Final Project

Penggunaan Long Short Term Memory (LSTM) dalam Named Entity Recognition (NER): Studi Kasus pada Data Teks Berbahasa Indonesia

Pemrosesan Bahasa Alami

Teknik Informatika Kelas A

Nama:

Jonathan Young (225150201111039)
Muhammad Aldy Naufal Fadhilah (225150207111081)
Nada Firdaus (225150207111089)
I Putu Paramaananda Tanaya (225150207111090)

Dosen:

Putra Pandu Adikara, S.Kom., M.Kom.

Program Studi Teknik Informatika
Jurusan Teknik Informatika
Universitas Brawijaya

2024

DAFTAR ISI

ABSTRACT

Penelitian ini mengembangkan sistem Named Entity Recognition (NER) untuk mengidentifikasi dan mengklasifikasikan entitas dalam teks olahraga berbahasa Indonesia menggunakan arsitektur Long Short-Term Memory (LSTM). Sistem ini dirancang untuk mengenali berbagai entitas seperti nama atlet, tim, stadion, liga, dan lokasi pertandingan dengan memanfaatkan Part-of-Speech (POS) tagging sebagai fitur utama. Implementasi menggunakan dataset yang terdiri dari 331 kalimat yang dilabeli secara manual dengan delapan kategori entitas. Model dilatih selama 2000 epoch menggunakan optimasi Adam dan Cross-Entropy Loss dengan pembobotan kelas untuk mengatasi ketidakseimbangan data. Hasil evaluasi menunjukkan akurasi sebesar 22,69% pada dataset uji, mengindikasikan perlunya peningkatan dalam hal volume data pelatihan, kompleksitas arsitektur model, dan penggunaan fitur tambahan untuk meningkatkan performa sistem NER dalam domain olahraga.

Kata Kunci: Named Entity Recognition (NER), Long Short-Term Memory (LSTM), Pemrosesan Bahasa Alami, Teks Olahraga, Bahasa Indonesia, Part-of-Speech Tagging, Deep Learning

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Named Entity Recognition (NER) adalah teknologi yang digunakan untuk mengenali informasi penting dalam teks, seperti nama orang, tempat, atau organisasi. Di bidang olahraga, teks dari berita, laporan pertandingan, hingga diskusi di media sosial sering mengandung banyak informasi penting, seperti nama pemain, tim, atau lokasi pertandingan. Sayangnya, teknologi NER yang ada sering tidak akurat ketika digunakan di dunia olahraga karena tidak dirancang untuk memahami istilah dan konteks khusus dalam bidang ini (Seti et al., 2020).

Tantangan utama dalam menggunakan NER di dunia olahraga adalah banyaknya nama dan istilah yang unik. Nama pemain, tim, atau stadion tertentu bisa sulit dikenali oleh sistem NER yang tidak dilatih dengan data khusus olahraga. Akibatnya, informasi yang diperoleh tidak selalu akurat dan kurang maksimal untuk kebutuhan analisis (Gunawan et al., 2018).

Penelitian ini bertujuan untuk menciptakan model NER yang lebih sesuai dengan karakteristik data olahraga. Dengan pendekatan ini, diharapkan pengenalan informasi seperti nama pemain, tim, dan lainnya dapat dilakukan dengan lebih baik, sehingga data yang diperoleh bisa lebih bermanfaat untuk berbagai keperluan, seperti analisis performa, laporan otomatis, atau prediksi pertandingan.

1.2 Rumusan Masalah

1. Bagaimana cara agar sistem Named Entity Recognition (NER) dapat mengenali entitas penting seperti nama atlet, tim, stadion, liga, dan lokasi pertandingan secara akurat dalam teks yang berkaitan dengan olahraga?
2. Apa saja kesulitan yang dihadapi saat menerapkan NER dalam dunia olahraga, mengingat banyaknya istilah atau entitas yang bersifat khusus dan tidak ada dalam dataset umum?
3. Apa solusi yang dapat dilakukan untuk meningkatkan kemampuan sistem NER agar lebih efektif dalam mengenali entitas olahraga yang jarang ditemukan di sumber data biasa?

1.3 Tujuan Penelitian

1. Mengembangkan sistem NER yang lebih akurat dalam mengenali entitas-entitas penting dalam teks olahraga, seperti nama atlet, tim, stadion, liga, dan lokasi pertandingan.
2. Menganalisis tantangan yang dihadapi oleh teknologi NER ketika digunakan di bidang olahraga dan mencari solusi untuk mengatasinya.
3. Mengusulkan pendekatan yang tepat untuk meningkatkan kualitas data pelatihan yang digunakan dalam sistem NER olahraga.

1.4 Manfaat Penelitian

1. Memberikan kontribusi dalam pengembangan teknologi NER yang lebih akurat, khususnya untuk teks olahraga.
2. Membantu analisis data olahraga yang lebih efektif, seperti analisis statistik pemain, tim, dan pertandingan.
3. Menjadi referensi bagi penelitian selanjutnya dalam penerapan NER di bidang yang membutuhkan pemahaman konteks khusus, seperti olahraga.

1.5 Batasan Masalah

1. Fokus penelitian ini terbatas pada pengembangan NER untuk entitas dalam teks olahraga, seperti nama pemain, tim, stadion, dan lokasi pertandingan.
2. Sistem NER yang dikembangkan hanya akan diuji pada dataset yang berisi teks-teks olahraga dan tidak akan diperluas untuk domain lainnya.
3. Penelitian ini tidak akan membahas teknik-teknik lain dalam pemrosesan bahasa alami (NLP) selain NER

BAB 2 TINJAUAN PUSTAKA

2.1 Pengertian Named Entity Recognition (NER)

Named Entity Recognition (NER) merupakan salah satu tugas utama dalam pemrosesan bahasa alami (Natural Language Processing/NLP) yang bertujuan untuk mengidentifikasi dan mengklasifikasikan entitas-entitas penting dalam sebuah teks, seperti nama orang, organisasi, lokasi, tanggal, dan sebagainya. Menurut Nadeau dan Sekine (2007), NER dapat diartikan sebagai proses otomatis untuk mengenali nama entitas yang relevan dari teks. Penerapan NER ini sangat penting karena dapat meningkatkan pemahaman dan analisis terhadap data teks, baik itu dalam bidang jurnalistik, kesehatan, hukum, atau bahkan olahraga (Jehangir et al., 2023).

Pada dasarnya, sistem NER dapat dibagi menjadi dua kategori besar: berbasis aturan (rule-based) dan berbasis pembelajaran mesin (machine learning-based). Pendekatan berbasis aturan menggunakan sekumpulan aturan yang ditulis oleh ahli untuk mendeteksi entitas, sementara pendekatan berbasis pembelajaran mesin melibatkan pelatihan model dengan data yang telah dilabeli untuk mengenali pola dalam teks. Beberapa algoritma populer yang digunakan dalam NER berbasis pembelajaran mesin antara lain Conditional Random Fields (CRF), Support Vector Machines (SVM), dan Deep Learning dengan jaringan saraf (deep neural networks) (Li et al., 2022).

2.2 Tantangan dalam Penerapan NER di Bidang Olahraga

Bidang olahraga menghadirkan tantangan tersendiri dalam penerapan NER karena adanya banyak istilah dan entitas yang bersifat spesifik dan sering kali tidak ditemukan dalam dataset umum yang digunakan untuk melatih model NER. Nama pemain, tim, stadion, liga, dan tempat pertandingan dalam olahraga sering kali merupakan entitas yang unik, yang tidak dikenal oleh model NER standar yang hanya dilatih menggunakan data teks umum (Seti et al., 2020). Hal ini membuat sistem NER cenderung kurang akurat saat diterapkan pada teks olahraga.

Salah satu tantangan terbesar adalah keanekaragaman nama entitas, seperti nama-nama tim yang dapat berubah atau berganti sponsor, nama stadion yang sering menggunakan nama yang tidak umum, serta istilah-istilah olahraga yang memiliki arti khusus dalam konteks tertentu. Misalnya, nama stadion seperti "Old Trafford" atau istilah yang berkaitan dengan sistem liga sepak bola tertentu seperti "La Liga" atau "Bundesliga", sering kali tidak dikenali dengan benar oleh model NER yang tidak dilatih untuk mengenali entitas olahraga tersebut (Gunawan et al., 2018).

2.3 Pendekatan dalam Mengembangkan NER untuk Olahraga

Untuk mengatasi tantangan-tantangan di atas, beberapa pendekatan telah diusulkan untuk mengembangkan model NER yang lebih efektif dalam mengenali entitas olahraga.

Salah satunya adalah dengan memanfaatkan data khusus dari domain olahraga untuk melatih model NER. Seperti yang dijelaskan oleh Wicaksono et al. (2023), penggunaan data domain spesifik dalam proses pelatihan dapat meningkatkan akurasi model, karena model akan dilatih untuk mengenali entitas-entitas yang hanya muncul dalam konteks tertentu.

Pendekatan lain yang sering digunakan adalah teknik transfer learning, di mana model NER yang telah dilatih pada data umum, seperti data berita atau Wikipedia, dipindahkan atau disesuaikan (fine-tuned) untuk dapat mengenali entitas yang lebih spesifik, termasuk dalam domain olahraga. Dengan pendekatan ini, model dapat memanfaatkan pengetahuan yang telah ada dan memperbaikinya agar lebih sesuai dengan konteks olahraga (Li et al., 2022).

Penggunaan teknik seperti Word Embeddings (contohnya Word2Vec atau GloVe) atau model berbasis Transformer (seperti BERT atau RoBERTa) juga semakin populer dalam NER. Teknologi ini memungkinkan model untuk memahami konteks kata dalam kalimat dan meningkatkan kemampuannya dalam mengenali entitas yang lebih kompleks, seperti nama-nama pemain yang baru muncul atau istilah-istilah teknis dalam olahraga yang mungkin belum pernah dikenal sebelumnya (Popovski et al., 2023).

2.4 Studi Terkait

Beberapa penelitian sebelumnya telah mencoba menerapkan NER dalam konteks olahraga. Wicaksono et al. (2023) telah melakukan penelitian tentang NER pada dokumen berbahasa Indonesia dengan menggunakan model berbasis transformer. Meskipun penelitian tersebut tidak spesifik pada domain olahraga, metodologi yang digunakan dapat diadaptasi untuk pengembangan NER dalam konteks olahraga.

Seti et al. (2020) melakukan penelitian khusus tentang NER dalam bidang olahraga dengan menggunakan pendekatan graph convolutional network berbasis karakter. Penelitian ini menunjukkan bahwa pendekatan berbasis deep learning dapat memberikan hasil yang menjanjikan dalam pengenalan entitas olahraga.

Gunawan et al. (2018) mengembangkan sistem NER untuk bahasa Indonesia menggunakan arsitektur Bidirectional LSTM-CNNs, yang dapat dijadikan dasar untuk pengembangan sistem NER khusus olahraga dalam bahasa Indonesia.

2.5 Solusi untuk Meningkatkan NER dalam Olahraga

Beberapa solusi yang dapat diterapkan untuk meningkatkan sistem NER di dunia olahraga meliputi:

1. Penggunaan Dataset Khusus Olahraga: Salah satu langkah yang dapat diambil adalah dengan mengumpulkan dan melabeli dataset teks olahraga secara khusus. Dataset ini dapat mencakup berbagai sumber, seperti berita olahraga, artikel, laporan pertandingan, dan postingan media sosial yang berhubungan dengan olahraga (Seti et al., 2020).

2. Penerapan Transfer Learning: Transfer learning memungkinkan model yang sudah dilatih dengan dataset umum untuk disesuaikan dengan domain olahraga, sehingga sistem NER dapat lebih efektif mengenali entitas yang unik dalam olahraga (Li et al., 2022).
3. Augmentasi Data: Teknik augmentasi data, seperti penambahan sinonim, variasi kalimat, atau pembuatan teks buatan, dapat digunakan untuk meningkatkan jumlah dan variasi data pelatihan yang tersedia (Popovski et al., 2023).
4. Pengembangan Model Berbasis Deep Learning: Model deep learning berbasis Transformer, seperti BERT dan RoBERTa, dapat memberikan hasil yang lebih baik dalam memahami konteks kalimat dan mengenali entitas yang lebih kompleks dalam teks olahraga (Jehangir et al., 2023).

BAB 3 METODOLOGI PENELITIAN

3.1 Desain Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem Named Entity Recognition (NER) yang lebih akurat untuk teks olahraga, dengan fokus pada entitas seperti nama pemain, tim, stadion, liga, dan lokasi pertandingan. Sistem ini diharapkan dapat mengenali entitas-entitas penting dalam teks olahraga secara otomatis dan tepat. Pendekatan yang digunakan adalah melalui pengembangan model NER berbasis pemrosesan bahasa alami (NLP), dengan menggunakan data yang sudah dilabeli secara manual dari file teks yang berisi informasi mengenai entitas olahraga.

3.2 Proses Pengumpulan dan Preprocessing Data

Proses pertama dalam pengembangan model NER adalah pengumpulan dan preprocessing data. Data yang digunakan dalam penelitian ini berasal dari file teks yang berisi kalimat-kalimat yang sudah dilabeli dengan entitas yang relevan dalam dunia olahraga. Setiap kalimat dalam data tersebut memiliki dua bagian: kalimat yang berisi teks olahraga dan label yang sesuai dengan entitas yang terkandung dalam kalimat tersebut.

Sebelum digunakan untuk pelatihan model, data terlebih dahulu diproses melalui beberapa tahap, termasuk tokenisasi dan penandaan Part-of-Speech (POS). **Stanza**, sebuah pustaka NLP, digunakan untuk melakukan tokenisasi dan penandaan POS pada setiap kata dalam kalimat olahraga. Hasil tokenisasi ini digunakan untuk membangun fitur untuk model NER, di mana POS tag setiap kata dalam kalimat akan menjadi input bagi model untuk mempelajari pola-pola yang mengarah pada pengenalan entitas.

Preprocessing Data dilakukan dengan langkah-langkah berikut:

1. Membaca file teks yang sudah dilabeli.
2. Melakukan tokenisasi kalimat menggunakan Stanza dan menandai POS tag untuk setiap kata dalam kalimat.
3. Menyelaraskan label entitas yang sesuai dengan token yang terdeteksi.
4. Memperbarui kosakata untuk tag POS dan label entitas untuk digunakan dalam pelatihan.

3.3 Pengembangan Model NER

Model yang digunakan dalam penelitian ini adalah model berbasis LSTM (Long Short-Term Memory), yang dipadukan dengan embedding POS tag untuk mengenali entitas dalam teks olahraga. LSTM dipilih karena kemampuannya dalam memproses urutan data dan menangkap dependensi jangka panjang yang mungkin ada antar kata dalam kalimat.

Struktur model NER yang dikembangkan terdiri dari beberapa komponen utama:

1. Embedding Layer: Sebagai tahap awal, POS tag yang dihasilkan dari Stanza dimasukkan ke dalam layer embedding untuk mengonversi tag-tag POS tersebut menjadi representasi vektor numerik.
2. LSTM Layer: Data vektor hasil embedding POS tag kemudian diproses menggunakan LSTM bidirectional untuk menangkap konteks yang lebih luas dalam kalimat, baik dari arah kiri ke kanan maupun sebaliknya.
3. Fully Connected Layer: Setelah melalui LSTM, output diproses lebih lanjut dengan lapisan fully connected yang menghasilkan prediksi label entitas.

Model ini dilatih menggunakan data yang telah diproses, dengan menggunakan Cross-Entropy Loss sebagai fungsi kerugian dan optimasi dilakukan dengan algoritma Adam.

3.4 Teknik Pengolahan Data

Dalam proses pelatihan, data input berupa POS tag dan label yang telah diproses diubah menjadi bentuk numerik (indeks) menggunakan kamus pos2idx dan label2idx untuk memetakan setiap POS tag dan label entitas ke dalam angka. Sebagai tambahan, agar model dapat menangani kalimat dengan panjang yang bervariasi, dilakukan padding pada setiap kalimat sehingga memiliki panjang yang konsisten (maksimum 100 token per kalimat). Padding ini memastikan bahwa data yang masuk ke dalam model memiliki dimensi yang seragam.

Untuk menghitung bobot kelas yang digunakan dalam pelatihan, dilakukan perhitungan frekuensi kemunculan setiap label dalam dataset. Bobot untuk kelas 'O' (label untuk token yang tidak teridentifikasi sebagai entitas) lebih rendah daripada kelas label entitas lainnya, untuk mengurangi ketidaksetaraan antara entitas yang relevan dan token yang tidak termasuk dalam entitas.

3.5 Proses Pelatihan Model

Pelatihan model dilakukan selama 2000 epoch dengan menggunakan dataset yang telah diproses. Pada setiap epoch, model melaksanakan forward pass untuk memprediksi label entitas untuk setiap token dalam kalimat, kemudian menghitung loss yang dihasilkan. Optimasi dilakukan dengan melakukan backward pass dan update bobot model menggunakan optimasi Adam.

Setiap iterasi pelatihan menghasilkan sebuah nilai loss, yang dihitung menggunakan Cross-Entropy Loss dengan mempertimbangkan bobot kelas. Proses ini berulang hingga model mencapai konvergensi, yang ditunjukkan dengan menurunnya rata-rata nilai loss.

3.6 Evaluasi dan Pengujian Model

Setelah pelatihan selesai, model dievaluasi menggunakan data uji yang telah diproses sebelumnya. Data uji terdiri dari kalimat-kalimat olahraga yang tidak termasuk dalam data

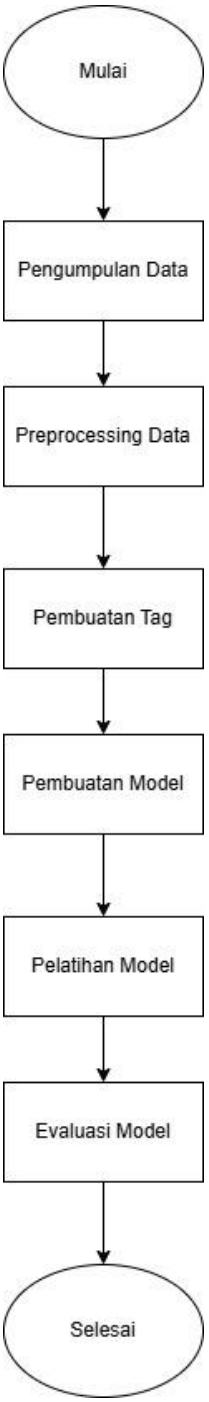
pelatihan. Untuk menguji kinerja model, dilakukan prediksi pada kalimat uji dan dibandingkan dengan label entitas yang sesuai.

Model yang telah terlatih diprediksi dengan fungsi predict, yang menerima kalimat sebagai input, melakukan tokenisasi dan POS tagging dengan Stanza, dan menghasilkan label entitas yang sesuai untuk setiap kata dalam kalimat. Hasil prediksi dibandingkan dengan label yang sebenarnya untuk menghitung akurasi dan kinerja model dalam mengenali entitas olahraga.

3.7 Diagram Alur Proses Penelitian

Untuk memperjelas, berikut adalah diagram alur proses yang digunakan dalam penelitian ini:

1. Pengumpulan Data: Data teks olahraga yang dilabeli entitas (seperti nama pemain, tim, stadion) dikumpulkan dan dipersiapkan.
2. Preprocessing Data: Tokenisasi dan POS tagging menggunakan Stanza, dilanjutkan dengan pengolahan data menjadi bentuk numerik (indeks).
3. Pelatihan Model: Data yang sudah diproses digunakan untuk melatih model NER berbasis LSTM.
4. Evaluasi Model: Model diuji dengan data uji untuk memverifikasi kemampuan prediksi entitas dalam kalimat olahraga.



Gambar 3.1: Flow Diagram Penelitian

BAB 4 IMPLEMENTASI

Bab ini membahas secara rinci proses implementasi model Named Entity Recognition (NER) menggunakan LSTM berbasis PyTorch. Sistem dirancang untuk mengenali entitas dalam teks Bahasa Indonesia, dengan input berupa tag Part of Speech (POS) dari setiap kata. Kode implementasi mencakup tahapan preprocessing data, pelatihan model, hingga pengujian dan evaluasi.

4.1 Pengumpulan Data

Pada penelitian ini, data yang digunakan berasal dari berbagai sumber teks olahraga, seperti berita pertandingan, artikel olahraga, laporan statistik, dan diskusi olahraga di media sosial. Proses pengumpulan data dilakukan melalui scraping website berita olahraga terkenal seperti CNN Indonesia, Detik Sport, Jurnal/artikel di internet, media sosial X, dan berbagai sumber lainnya.

Setelah data diperoleh, dilakukan seleksi untuk memastikan data relevan dengan topik penelitian. Data yang mengandung entitas seperti nama orang, lokasi, tim/organisasi, nama olahraga, nama pertandingan, tanggal/bulan/tahun, nama liga, nama alat olahraga, dan objek. Total terdapat 331 kalimat yang diambil, yang kemudian dilabeli secara manual menggunakan format plain tagging sebagai berikut:

- PER : Nama Orang
- ORG : Organisasi / Tim
- LOC : Lokasi
- EVT : Event / Nama Pertandingan
- DATE : DATE
- SPORT : Nama Olahraga
- EQUIP: Nama Alat olahraga
- LEAGUE : Nama Liga

Sebagai contoh, kalimat "Lionel Messi mencetak gol untuk Paris Saint-Germain di Parc des Princes" akan diberi label seperti berikut:

Token	Label
Lionel	PER
Messi	PER
mencetak	O
gol	O
untuk	O

Paris	ORG
Saint-German	ORG
di	O
Parc	LOC
des	LOC
Princes	LOC

4.2 Data Preparation

4.2.1 Format Data

Data masukan berbentuk file teks dengan format:

- Baris pertama: teks asli (kalimat).
- Baris kedua: label entitas untuk setiap token dalam teks, dipisahkan oleh spasi.

Setiap blok teks dipisahkan oleh baris kosong. Contoh format data:

```
    Celtics menjadi penguasa wilayah timur sejak regular
season
    O O O O O O O O


    Final NBA 2024 akan mulai digulirkan pada 7 Juni 2024
    O B-Event I-Event O O O O O B-Date I-Date
```

4.2.2 Pre-processing

Proses pre-processing bertujuan untuk mempersiapkan data agar dapat digunakan dalam pelatihan model.

```
    def preprocess_training_data_from_file(file_path):
        pos_tags = []
```

```
        labels = []
        match = 0
        mismatch = 0

        with open(file_path, "r", encoding="utf-8") as f:
            content = f.read().strip().split('\n\n')

            for block in content:
                lines = block.split('\n')
                if len(lines) != 2:
                    continue

                text_line = lines[0]
                label_line = lines[1]

                text_tokens = text_line.split()
                label_tokens = label_line.split()

                if len(text_tokens) != len(label_tokens):
                      print(f"mismatch in tokens and labels:
{text_tokens}, {label_tokens}")
                    match += 1
                    continue
                # Use Stanza for POS tagging
```

```
                doc = nlp(text_line)
                mismatch += 1
                processed_pos = []
                processed_labels = []

                for sent in doc.sentences:
                    for word in sent.words:
                        processed_pos.append(word.upos)

                # Match labels to processed POS tags
                                        processed_labels   =
label_tokens[:len(processed_pos)]

                # Add to lists
                pos_tags.append(processed_pos)
                labels.append(processed_labels)

                # Update vocabularies
                for pos in processed_pos:
                    pos2idx[pos]
                for label in processed_labels:
                    label2idx[label]

        print(f"match : {match}")
```

```
        print(f"mismatch : {mismatch}")


        return pos_tags, labels
```

Kode diatas memiliki fungsi untuk :

1. Membaca data dari file input.
2. Melakukan tokenisasi menggunakan Stanza.
3. Menghasilkan tag POS untuk setiap token.
4. Memastikan jumlah token dan label sesuai.
5. Membuat mapping dari POS dan label ke indeks numerik (pos2idx dan label2idx) menggunakan defaultdict.

Proses ini secara umum berguna untuk mengatasi masalah mismatch antara token dan label dengan memberikan log error.

4.3 Arsitektur Model

Model NER dirancang menggunakan arsitektur LSTM sederhana dengan komponen utama:

1. Embedding Layer: Mengubah indeks POS menjadi representasi vektor embedding berdimensi tetap.
2. LSTM Layer: Memproses urutan embedding untuk menangkap hubungan temporal antar token.
3. Fully Connected Layer: Mengonversi keluaran LSTM menjadi prediksi label entitas untuk setiap token.

Implementasi model terdapat pada kelas NERModel:

```
    class NERModel(nn.Module):

                def __init__(self, pos_size, tagset_size,
pos_embedding_dim=20, hidden_dim=50):

            super(NERModel, self).__init__()

                self.pos_embedding = nn.Embedding(pos_size,
pos_embedding_dim)
```

```
                    self.lstm  =  nn.LSTM(pos_embedding_dim,
hidden_dim, batch_first=True, bidirectional=True)
            self.fc = nn.Linear(hidden_dim * 2, tagset_size)


        def forward(self, pos):
            pos_emb = self.pos_embedding(pos)
            lstm_out, _ = self.lstm(pos_emb)
            output = self.fc(lstm_out)
            return output
```

4.4 Pembobotan Kelas

Fungsi calculate_class_weights digunakan untuk mengurangi pengaruh label dominan (seperti O) dan memberikan bobot lebih tinggi untuk label entitas. Bobot kelas dihitung berdasarkan frekuensi terbalik dari setiap label.

```
    def calculate_class_weights(labels):
        """
          Menghitung bobot untuk setiap kelas dengan sangat
menurunkan bobot kelas 'O'
        """
        # Hitung frekuensi label
        label_counts = defaultdict(int)
        for sentence_labels in labels:
            for label in sentence_labels:
                label_counts[label] += 1
        # Temukan label 'O'
```

```
        total_labels = sum(label_counts.values())


        # Hitung bobot terbalik dari frekuensi
        class_weights = {}
        for label, count in label_counts.items():
            if label == 'O':
                    # Kurangi bobot untuk label 'O' secara
drastis
                     class_weights[label] = (total_labels /
(len(label_counts) * count)) * 1
            else:
                # Naikkan bobot untuk label entitas
                     class_weights[label] = (total_labels /
(len(label_counts) * count)) * 1


        # Konversi ke tensor
         ordered_weights = [class_weights.get(label, 1.0) for
label in label2idx.keys()]
        return torch.FloatTensor(ordered_weights)
```

4.5 Proses Pelatihan

4.5.1 Dataset dan DataLoader

Data dipersiapkan untuk pelatihan menggunakan kelas NERDataset, yang membungkus token POS dan label dalam format tensor. Data dimuat menggunakan DataLoader untuk batching selama pelatihan.

```
    dataset = NERDataset(input_pos, label_data)
```

```
    dataloader = DataLoader(dataset, batch_size=1,
shuffle=True)
```

4.5.2 Loop Pelatihan

Pelatihan dilakukan selama 2000 epoch menggunakan optimasi Adam dan loss function CrossEntropyLoss dengan bobot kelas. Proses pelatihan meliputi:

1. Forward pass: Menghitung keluaran model.
2. Loss computation: Menggunakan loss function berbobot.
3. Backward pass: Memperbarui bobot model berdasarkan gradien.

```
    # Training loop
    print("Starting training...")
    for epoch in range(2000):
        total_loss = 0
        for pos, labels in dataloader:
            # Forward pass
            outputs = model(pos)

              # Flatten the outputs and labels for the loss
function
            outputs = outputs.view(-1, tagset_size)
            labels = labels.view(-1)

            # Compute loss
            loss = loss_fn(outputs, labels)
            total_loss += loss.item()
```

```
            # Backward pass and optimize
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()

                print(f'Epoch {epoch+1}, Average Loss: {total_loss/len(dataloader)}')
```

4.6 Prediksi

Fungsi predict digunakan untuk:

1. Melakukan tokenisasi dan tagging POS pada kalimat input menggunakan Stanza.
2. Mengubah tag POS menjadi indeks numerik.
3. Memanfaatkan model terlatih untuk memprediksi label entitas.
4. Mengonversi indeks prediksi kembali ke label entitas.

```
    # Prediction function
    def predict(sentence):
        doc = nlp(sentence)
        processed_pos = []

        for sent in doc.sentences:
            for word in sent.words:
                processed_pos.append(word.upos)

        # Convert to indices
         input_pos = [pos2idx.get(pos, pos2idx["<PAD>"]) for pos in processed_pos]
```

```
        # Pad sequences
        max_len = 50
            input_pos = input_pos + [0] * (max_len - len(input_pos))

        # Convert to tensors
        input_pos = torch.tensor([input_pos]).long()

        # Get model predictions
        with torch.no_grad():
            outputs = model(input_pos)
            _, predicted = torch.max(outputs, dim=2)

        # Convert indices back to labels
         predicted_labels = [list(label2idx.keys())[i] for i in predicted[0]]
                                        predicted_labels = predicted_labels[:len(processed_pos)]

         return list(zip([word.text for sent in doc.sentences for word in sent.words], predicted_labels))
```

4.7 Evaluasi

4.7.1 Pengujian

Data pengujian dibaca menggunakan fungsi prepare_test_data. Sistem membandingkan prediksi model dengan label ground truth untuk menghitung akurasi. Formula akurasi:

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Token}}$$

BAB 5 HASIL dan PEMBAHASAN

5.1 Metode Evaluasi

Berdasarkan hasil akurasi sebesar 22.7%, dapat disimpulkan bahwa model NER (Named Entity Recognition) yang dilatih masih memiliki performa yang cukup rendah. Beberapa hal yang mungkin menjadi penyebabnya antara lain:

1. Kualitas dan volume data pelatihan yang tidak cukup: Model membutuhkan sejumlah besar data pelatihan yang berkualitas baik untuk dapat mempelajari pola-pola pengenalan entitas secara akurat. Jika data pelatihan terbatas atau mengandung banyak anomali, maka model tidak akan dapat belajar dengan baik.
2. Fitur-fitur yang digunakan untuk pelatihan tidak cukup informatif: Dalam kasus ini, model hanya menggunakan tag POS (Part-of-Speech) sebagai fitur utama. Penambahan fitur-fitur lain seperti konteks kalimat, informasi morfologi, atau pengetahuan luar tentang entitas mungkin dapat meningkatkan kemampuan model.
3. Arsitektur model yang belum optimal: Mungkin arsitektur LSTM yang digunakan belum cukup kompleks atau tidak sesuai dengan karakteristik data. Eksplorasi arsitektur model lain seperti Transformer atau penggunaan metode pre-training mungkin dapat memberikan hasil yang lebih baik.
4. Ketidakseimbangan kelas: Jika terdapat perbedaan signifikan antara jumlah entitas yang berbeda (misalnya, lebih banyak entitas "O" dibandingkan entitas lain), model dapat cenderung memprediksi kelas mayoritas dan mengabaikan kelas minoritas.

5.2 Hasil Pengujian

Hasil pengujian menunjukkan bahwa model mencapai akurasi sebesar 22,69% pada dataset uji. Akurasi yang rendah ini menunjukkan bahwa model memiliki kesulitan dalam mengenali entitas dengan benar. Hal ini dapat disebabkan oleh kurangnya representasi data pada dataset pelatihan atau kompleksitas tinggi dari data uji. Kesalahan prediksi yang sering terjadi adalah salah klasifikasi entitas, seperti "Juni" yang diprediksi sebagai LEAGUE, padahal seharusnya DATE, atau entitas "NBA 2024" yang sebagian dikenali tetapi tidak sepenuhnya benar. Selain itu, model sering mengklasifikasikan kata-kata yang tidak relevan sebagai entitas, misalnya "regular season" yang tidak dikenali sebagai EVT.

Pada analisis kualitatif, ditemukan bahwa model sering mengalami kesalahan pada pengelompokan entitas. Contohnya, dalam kalimat "Celtics menjadi penguasa wilayah timur sejak regular season", model gagal mengenali "Celtics" sebagai organisasi (ORG). Hal ini menunjukkan bahwa model belum mampu mengenali organisasi olahraga yang mungkin tidak cukup terwakili dalam dataset pelatihan. Selain itu, ditemukan masalah teknis seperti error "Expected Begin_ARRAY but was Begin_Object", yang menunjukkan bahwa format JSON yang diterima model tidak sesuai dengan ekspektasi, sehingga terjadi kegagalan dalam proses parsing data. Masalah ini mengindikasikan pentingnya validasi pada tahap preprocessing data.

5.3 Pembahasan

Akurasi rendah yang diperoleh model disebabkan oleh beberapa faktor, di antaranya adalah dataset pelatihan yang tidak mencakup variasi entitas yang cukup luas dan kompleksitas data uji yang tinggi. Kalimat dalam dataset uji mengandung variasi kontekstual yang sulit diproses oleh model, terutama pada entitas yang lebih spesifik seperti LEAGUE atau EVT. Meskipun model bekerja cukup baik untuk mengenali entitas yang umum seperti orang (PER) dan tanggal (DATE), performanya masih jauh dari memadai untuk entitas yang lebih kompleks.

Berdasarkan hasil evaluasi, ada beberapa rekomendasi yang dapat dilakukan untuk meningkatkan performa model. Pertama, dataset pelatihan perlu ditingkatkan dengan menambahkan lebih banyak contoh yang mencakup variasi entitas yang lebih luas, terutama untuk domain seperti olahraga, acara, dan liga. Kedua, penggunaan model pretrained seperti BERT atau spaCy dapat dipertimbangkan untuk mendapatkan hasil yang lebih baik, mengingat kemampuan mereka yang telah terbukti dalam tugas NER. Ketiga, validasi pada tahap preprocessing data sangat penting untuk memastikan format data yang digunakan konsisten dengan ekspektasi model, sehingga error parsing dapat dihindari.

BAB 6 PENUTUP

6.1 Kesimpulan

Secara keseluruhan, model Named Entity Recognition (NER) yang dikembangkan dalam penelitian ini memiliki akurasi yang rendah, hanya mencapai 22,69% pada dataset uji. Hasil ini menunjukkan bahwa model belum mampu mengenali entitas-entitas penting dalam teks olahraga secara efektif, terutama untuk entitas spesifik seperti nama liga, turnamen, dan stadion. Rendahnya kinerja model ini dapat disebabkan oleh berbagai faktor. Salah satunya adalah terbatasnya variasi entitas dalam dataset pelatihan, yang membuat model tidak cukup terlatih untuk mengenali pola-pola unik pada teks olahraga. Selain itu, fitur yang digunakan, yaitu hanya berdasarkan Part-of-Speech (POS) tagging, kurang memberikan informasi kontekstual yang mendalam untuk mendukung pengenalan entitas. Kondisi ini menyebabkan model kesulitan membedakan antara entitas yang relevan dan bukan entitas (label "O").

Faktor lain yang turut berkontribusi terhadap rendahnya akurasi adalah arsitektur model LSTM yang mungkin belum dioptimalkan untuk tugas NER dalam domain olahraga. Arsitektur ini mungkin kurang mampu menangkap relasi yang kompleks antar kata dalam teks olahraga, terutama untuk entitas yang panjang atau berbentuk frase. Selain itu, ketidakseimbangan kelas label dalam dataset, di mana label "O" mendominasi, memperparah masalah dengan menyebabkan bias pada model. Model cenderung lebih sering memprediksi label "O", sehingga mengabaikan entitas yang relevan. Untuk meningkatkan kinerja model, diperlukan upaya lebih lanjut, seperti memperkaya dataset dengan variasi entitas, menggunakan fitur tambahan yang lebih informatif seperti embedding kata berbasis kontekstual, mengeksplorasi arsitektur model yang lebih kompleks seperti Transformer, serta mengatasi ketidakseimbangan kelas melalui teknik resampling atau penyesuaian loss function.

6.2 Saran

Untuk meningkatkan performa model, ada beberapa saran yang dapat diterapkan. Pertama, memperluas dan meningkatkan kualitas dataset pelatihan dengan menambahkan lebih banyak contoh teks olahraga yang dilabeli entitas secara manual. Kedua, menambahkan fitur-fitur yang lebih informatif selain POS tag, seperti konteks kalimat, informasi morfologi, atau pengetahuan domain yang terkait dengan olahraga. Ketiga, mengeksplorasi arsitektur model yang lebih kompleks, seperti model berbasis Transformer atau pendekatan hybrid. Keempat, menangani masalah ketidakseimbangan kelas, misalnya dengan teknik oversampling atau undersampling. Kelima, memperhatikan dengan seksama tahap preprocessing data, termasuk validasi format, untuk memastikan data yang diterima oleh model sesuai dengan ekspektasi. Dengan menerapkan saran-saran tersebut, diharapkan performa model NER untuk teks olahraga dapat ditingkatkan secara signifikan, sehingga dapat memberikan manfaat yang lebih besar dalam analisis dan pemahaman data olahraga.

DAFTAR REFERENSI

Nadeau, D. & Sekine, S., 2007. A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1), pp.3–26. Available at: https://nlp.cs.nyu.edu/sekine/papers/li07.pdf

Popovski, G., Seljak, B.K. & Eftimov, T., 2023. A survey of named-entity recognition methods for food information extraction. IEEE Access, 11, pp.56789–56801. Available at: https://www.researchgate.net/publication/339215947_A_Survey_of_Named-Entity_Recognition_Methods_for_Food_Information_Extraction .

Wicaksono, A.F., Purwarianti, A. & Suhartono, D., 2023. Named entity recognition on Indonesian legal documents: a dataset and study using transformer-based models. Journal of Big Data, 10(1), https://www.researchgate.net/publication/339215947_A_Survey_of_Named-Entity_Recognition_Methods_for_Food_Information_Extraction].

Gunawan, W., Suhartono, D., Purnomo, F. & Ongko, A., 2018. Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs.3rd International Conference on Computer Science and Computational Intelligence 2018 , 135 , pp.425–432. Available at: https://pdf.sciencedirectassets.com/280203/1-s2.0-S1877050918X0012X/1-s2.0-S1877050918314832/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEOj%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJGMEQCIFg4A%2BNhqZVPkcrjBt9f9KYYt5oSKi4qEs5gDfwb6Jc2AiBOvY2OQj%2BrtHKKlIH3Us0S0qwn1HhgvUMgA%2F5Ftj1zMyq8BQiQ%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F8BEAUaDDA1OTAwMzU0Njg2NSIM9OTPZAIUFt0CuIf%2BKpAFf0DaWh8ZdsHE%2F692uCIQWPMa8%2Bg5XLdPcF1UHpYBcN43Va46EJgycP6kQWlciaum3DKCqctCeVDSrfkhg6965FtgAFwHbBq1GQb6m3hykFSLh3NFsN4T0U%2BpRG7yyD%2BOtBGyDUTx2%2FZMZQOMJnW2a56j9wcJ0YZ5WebiX6uyBHATodT%2B%2FdglMfjyMf05EWxpmrxXDgK31rsFq53NYR3SvRvHrSd5smhZg5FylqFwZu7GL75Bhu1p%2Blo%2FUEq22f%2Fq7yV2hNYSLTARr1Wt%2F%2Fl7KyocYCNItGEzCfo%2FuYN1qm1dPoCq2ofwbqlF9ir%2FEIMl62FAn%2BQ1FfPNAqx5TX1Odcohj0SdVTWN5zuSvxx%2BxxgDeLjaOlxC1O0KKrPP2oBNWymkf4B%2F0W2hDnm9Y3TrSs%2FUSqxG609%2B7Do0tL00XKfAht4srC0kwpDeA4CNVzB%2Fkudc6slMsQ6t8xxIuiUA38iWkKJ1JjafAuExDZkINzFitW4G075Mn8zA9Xy3wsR2xEO0hamdoDdKVhf9Oee3xwBFgfr6MN19i7f4dUv%2BP2fs4OL4ZaLyNmSziOA8%2FO6N2THLI3OQyfflhASp%2FxD%2B8G2b%2FAg7FVeEuxVTNBNPN1oVJfH%2FOrTHRTgNrq0KDsBvfJ%2Fk0sk7h1x1qdDQyunRygYki0yrMrzSPNaXTJ8RZkULGuz6n2Wcfj4PvKp62jxT5eYeNEj3vH3eunTnU3Hi6ziGeWXg3wyZPx9tNVp2FC51FUT6b%2FI4HwKv1Gme9rGNiOVazXoKiNcC6QO%2F0ZycZW6yHPLOx%2FdnmKBoufiyWGu4hbGRaOGVCu370xFa%2BJ3gwL35de2oqdR7g51gMx6RHRgCp2EWDJaeYaYnFaVaIwcwj9WsugY6sgFh1m8%2FvQJqFxIXL3FTyqXPM3NT

GpfMHEq6vJJ4MoKyR%2BeUNYWs5UaiMeUTTKapkDEkxbKfNcYaQaisUdvbo%2Be91cKQvx9QQYbR1xudTQSKjOpR%2F23oqucjZmylQwoPpWe1O%2B9VVvI7GTJggL0cIofDuqHehS%2F6qx8vFlKr06PVz%2BxGBHTUh%2FoI9FikCZjME7sDP369Asg9rjd9fDf2pN%2B7uXf6IT0j48E8Nw%2B6BzQ1vzY8&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20241130T161904Z&X-Amz-SignedHeaders=host&X-Amz-Expires=300&X-Amz-Credential=ASIAQ3PHCVTYUP2DI6MG%2F20241130%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=e044426e631f62a90217ca24ab4ef553ba187676da0a64852489f0825fe1b22f&hash=8c563e051ee2e529d2c05c7a1fa3815a39c19c8df992f23ee46f810bab18d094&host=68042c943591013ac2b2430a89b270f6af2c76d8dfd086a07176afe7c76c2c61&pii=S1877050918314832&tid=spdf-717373fa-b3ee-4499-b3bc-75d6ceda46e3&sid=c225f3f17750964a778ba48472a572029f15gxrqb&type=client&tsoh=d3d3LnNjaWVuY2VkaXJlY3QuY29t&ua=0f045e0107015d5d5053&rr=8eac28300e52409c&cc=id .

Jehangir, B., Radhakrishnan, S. & Agarwal, R., 2023. A survey on named entity recognition—datasets, tools, and methodologies. Expert Systems with Applications, 215, p.119431. Available at: https://www.sciencedirect.com/science/article/pii/S2949719123000146 .

Li, J., Sun, A., Han, J. & Li, C., 2022. A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering, 34(1), pp.50–70. Available at: https://ieeexplore.ieee.org/document/10184827 .

Seti, X. et al., 2020. Named-entity recognition in sports field based on a character-level graph convolutional network. Information, 11(1), p.30. Available at: https://www.mdpi.com/2078-2489/11/1/30 .