NLP → Hate Speech Detection for Indonesia Tweets Using Word Embedding And Gated Recurrent Unit (penulis : Junanda Patihullah, Edi Winarko ,Program Studi S2 Ilmu Komputer FMIPA UGM, Yogyakarta)

Abstrak

- Paper ini membahas tentang deteksi ujaran kebencian di media sosial terutama di twitter menggunakan word embedding dan GRU(gated recurrent Unit),karena mustahil kita mendeteksi secara manual.
- Metode Gated Recurrent Unit (GRU) adalah salah satu metode deep learning yang memiliki kemampuan mempelajari hubungan informasi dari waktu sebelumnya dengan waktu sekarang
- Paper ini menggunakan fitur ekstraksi word2Vec, karena memiliki kemampuan mempelajari semantik antar kata
- Pada paper ini membandingkan metode GRU dg fitur word2Vec sebesar 92,96% dengan metode supervised lainya :
    - Support vector machine
    - Naive bayes
    - Random forest classifier
    - Logistic regression
- Pada metode pembanding yang menggunakan word2Vec memberikan hasil accuracy yang lebih rendah dibandingkan dengan penggunaan fitur TF dan TF-IDF

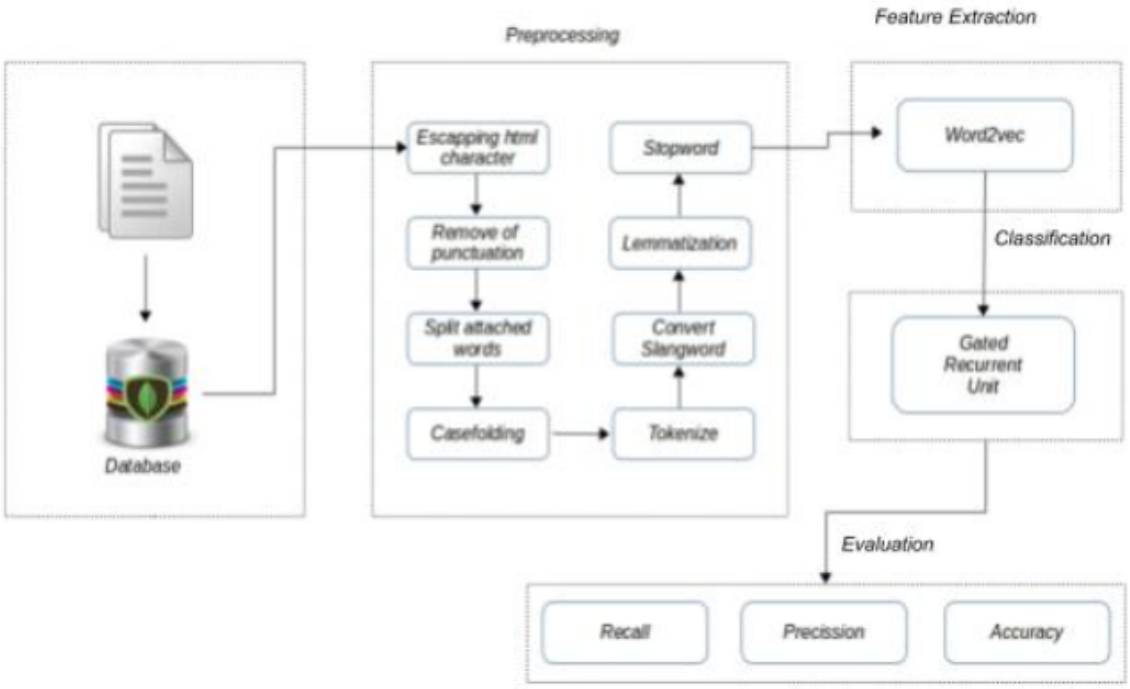
1. Introduction

- Pada introduction penulis membahas tentang perkembangan media sosial sebagai sarana komunikasi dan bertukar informasi . Dimana perkembanganya berbanding lurus dengan meningkatnya pengguna media sosial saat ini.
- Dengan meningkatnya pengguna media sosial banyak membuka peluang untuk terjadinya cyber crime di dunia maya seperti penyebaran informasi yang mengandung ujaran kebencian yang dilakukan secara tertulis yang ditujukan kepada seseorang atau sebuah kelompok sehingga menimbulkan kerugian dari pihak yang dituju.
- Mendeteksi ujaran kebencian ini penting untuk menganalisis sentimen publik terhadap sesuatu hal, atau sentimen publik dari kelompok tertentu terhadap kelompok lain, sehingga dapat mencegah dan meminimalkan tindakan atau hal yang tidak diinginkan.
- Penelitian tentang deteksi ujaran kebencian dengan bahasa Indonesia sudah dilakukan sebelumnya menggunakan:
    - Bag of word
    - Word n-gram
    - Character n-gram
- Deteksi ujaran kebencian untuk bahasa indonesia sebelumnya telah dilakukan dengan algoritma machine learning menggunakan metode kalsifikasi :
    - Bayesian Logistic Regression
    - Naive bayes
    - Support Vector machine

    - Random forest decision tree
- Untuk saat ini ,nilai tertinggi didapatkan dengan n-gram yang dikombinasikan dengan :
    - Random forest decision tree (93,5%),
    - Bayesian Logistic Regression(91.5%)
    - Naive bayes(90,2%)
- Deteksi ujaran kebencian dengan bahasa Indonesia juga dapat dilakukan dengan menggunakan algoritma jaringan syaraf tiruan(backpropagation) dengan kombinasi fitur berbasis lexicon dan bag of words dengan akurasi tertinggi diperoleh (78,81%)
- Pada paper ini penulis akan membahas kombinasi word embedding sebagai fitur dan GRU sebagai pengklasifikasian untuk mendeteksi ujaran kebencian di tweet Indonesia.

2. Metode

Hate Speech Detection Architecture



Scraping data → masukin database → preprocessing (escaping html character,remove punctuation,split attached word,caseFolding,tokenize,convert stopwords,lemmatization,stopwords → feature extraction(word2Vec) → classification (GRU) → evaluation(recall,precision,accuracy)

2.1 Preprocessing

Ini adalah fase penting dalam klasifikasi untuk mendapatkan model terbaik. Prepro terdiri dari beberapa step :

- Escaping html characters → menghilangkan link URL,karakter HTML
- Removal of punctuation → menghapus karakter seperti hastag(#),@user,retwet(RT) dan tanda baca

- Split attached words→ mengubah data teks yang sifatnya informal menjadi bentuk normal menggunakan simple rules dan regex
- Case folding→ proses converting menjadi lowercase
- Tokenization→ split text menjadi unit kecil
- Convert slang words → transformasi dari slang words menjadi kata kata yg standar
- Removal of stop-words → membuang informasi yang tidak dibutuhkan berdasarkan kamus stoplist.Penelitian ini menggunakan list stopwords dari Rahmawan

2.2 Feature Extraction

- Komponen utama untuk menghasilkan nilai vektor di word2Vec adalah jaringan saraf tiruan yang dibangun oleh arsitektur CBOW dan skip-gram.
- Pertama word2Vec akan membuat sebuah model distribusi kata pada saat training menggunakan dokumen bahasa indonesia yang dikumpulkan dari wikipedia
- Jumlah dokumen yang digunakan 1.120.973
- Feature model word2Vec menggunakan tiga proses :
    - Vocabulary Builder
    - Context builder
    - Neural network

2.2.1 Vocabulary builder

- Pembangun kosakata adalah blok bangunan pertama dari model word2vec.
- Dibutuhkan data teks mentah, sebagian besar berupa kalimat. Pembangun kosa kata digunakan untuk membangun kosa kata
- dari korpus teks. Ini akan mengumpulkan semua kata-kata unik dari corpus dan membangun kosa kata.
- Dipembangun kosakata ini, data yang digunakan adalah dokumen yang telah diunduh dari Wikipedia.
- Hasil dari proses pembangun kosakata adalah kamus kata dengan kata
- indeks dan nilai kemunculan setiap kata

2.2.2 Context Builder

- Context builder menggunakan output dari pembuat kosakata. Context builder adalah sebuah proses untuk mengetahui hubungan antara kemunculan satu kata dengan kata lain disekitarnya dengan cara menggunakan context window atau disebut juga sliding window.
- Secara umum, ukuran context window di NLP adalah 5 hingga 8 kata yang berdekatan.
- Jika kita memilih ukuran jendela isinya 5, kemudian 5 kata yang muncul di kiri dan kanan kata tengah.

- Dalam penelitian ini,ukuran jendela konten yang digunakan adalah 5. Tabel 1 memberikan contoh jendela konten dengan ukuran jendela 1.
- Kata yang digarisbawahi adalah kata tengah. Hasil konten jendela pembuat konten akan digunakan dalam proses selanjutnya, yaitu jaringan saraf bagian

Table 1 Examples context window

Text	Word pairing
I like deep learning.	(I, like)
I *like* deep learning.	(like, deep), (like, I)
I like *deep* learning.	(deep, learning), (deep, like)
I like deep *learning*.	(learning, .), (learning, deep)

2.2.3 Neural Networks (CBOW dan arsitektur skip-gram)

- Word2vec menggunakan arsitektur jaringan syaraf tiruan yang terbentuk dari arsitektur CBOW dan Skipgram.
- Jaringan syaraf tiruan ini digunakan untuk melakukan pelatihan agar setiap kata dapat diwakili oleh vektor.
- Dalam hal ini arsitektur neural network menggunakan 3 layer, input lapisan, lapisan tersembunyi dan lapisan keluaran
- Dalam penelitian ini, lapisan tersembunyi berisi 200 neuron dan lapisan output memiliki jumlah yang sama dengan lapisan input. Input untuk jaringan adalah nilai setiap kata yang telah diubah menjadi pengkodean satu-panas.

2.3 Klasifikasi

- Penelitian ini menggunakan GRU untuk mendeteksi ujaran kebencian dalam bahasa Indonesia. GRU adalah variasi pada LSTM yang lebih sederhana dari LSTM, dan dalam beberapa kasus menghasilkan hasil yang sama-sama sangat baik
- Sebagai LSTM, GRU (Gated Recurrent Unit) bertujuan untuk menyelesaikan masalah gradien hilang yang datang dengan jaringan saraf berulang standar.
- GRU menggabungkan gerbang lupa dan input gerbang menjadi satu gerbang pembaruan dan memiliki gerbang reset tambahan seperti yang ditunjukkan pada Gambar 4.
- GRU semakin populer dan banyak yang menggunakannya untuk menyelesaikan masalah NLP
- Untuk menyelesaikan vanishing gradient problem dari standard RNN , GRU menggunakan update gate dan reset gate.
- Pada dasarnya, ini adalah dua vektor yang memutuskan informasi apa yang harus diteruskan.

- Activation function dari GRU ht_j pada waktu t adalah interpolasi linier antara Activation function sebelumnya ht_j-1 dan output activationkandidat seperti yang terlihat pada Persamaan

$$h_t^j = z_t^j \circ h_{t-1}^j + (1 - z_t^j) \circ \widetilde{h_t^j} \quad (1)$$

  - ■
- Gerbang update function adalah z_{tj}i untuk memutuskan berapa banyak unit sebelumnya yang harus disimpan, sebagaimana dapat dilihat pada Persamaan

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \quad (2)$$

  - ○
- Ketika xt dimasukkan ke unit jaringan, itu dikalikan dengan beratnya sendiri Wz.. Itu hal yang sama berlaku untuk ht-1 yang menyimpan informasi untuk unit t-1 sebelumnya dan dikalikan dengan berat sendiri
- Gerbang reset function rtj digunakan dari model untuk memutuskan berapa banyak masa lalu informasi untuk dilupakan, seperti dapat dilihat pada Persamaan (3). Fungsi ini sama dengan fungsi untuk update gate ztj. . Perbedaannya terletak pada bobot Wr, Ur , dan penggunaan gerbang

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j \quad (3)$$

  - ○
- Kandidat keluaran activation function $\tilde{h}_{tj}$c menghitung nilai unit sebelum itu memutuskan untuk diperbarui atau tidak dan (◦) menunjukkan elemen perkalian produk Hadamard. Output kandidat aktivasi fungsi dapat dilihat pada Persamaan (4)

$$\widetilde{h_t^j} = \tanh(W x_t + r_t \circ U h_{t-1})^j \quad (4)$$

  - ○

3. HASIL & DISKUSI

Penelitian ini menggunakan ujaran kebencian Twitter dalam bahasa Indonesia yang telah dikumpulkan dan diberi label oleh [3]. Jumlah data tweet sebanyak 713 data, 260 tweet dilabeli sebagai ujaran kebencian ,453 dilabeli sebagai tweet non ujaran kebencian

A.Perbandingan word2Vec ieith TF dan TF-IDF

- Pada percobaan pertama kita akan mencoba membandingkan fitur word2vec dengan TF dan TF-IDF untuk mengetahui kemampuan word2vec sebagai fitur dalam model klasifikasi. Algoritma supervised yang akan digunakan untuk percobaan ini
  - Support vector machine,
  - Naive Bayes,
  - Regresi Logistik Bayesian
  - Random Forest.

- Percobaan ini dilakukan berdasarkan asumsi bahwa word2vec memiliki kemampuan yang lebih baik untuk mendeteksi ujaran kebencian dibandingkan fitur lainnya,yaitu, TF dan TF-IDF.

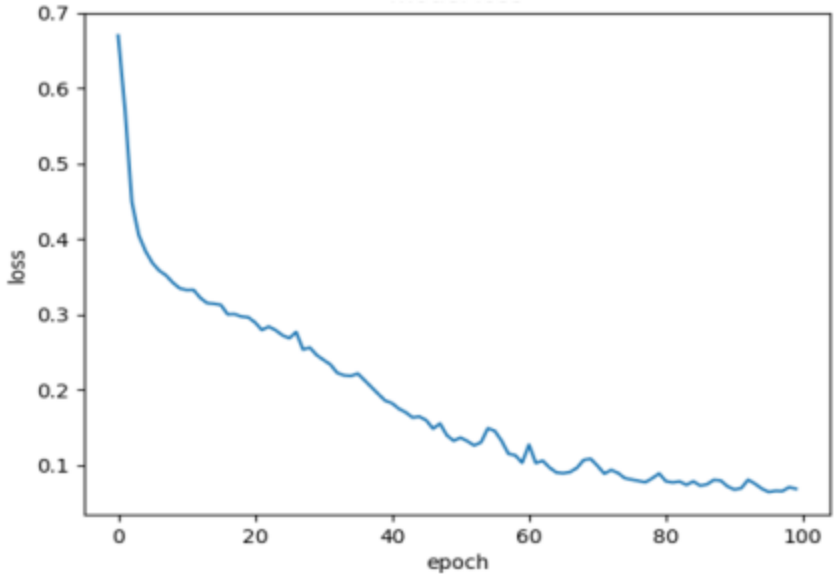
Table 2 Comparison of word2vec against TF and TF-IDF

Feature	Accuracy %			
	SVM	NB	BLR	RFDT
Word2vec	73.07	77.88	73.07	79.80
TF	83.65	79.80	78.84	81.73
TF-IDF	80.76	78.80	80.76	82.69

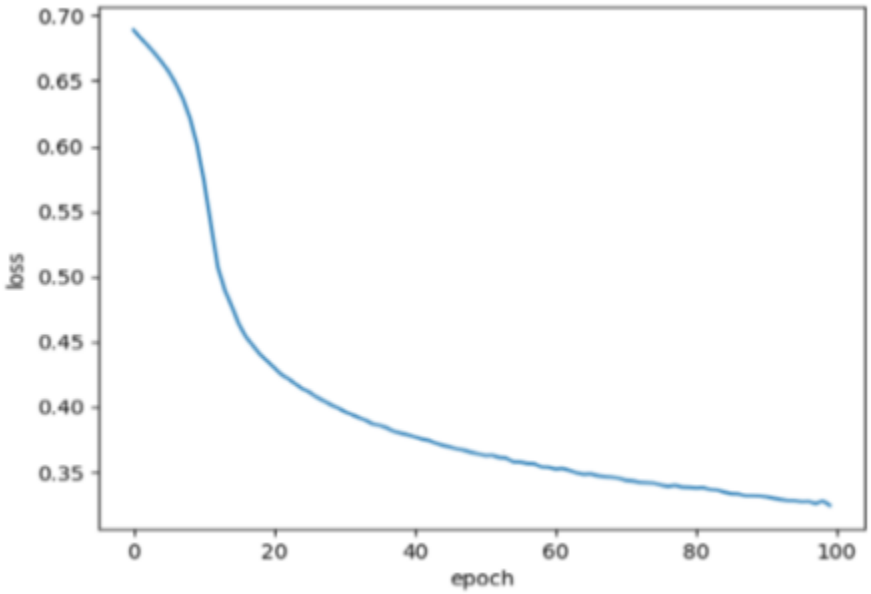
- Hasil percobaan dapat dilihat pada Tabel 2. Tabel ini menunjukkan bahwa akurasi tertinggi dari fitur word2vec dicapai dengan menggunakan algoritma random forest, dengan nilai akurasi 79,80%.
- Akurasi ini lebih rendah dari akurasi penggunaan TF dan TF-IDF pada semua algoritma.
- Eksperimen dasar ini menunjukkan bahwa untuk algoritma klasik yang digunakan dalam penelitian ini,
- Fitur word2vec menghasilkan akurasi yang lebih rendah dibandingkan dengan fitur TF dan TF-IDF

B.Menentukan Learning rate

- Eksperimen untuk menentukan learning rate dilakukan dengan lapisan GRU tunggal oleh setting jumlah neuron 200, dan epoch 100.
- Besarnya learning rate tentu tidak terlalu besar dan tidak terlalu kecil. Pemilihan learning rate yang besar akan membuat proses pembelajaran tidak terlalu optimal,
- sedangkan nilai learning rate yang terlalu kecil dapat menyebabkan proses pelatihan menjadi kurang baik dalam kompleksitas waktu.
- Nilai tingkat kemiringan diatur ke 0,001, 0,0001, dan 0,00001



a. Learning rate 0.001



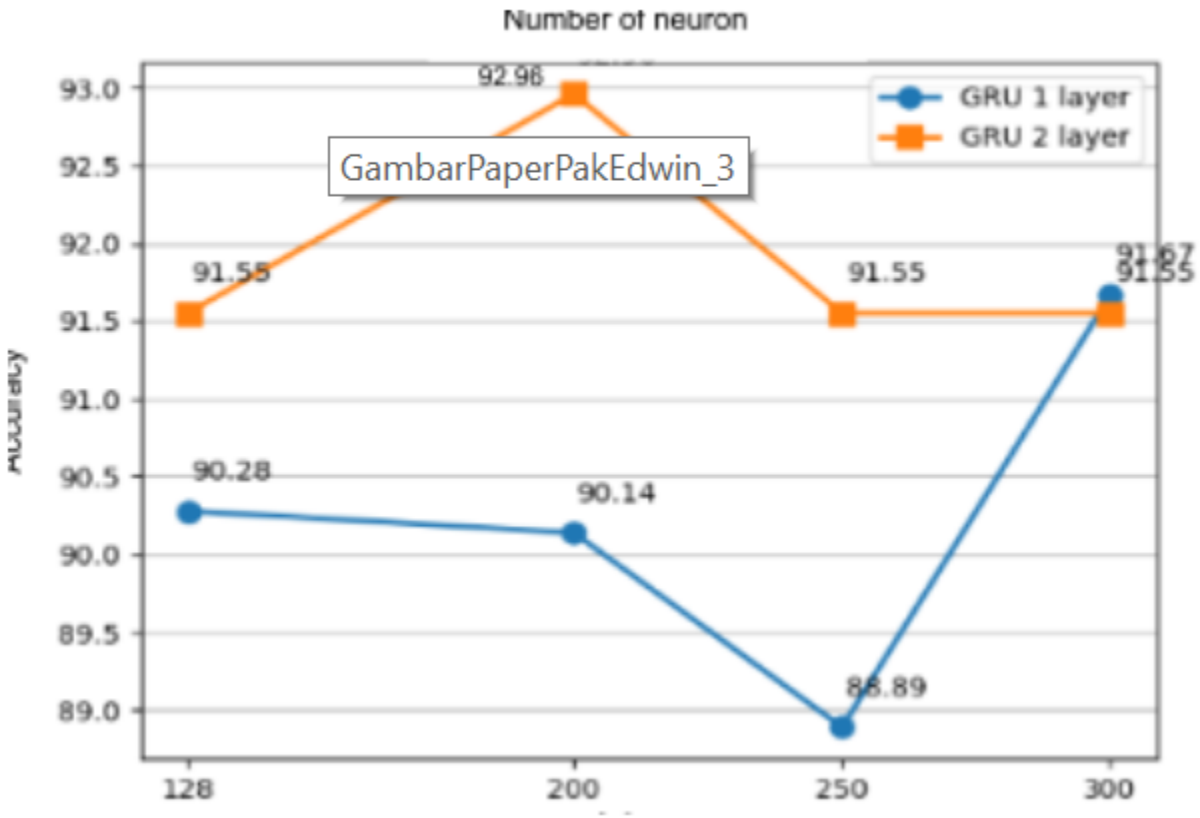
b. Learning rate 0.0001

- Hasil percobaan menggunakan learning rate 0,001 dan 0,0001 ditunjukkan pada Gambar5.
- Model dengan nilai learning rate 0,00001 belum mampu mencapai konvergen yang diharapkan
- nilai kerugian pelatihan.

- Pengurangan nilai learning rate selanjutnya menjadikan model konvergen dan nilai kerugian mendekati nol tetapi waktu yang dibutuhkan untuk pelatihan lebih lama.
- Oleh karena itu selanjutnya percobaan tingkat pembelajaran yang dipilih adalah 0,001.

B.Menentukan jumlah neuron pada lapisan tersembunyi

- Eksperimen ini digunakan untuk menentukan jumlah neuron yang optimal pada lapisan tersembunyi dengan mengatur learning rate menjadi 0,001.
- Penulis menggunakan GRU dengan arsitektur 1 dan 2. Jumlah neuron pada hidden layer yang akan diuji adalah 128, 200, 250 dan 300.
- Hasil percobaan kita dapat dilihat pada Gambar 6. Hasilnya menunjukkan bahwa penambahan atau pengurangan jumlah neuron dapat mempengaruhi keakuratan model.
- Akurasi awal GRU dengan 1 lapisan adalah 90,28% dan meningkat dengan penambahan jumlah neuron.
- Sebaliknya, akurasi yang diperoleh GRU dengan 2 layer paling tinggi dengan nilai 92,96 ketika jumlah neuron adalah 200. Penambahan jumlah neuron tidak dapat meningkatkan ketepatan



C.Performa model keseluruhan

- Tabel 4 menunjukkan kinerja terbaik dari GRU dan algoritme klasik yang digunakan dalam percobaan.
- Semua nilai dalam tabel ini adalah nilai rata-rata dari percobaan menggunakan 10-validasi cross validation pada 713 data latih.
- Performa terbaik model GRU dicapai dengan GRU dengan 2 layer, dengan learning rate 0,001, 200 neuron pada hidden layer, yang memiliki akurasi 92,96%.
- Hal ini menunjukkan bahwa kemampuan model GRU lebih baik karena
- Model GRU dibangun dengan memiliki update gate dan reset gate yang dapat menyimpan dan membuang data sebelumnya.
- Fungsi yang dimiliki oleh update gate dan reset gate menjadikan model GRU dapat mengetahui informasi waktu sebelumnya dan informasi waktu sekarang sehingga dapat meningkatkan akurasi dalam menentukan kelas pada tweet

Table 4. esult compared models

	Precision (%)	**Recall (%)**	**F-measure (%)**	**Accuracy (%)**
GRU	88.46	92.00	90.20	92,96
SVM	73.08	90.48	80.85	83.65
NB	76.92	86.96	81.63	79.80
BLR	69.23	100.00	81.82	78.84
RFDT	84.62	80.00	86.27	82.69

-

4. Kesimpulan

- Berdasarkan hasil percobaan dari penelitian ini, dapat disimpulkan bahwa Gated Recurrent Unit dengan fitur word2vec lebih baik dibandingkan dengan supervised tradisional untuk mendeteksi ujaran kebencian bahasa indonesia.
- Ekstraksi fitur menggunakan word2vec mampu menghasilkan semantic nilai untuk setiap kata dengan kata lain yang memiliki arti yang sama sehingga menghasilkan klasifikasi diperoleh cukup baik.
- Kurangnya word2vec ditemukan dalam ketergantungan word2vec pada pelatihan data, semakin banyak data pelatihan, semakin besar peluang word2vec untuk dapat mewakili semua kata-kata yang diinginkan