak terbatas hanya untuk data citra saja[3].

2. Metodologi

Deteksi sarkasme adalah salah satu bagian dari sentimen analisis dari itu dapat menggunakan metode yang serupa[4]. Dalam membangun model untuk melakukan deteksi sarkasme dan ironi akan dilakan dalam beberapa tahapan.

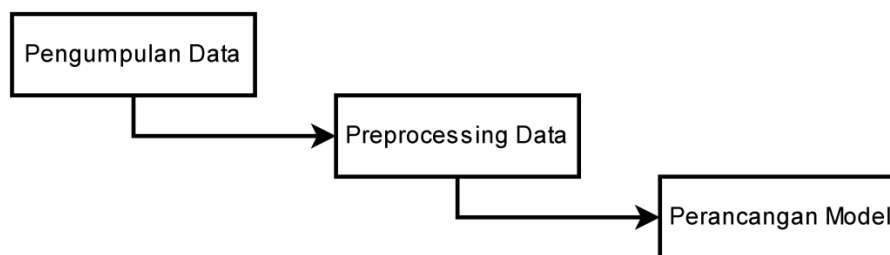


Figure 1. Metodologi Deteksi Sarkasme

Dalam tulisan ini akan dilakukan dalam beberapa tahapan dimana pertama-tama adalah dalam pengumpulan data. Ketika sudah ditemukan kemudian dataset akan dipreprocessing dimana data akan diolah untuk dapat digunakan dalam model. Data yang telah dipreprocessing akan dilakukan tokenisasi. Selanjutnya dataset yang telah ditokenisasi akan digunakan untuk melatih model *convolutional neural network*. Selanjutnya akan dilakukan evaluasi dan analisis daripada model yang dibuat.

2.1. Pengumpulan Data

Dalam pemrosesan bahasa alami banyak sekali sumber data yang dapat digunakan untuk penelitian dari menggunakan artikel-artikel sampai menggunakan data dari sosial media. Dalam penelitian ini digunakan data yang berasal dari sosial media twitter. Alasan diambilnya data dari media sosial Twitter sebab banyaknya ketersediaan data di media sosial.

Selain itu, kini juga mudah dan banyak metode yang dapat digunakan untuk mendapatkan data dari sosial media dari menggunakan API, melalui *web scraping*, dan lain sebagainya[5]. Twitter sendiri memiliki API tersendiri yang banyak dipakai di berbagai penelitian[6]. Selain metode pengumpulan data tersebut terdapat pula data - data dari sumber lain seperti dari repositori - repositori publik. Dalam penelitian ini akan dicari data dari repositori tersebut, sebab pengumpulan data melalui *web scraping* dan API akan menghabiskan waktu yang cukup banyak terutama dalam pelabelan data.

Apabila dataset telah ditemukan selanjutnya akan digunakan sebagai data latih serta data validasi. Sebelum data digunakan dalam melatih model akan dilakukan preprocessing data serta tokenisasi.

2.2. Preprocessing Data

Preprocessing data bertujuan dalam mempersiapkan data sebelum dilakukan pelatihan pada model[7]. Dalam tahapan ini terdapat berbagai pemrosesan terhadap dataset yang didapatkan diantaranya yaitu menghapus mention, menghapus URL, menghapus hashtag, menghapus angka, menghapus emoticon, *case folding*, menghapus tanda baca, dan tokenisasi.

a. Tweet Cleaning

Tahapan preprocessing ini bertujuan untuk membersihkan data pada tweet terlebih dahulu. Pada tweet twitter biasanya terdiri atas banyak komponen yang kurang bermanfaat dalam proses deteksi sarkasme seperti diantaranya angka, *emoticon*, *hashtag*, *mention*, dan url.

b. Lowercasing

Tahapan ini adalah salah satu tahapan preprocessing yang ternilai sederhana dimana huruf kapital akan diubah ke dalam huruf kecil. Tahapan ini meski sederhana namun memiliki dampak yang penting. Dengan menggunakan lowercasing maka kata yang semulanya terpisah menjadi kata berbeda karena perbedaan huruf kapital maka kini dapat dihitung sebagai kata yang sama.

c. Punctuation Removal

Dalam *punctuation removal* dilakukan penghapusan tanda baca serta angka dalam tweet. Hal tersebut dilakukan sebab angka dan tanda baca dianggap tidak memiliki peran dalam menunjukkan apakah kalimat tersebut termasuk dalam sarkasme dan ironi atau bukan.

d. Tokenization

Tokenisasi adalah proses untuk membagi teks ke dalam token - token[8]. Tiap token terdiri dari satu kata. Pada tokenisasi tiap kata akan direpresentasikan dengan angka bilangan bulat yang unik. Data teks yang telah ditokenisasi adalah data yang baru dapat untuk digunakan untuk melatih model *neural network*.

2.3. Perancangan Model

Dalam penelitian ini akan digunakan convolutional neural network dalam mendeteksi sarkasme dan ironi. *Convolutional neural network* adalah salah satu jenis *neural network* yang sering digunakan dalam klasifikasi citra. Terdapat pula berbagai jenis *convolutional neural network* atau CNN, seperti yang digunakan dalam klasifikasi citra yaitu CNN dua dimensi. Untuk CNN satu dimensi dapat digunakan untuk berbagai aplikasi seperti *speech recognition*, *electrocardiogram monitoring*, dan lain sebagainya[9]. Dalam perancangannya berdasarkan data yang digunakan model akan menggunakan konvolusi satu dimensi. Hal itu dikarenakan data teks akan direpresentasikan sebagai kumpulan kata.

3. Hasil dan Pembahasan

Pada bagian ini akan diberikan gambaran daripada hasil model yang dibuat. Model diimplementasikan pada Google Colab dengan menggunakan Tensorflow. Terdapat beberapa tahapan pokok dalam implementasi ini yaitu pengambilan data, *preprocessing*, pemodelan, dan diakhiri dengan evaluasi.

3.1. Pengambilan Data

Pada tahapan ini dilakukan pengambilan data dari Kaggle. Dataset yang digunakan adalah dataset "Tweets with Sarcasm and Irony" yang diupload oleh Nikhil John. Dataset terdiri atas file *train.csv* dan *test.csv*. File *train.csv* terdiri atas 67.997 buah data sedangkan file *test.csv* terdiri atas 7.994 buah data yang terbagi menjadi empat buah kelas yaitu *regular*, *sarcasm*, *irony*, dan *figurative*.

Table 1. Dataset Tweet Sarkasme dan Ironi

Nama file	Jumlah data
<i>train.csv</i>	67.997
<i>test.csv</i>	7.994

Dalam kedua file tersebut terdapat atas dua kolom yaitu *tweet* dan *class*. Kolom *tweet* berisikan dengan teks tweet dari Twitter, sedangkan kolom *class* berisikan kelas daripada tweet tersebut. Pada dataset ini terdapat empat buah kelas yaitu *irony*, *sarcasm*, *figurative*, dan *regular*. Kelas *irony* yaitu tweet yang mengandung ironi. Kelas *sarcasm* yaitu tweet yang mengandung sarkasme. Kelas *figurative* yaitu tweet yang mengandung ironi dan sarkasme. Kelas *regular* yaitu tweet yang tidak mengandung ironi maupun sarkasme.

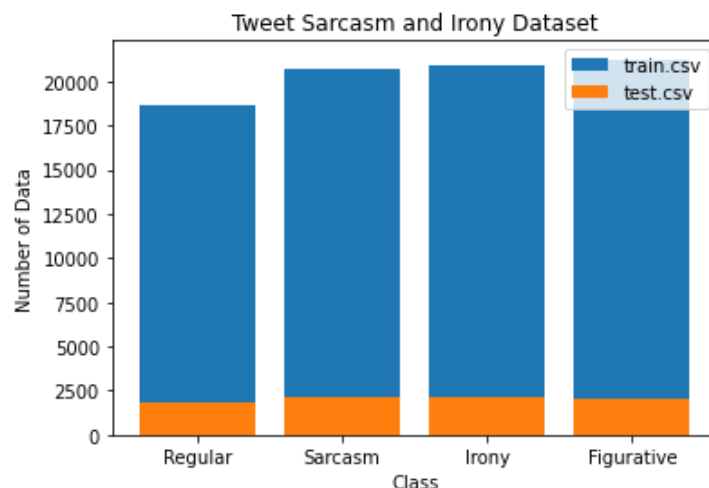


Figure 2. Distribusi Kelas Dataset

Diagram diatas menggambarkan distribusi kelas daripada dataset yang digunakan dimana yang berwarna biru adalah data dalam file *train.csv* yang akan digunakan sebagai data latih dan yang berwarna oranye adalah data dalam file *test.csv* yang akan digunakan sebagai data validasi.

3.2. Preprocessing

Data yang diambil selanjutnya akan melalui proses *preprocessing*. Terdapat beberapa tahapan dalam preprocessing yaitu *tweet cleaning*, *lowercasing*, *punctuation removal*, dan *tokenization*.

a. Tweet Cleaning

Berikut adalah tahapan *tweet cleaning* pada salah satu tweet dalam dataset.

Table 2. Tweet Cleaning

Before	After
My jobs great because I love what I do (operate rides for kids) and sometimes if the weather's right, I get a shower too!! #rain #sarcasm	My jobs great because I love what I do (operate rides for kids) and sometimes if the weather's right, I get a shower too!!

b. Lowercasing

Berikut adalah tahapan *lowercasing* pada salah satu tweet dalam dataset.

Table 3. Lowercasing

Before	After
My jobs great because I love what I do (operate rides for kids) and sometimes if the weather's right, I get a shower too!!	my jobs great because i love what i do (operate rides for kids) and sometimes if the weather's right, i get a shower too!!

c. Punctuation Removal

Berikut adalah tahapan *punctuation removal* pada salah satu tweet dalam dataset.

Table 4. Punctuation Removal

Before	After
my jobs great because i love what i do (operate rides for kids) and sometimes if the weather's right, i get a shower too!!	my jobs great because i love what i do operate rides for kids and sometimes if the weather's right i get a shower too

d. Tokenization

Dalam proses ini digunakan *Tokenizer* dari Tensorflow Keras. Dalam proses tokenisasi data teks dipisahkan menjadi kata - kata serta memiliki representasi bilangan bulat sehingga dapat digunakan sebagai input dari model *neural network*.

3.3. Pemodelan

Untuk data berupa teks digunakan *convolution neural network* satu dimensi. Terdapat pula berbagai *layers* yang akan digunakan dalam model yaitu *embedding*, *conv1d*, *global max pooling 1d*, dan *dense*. Berikut adalah rincian daripada model yang dibuat.

Table 2. Model Neural Network

Layer (type)	Output Shape	Param
embedding (Embedding)	(None, 120, 16)	16000
conv1d (Conv1D)	(None, 118, 64)	3136
conv1d_1 (Conv1D)	(None, 116, 64)	12352
global_max_pooling1d (GlobalMaxPooling1D)	(None, 64)	0
dense (Dense)	(None, 32)	2080
dense_1 (Dense)	(None, 4)	132

Layer *Embedding* biasanya digunakan dalam data berupa teks. Layer ini menerima inputan dari data yang telah ditokenisasi. Setelah masuk *embedding* selanjutnya akan masuk ke layer konvolusi dimana untuk data teks digunakan konvolusi satu dimensi. Pada layer konvolusi data akan dibagi menjadi bagian - bagian yang lebih kecil. Pada layer *max pooling* bagian-bagian dari konvolusi akan digabungkan kembali. Pada layer *dense* terakhir akan ditentukan apakah termasuk sarkasme, ironi, atau tidak.

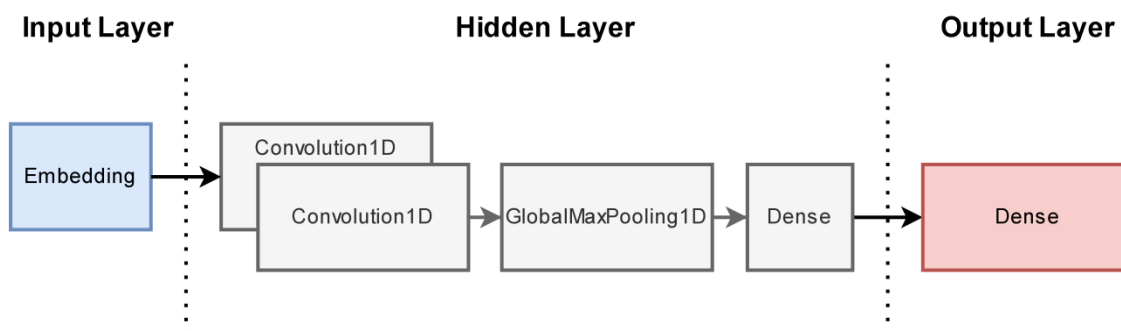


Figure 3. Neural Network Model

Pada model *neural network* yang dibangun digunakan layer *embedding* sebagai layer input. Untuk *hidden layer* dari model yang dibangun terdiri atas layer konvolusi, *max pooling*, dan layer *dense*. Untuk layer *output* digunakan layer *dense* dengan aktivasi *softmax*.

3.4. Hasil

Data kemudian dilatih sebanyak 10 epoch. Setelah data dilatih didapatkan nilai akurasi 73,8% untuk data latih dan 74,6% untuk data validasi. Sedangkan, nilai loss yang didapatkan yaitu 0,5

untuk data latih dan 0,49 untuk data validasi. Berikut adalah tabel akurasi dan loss dari model yang dibuat.

Table 3. Akurasi dan Loss Data Latih dan Validasi

Data	Akurasi	Loss
Data Latih	73,8%	0,5
Data Validasi	74,6%	0,49

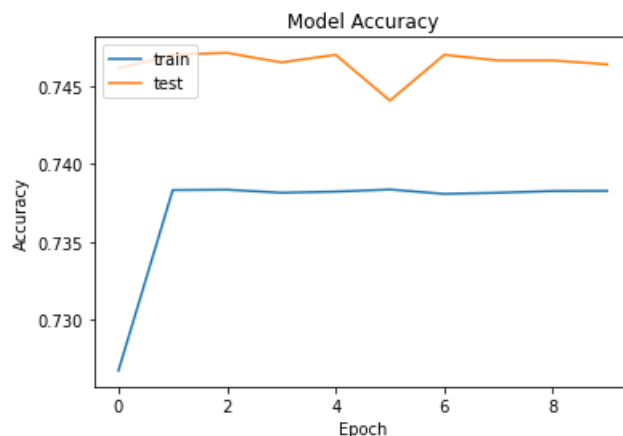


Figure 4. Grafik Akurasi Model

Diatas adalah grafik dari akurasi yang dicapai model. Dari grafik diatas didapatkan bahwa model mencapai puncak akurasi dengan cukup cepat dimana model telah mencapai puncak akurasi pada *epoch* awal.

4. Kesimpulan

Berdasarkan penelitian ini ditemukan bahwa metode *convolutional neural network* dapat digunakan dalam pemrosesan bahasa alami yakni dalam hal deteksi sarkasme. Dari hasil latih model didapatkan akurasi 73,8% untuk data latih dan 74,6% untuk data validasi.

References

- [1] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright. "Sarcasm detection using machine learning algorithms in Twitter: A systematic review" *International Journal of Market Research*, vol.62, no.5, p. 578-598
- [2] P. M. Nadkarni, L. Ohno-Machado and W. W. Chapman. "Natural language processing: an introduction" *Journal of the American Medical Informatics Association*, vol.18, no.5, p. 544-551, 2011
- [3] W. Yin, K. Kann, M. Yu, and H. Schutze. "Comparative study of CNN and RNN for natural language processing". 2017
- [4] A. Kedia and M. Rasu. "Hands-on Python natural language processing: explore tools and techniques to analyze and process text with a view to building real-world NLP applications". Packt Publishing Ltd. 2020
- [5] H. Liang and J. J. Zhu. "Big data, collection of (social media, harvesting)" *The international encyclopedia of communication research methods*, p. 1-18. 2017
- [6] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi. "Twitter and research: A systematic literature review through text mining" *IEEE Access*, vol. 8, p. 67698-67717. 2020
- [7] J. Camacho-Collados and M. T. Pilehvar. "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis". 2017
- [8] T. Jo. "Text mining: Concepts, implementation, and big data challenge", vol. 45. Springer. 2018

- [9] S. Kiranyaz, O. Avci, O. Abdeljabel, T. Ince, M. Gabbouj, and D. J. Inman. "1 D convolutional neural networks and applications: A survey" *Mechanical systems and signal processing*, 151, 107398

Halaman ini sengaja dibiarkan kosong