Project 1

Sentiment Analysis

Jensen Huang | NLP Batch 7

‘The Jensens’ Team

Ferrel

Hisyam

Imam

Nurul

Ruth

Poska

Background

- **Pemilu Presiden (Pilpres) 2019 di Indonesia** merupakan salah satu momen politik paling dinamis dan berpengaruh dalam sejarah demokrasi negara. Persaingan antara kandidat, dinamika kampanye, serta keterlibatan masyarakat dalam diskusi politik meningkat secara signifikan.
- **Twitter sebagai platform utama** digunakan oleh masyarakat untuk menyampaikan opini, dukungan, kritik, serta reaksi terhadap berbagai isu yang berkembang selama Pilpres 2019. Interaksi dalam bentuk tweet, retweet, dan hashtag mencerminkan pandangan publik secara luas.
- **Analisis sentimen memiliki peran penting** dalam memahami opini publik terhadap kandidat dan isu-isu terkait. Dengan teknik pemrosesan bahasa alami (NLP), analisis ini dapat mengungkap tren sentimen, persepsi masyarakat, serta faktor-faktor yang mempengaruhi opini publik selama periode pemilu.

Objectives

- Mengidentifikasi algoritma terbaik (**Random Forest** vs **LSTM**) untuk analisis sentimen
- Mengoptimalkan pemrosesan data teks dengan berbagai teknik **preprocessing** seperti stemming, stopwords removal, tokenization, dan normalisasi teks
- Membandingkan performa model berdasarkan matriks evaluasi seperti **akurasi**, **presisi**, **recall**, dan **F1-score** untuk memilih model terbaik

Data Understanding

1. Ukuran Dataset
   Dataset terdiri dari **1.815 baris** dan **3 kolom**

2. Fitur dan Target Variabel
   - **Fitur utama : tweet**
   - Target variabel **: sentimen**

3. Distribusi Data
   - Positif → 612 tweet
   - Netral → 607 tweet
   - Negatif → 596 tweet

4. Karakteristik Data Teks
   Tweet berasal dari media sosial, sehingga kemungkinan besar mengandung:
   - Bahasa informal & slang
   - Kesalahan ejaan & variasi kata
   - Penggunaan emoji, hashtag, mention (@username)
   - Kata-kata yang tidak memiliki arti penting (stopwords)

Algoritma Understanding

1. Random Forest
   - Model machine learning dengan banyak **decision tree**
   - Mengambil keputusan berdasarkan **voting** dari beberapa pohon
   - Cepat, sederhana, tapi **tidak memahami urutan kata**

2. Long Short-Term Memory (LSTM)
   - Model deep learning berbasis **RNN**
   - Dapat **mengingat urutan kata** dalam teks
   - Lebih akurat namun membutuhkan **lebih banyak data dan komputasi tinggi**

Kedua algoritma ini akan **diuji** untuk untuk menemukan model yang paling **efektif** dan **akurat** dalam menganalisis sentimen tweet Pilpres 2019, dengan **optimasi hyperparameter** dan **teknik preprocessing** sebagai faktor pendukung utama.

Text Preprocessing

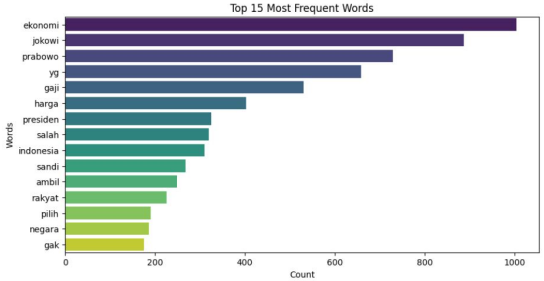
Tahapan Text Preprocessing

1. Duplicate and null removal ✅
2. Tokenization ✅
3. Stopwords removal ✅
4. Punctuation and special character removal ✅
5. Normalization (singkatan to full) ✅
6. Lemmatize using KBBI ✅
7. Stemming ✅
8. Padding ✅
9. Labelling ✅
10. Vectorization (TF-IDF dan Word2Vec) ✅

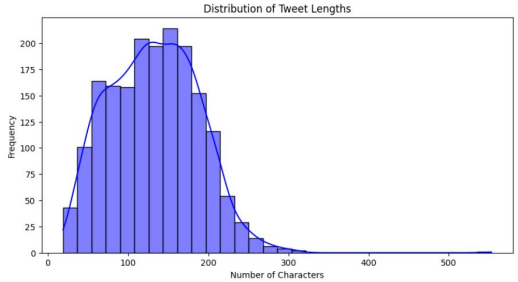
Data Analysis

Analysis

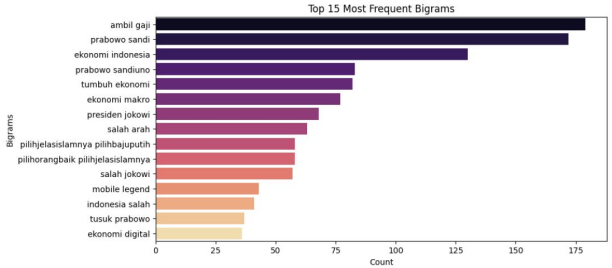
1. Most frequent words



2. Distribution of tweet lengths
   Median around 110 - 120 words



Most frequent Bigrams



Data Modelling (Algoritma yang digunakan)

Aspek	Random Forest	LSTM
Jenis Model	Machine Learning (Ensemble Learning)	Deep Learning (RNN)
Memahami Urutan Kata?	❌ Tidak	✅ Ya
Kecepatan Training	✅ Cepat	❌ Lambat
Butuh Banyak Data?	❌ Tidak terlalu	✅ Ya
Akurasi di Data Teks?	🔹 Bisa cukup baik jika preprocessing bagus	🔥 Lebih akurat jika data cukup besar
Kompleksitas	🔹 Rendah, bisa dijalankan di CPU	🔥 Tinggi, lebih baik menggunakan GPU
Kesesuaian dengan Vectorization	✅ Cocok dengan TF-IDF dan Word2Vec	❌ Tidak cocok dengan TF-IDF, ✅ Cocok dengan Word2Vec

Data Modelling (Hyperparameter Tuning)

Random Forest

1. Menggunakan **RandomizedSearchCV** untuk mempercepat pencarian parameter terbaik.
2. Parameter yang digunakan meliputi :
   a. n_estimators (Jumlah Pohon)
   b. max_depth (Kedalaman Maksimum)
   c. min_samples_split (Minimum jumlah sampel untuk membagi node)
   d. min_samples_leaf (Minimum jumlah sampel dalam satu daun)
   e. max_features(Jumlah fitur yang digunakan di setiap split.)

LSTM

1. Menggunakan **Keras Tuner** untuk mengeksplorasi arsitektur terbaik.
2. Parameter yang digunakan meliputi :
   a. Lapisan LSTM
   b. Jumlah Unit di Setiap Lapisan
   c. Dropout
   d. Kernel Regularizer
   e. Optimizer
   f. Learning Rate

Model Evaluation

Model	Accuracy Sebelum Tuning	Accuracy Setelah Tuning	Peningkatan Accuracy	Best Hyperparameters
TF-IDF + Random Forest	58.4%	60.8%	+2.4%	{'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 395}
Word2Vec + Random Forest	48.4%	60.8%	+12.4%	{'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 395}
Word2Vec + LSTM	36.8%	38.8%	+2.0%	1 Layer, Units: 128, Dropout: 0.5, L2 Regularizer: 0.001, Optimizer: rmsprop, Learning Rate: 0.00001

Model Evaluation

Model	Precision Sebelum	Precision Setelah	Recall Sebelum	Recall Setelah	F1-Score Sebelum	F1-Score Setelah
TF-IDF + Random Forest	0.60	0.61	0.59	0.61	0.58	0.61
Word2Vec + Random Forest	0.48	0.61	0.49	0.61	0.48	0.61
Word2Vec + LSTM	0.67	0.69	0.89	0.99	0.76	0.81

Conclusion

Berdasarkan hasil eksperimen, **Word2Vec + Random Forest** dipilih sebagai algoritma terbaik. Model ini mencapai akurasi tertinggi sebesar **60.8%** setelah tuning, setara dengan TF-IDF + Random Forest, tetapi lebih unggul dalam memahami hubungan semantik antar kata.

Selain itu, waktu pelatihannya lebih efisien dibandingkan Word2Vec + LSTM, yang membutuhkan sumber daya lebih besar untuk mencapai performa optimal.

Let’s Discuss!