

## Laporan Analisis Ekstraksi Informasi

Amati performansi NER dengan setting fitur yang berbeda-beda tersebut, bandingkan hasilnya dengan penggunaan keseluruhan fitur seperti pada setting awal. Berikan analisis terhadap hasil yang diperoleh, **jelaskan apa dugaan penyebabnya jika kinerja lebih baik atau lebih buruk** tanpa penggunaan fitur-fitur tersebut!

### 1. TANPA fitur POSTag *current* kata dan POSTag kata konteks

```
[ ] def word2features(sent, i):
    word = sent[i][0]
    postag = sent[i][1]

    features = {
        'bias': 1.0,
        'word.lower()': word.lower(),
        'word[-3:]': word[-3:],
        'word[-2:]': word[-2:],
        'word.isupper()': word.isupper(),
        'word.istitle()': word.istitle(),
        'word.isdigit()': word.isdigit(),
        # 'postag': postag,
        # 'postag[:2]': postag[:2],
    }
    if i > 0:
        word1 = sent[i-1][0]
        postag1 = sent[i-1][1]
        features.update({
            '-1:word.lower()': word1.lower(),
            '-1:word.istitle()': word1.istitle(),
            '-1:word.isupper()': word1.isupper(),
            # '-1:postag': postag1,
            # '-1:postag[:2]': postag1[:2],
        })
    else:
        features['BOS'] = True
    if i < len(sent)-1:
        word1 = sent[i+1][0]
        postag1 = sent[i+1][1]
        features.update({
            '+1:word.lower()': word1.lower(),
            '+1:word.istitle()': word1.istitle(),
            '+1:word.isupper()': word1.isupper(),
            # '+1:postag': postag1,
            # '+1:postag[:2]': postag1[:2],
        })
    else:
        features['EOS'] = True

    return features

def sent2features(sent):
    return [word2features(sent, i) for i in range(len(sent))]

def sent2labels(sent):
    return [label for token, postag, label in sent]

def sent2tokens(sent):
    return [token for token, postag, label in sent]
```

### Laporan kinerja NER :

	precision	recall	f1-score	support
B-art	0.00	0.00	0.00	29
B-eve	1.00	0.17	0.29	24
B-geo	0.76	0.87	0.81	1043
B-gpe	0.87	0.78	0.82	588
B-nat	0.80	0.40	0.53	10
B-org	0.74	0.61	0.67	649
B-per	0.81	0.77	0.79	546
B-tim	0.91	0.83	0.87	589
I-art	0.00	0.00	0.00	7
I-eve	1.00	0.11	0.20	18
I-geo	0.68	0.71	0.70	204
I-gpe	0.50	0.53	0.51	17
I-nat	1.00	0.50	0.67	2
I-org	0.78	0.73	0.75	545
I-per	0.79	0.89	0.83	574
I-tim	0.79	0.63	0.70	185
micro avg	0.79	0.77	0.78	5030
macro avg	0.71	0.53	0.57	5030
weighted avg	0.79	0.77	0.77	5030

2. TANPA fitur ortografi (terlihat dari fungsi `istitle()`, `isupper()`, dan `isdigit()`) *current* kata dan pada kata konteks

```
[ ] def word2features1(sent, i):
    word = sent[i][0]
    postag = sent[i][1]

    features = {
        'bias': 1.0,
        'word.lower()': word.lower(),
        'word[-3:]': word[-3:],
        'word[-2:]': word[-2:],
        # 'word.isupper()': word.isupper(),
        # 'word.istitle()': word.istitle(),
        # 'word.isdigit()': word.isdigit(),
        'postag': postag,
        'postag[:2]': postag[:2],
    }

    if i > 0:
        word1 = sent[i-1][0]
        postag1 = sent[i-1][1]
        features.update({
            '-1:word.lower()': word1.lower(),
            # '-1:word.istitle()': word1.istitle(),
            # '-1:word.isupper()': word1.isupper(),
            '-1:postag': postag1,
            '-1:postag[:2]': postag1[:2],
        })
    else:
        features['BOS'] = True

    if i < len(sent)-1:
        word1 = sent[i+1][0]
        postag1 = sent[i+1][1]
        features.update({
            '+1:word.lower()': word1.lower(),
            # '+1:word.istitle()': word1.istitle(),
            # '+1:word.isupper()': word1.isupper(),
            '+1:postag': postag1,
            '+1:postag[:2]': postag1[:2],
        })
    else:
        features['EOS'] = True

    return features

def sent2features1(sent):
    return [word2features1(sent, i) for i in range(len(sent))]

def sent2labels1(sent):
    return [label for token, postag, label in sent]

def sent2tokens1(sent):
    return [token for token, postag, label in sent]
```

Laporan kinerja NER :

	precision	recall	f1-score	support
B-art	1.00	0.03	0.07	29
B-eve	1.00	0.25	0.40	24
B-geo	0.75	0.88	0.81	1043
B-gpe	0.88	0.76	0.82	588
B-nat	0.75	0.30	0.43	10
B-org	0.75	0.61	0.67	649
B-per	0.81	0.81	0.81	546
B-tim	0.90	0.84	0.87	589
I-art	0.00	0.00	0.00	7
I-eve	1.00	0.22	0.36	18
I-geo	0.67	0.70	0.68	204
I-gpe	0.45	0.53	0.49	17
I-nat	1.00	0.50	0.67	2
I-org	0.78	0.72	0.74	545
I-per	0.79	0.90	0.84	574
I-tim	0.79	0.68	0.73	185
micro avg	0.79	0.77	0.78	5030
macro avg	0.77	0.55	0.59	5030
weighted avg	0.80	0.77	0.78	5030

### 3. TANPA fitur akhiran pada *current* kata

```
def word2features2(sent, i):
    word = sent[i][0]
    postag = sent[i][1]

    features = {
        'bias': 1.0,
        'word.lower()': word.lower(),
        # 'word[-3:]': word[-3:],
        # 'word[-2:]': word[-2:],
        'word.isupper()': word.isupper(),
        'word.istitle()': word.istitle(),
        'word.isdigit()': word.isdigit(),
        'postag': postag,
        'postag[:2]': postag[:2],
    }
    if i > 0:
        word1 = sent[i-1][0]
        postag1 = sent[i-1][1]
        features.update({
            '-1:word.lower()': word1.lower(),
            '-1:word.istitle()': word1.istitle(),
            '-1:word.isupper()': word1.isupper(),
            '-1:postag': postag1,
            '-1:postag[:2]': postag1[:2],
        })
    else:
        features['BOS'] = True
    if i < len(sent)-1:
        word1 = sent[i+1][0]
        postag1 = sent[i+1][1]
        features.update({
            '+1:word.lower()': word1.lower(),
            '+1:word.istitle()': word1.istitle(),
            '+1:word.isupper()': word1.isupper(),
            '+1:postag': postag1,
            '+1:postag[:2]': postag1[:2],
        })
    else:
        features['EOS'] = True

    return features

def sent2features2(sent):
    return [word2features2(sent, i) for i in range(len(sent))]

def sent2labels2(sent):
    return [label for token, postag, label in sent]

def sent2tokens2(sent):
    return [token for token, postag, label in sent]
```

### Laporan kinerja NER:

	precision	recall	f1-score	support
B-art	1.00	0.03	0.07	29
B-eve	1.00	0.25	0.40	24
B-geo	0.74	0.88	0.81	1043
B-gpe	0.88	0.76	0.81	588
B-nat	0.67	0.20	0.31	10
B-org	0.74	0.62	0.68	649
B-per	0.82	0.80	0.81	546
B-tim	0.92	0.83	0.87	589
I-art	0.00	0.00	0.00	7
I-eve	1.00	0.22	0.36	18
I-geo	0.69	0.71	0.70	204
I-gpe	0.45	0.53	0.49	17
I-nat	1.00	0.50	0.67	2
I-org	0.78	0.73	0.76	545
I-per	0.80	0.91	0.85	574
I-tim	0.81	0.68	0.74	185
micro avg	0.79	0.78	0.78	5030
macro avg	0.77	0.54	0.58	5030
weighted avg	0.80	0.78	0.78	5030

#### 4. Dengan Semua Fitur

```
def word2features3(sent, i):
    word = sent[i][0]
    postag = sent[i][1]

    features = {
        'bias': 1.0,
        'word.lower()': word.lower(),
        'word[-3:]': word[-3:],
        'word[-2:]': word[-2:],
        'word.isupper()': word.isupper(),
        'word.istitle()': word.istitle(),
        'word.isdigit()': word.isdigit(),
        'postag': postag,
        'postag[:2]': postag[:2],
    }
    if i > 0:
        word1 = sent[i-1][0]
        postag1 = sent[i-1][1]
        features.update({
            '-1:word.lower()': word1.lower(),
            '-1:word.istitle()': word1.istitle(),
            '-1:word.isupper()': word1.isupper(),
            '-1:postag': postag1,
            '-1:postag[:2]': postag1[:2],
        })
    else:
        features['BOS'] = True
    if i < len(sent)-1:
        word1 = sent[i+1][0]
        postag1 = sent[i+1][1]
        features.update({
            '+1:word.lower()': word1.lower(),
            '+1:word.istitle()': word1.istitle(),
            '+1:word.isupper()': word1.isupper(),
            '+1:postag': postag1,
            '+1:postag[:2]': postag1[:2],
        })
    else:
        features['EOS'] = True

    return features

def sent2features3(sent):
    return [word2features3(sent, i) for i in range(len(sent))]

def sent2labels3(sent):
    return [label for token, postag, label in sent]

def sent2tokens3(sent):
    return [token for token, postag, label in sent]
```

#### Laporan kinerja NER:

	precision	recall	f1-score	support
B-art	1.00	0.03	0.07	29
B-eve	0.86	0.25	0.39	24
B-geo	0.75	0.88	0.81	1043
B-gpe	0.89	0.78	0.83	588
B-nat	0.67	0.20	0.31	10
B-org	0.75	0.64	0.69	649
B-per	0.81	0.81	0.81	546
B-tim	0.90	0.85	0.87	589
I-art	0.00	0.00	0.00	7
I-eve	0.57	0.22	0.32	18
I-geo	0.71	0.71	0.71	204
I-gpe	0.47	0.53	0.50	17
I-nat	1.00	0.50	0.67	2
I-org	0.78	0.73	0.76	545
I-per	0.80	0.90	0.85	574
I-tim	0.79	0.68	0.73	185
micro avg	0.80	0.78	0.79	5030
macro avg	0.73	0.54	0.58	5030
weighted avg	0.80	0.78	0.78	5030

#### Analisis hasil eksperimen

##### 1. Percobaan pertama tanpa menggunakan fitur POSTag current kata dan POSTag kata konteks

Dengan menggunakan metrik evaluasi untuk mendapatkan informasi hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi sebenarnya, untuk precision berhasil mendapatkan micro avg 79%, macro avg 71% dan weighted avg 79% yang menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan, untuk recall berhasil mendapatkan micro avg 77%, macro avg 53% dan weighted avg 77% yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi, untuk f1-score berhasil mendapatkan micro avg 78%, macro avg 57% dan weighted avg 77% yang mengindikasikan bahwa model klasifikasi kita punya precision dan recall yang baik

2. Percobaan kedua tanpa fitur ortografi (terlihat dari fungsi `istitle()`, `isupper()`, dan `isdigit()`) *current* kata dan pada konteks

Dengan menggunakan metrik evaluasi untuk mendapatkan informasi hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi sebenarnya, untuk precision berhasil mendapatkan micro avg 79%, macro avg 77% dan weighted avg 80% yang menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan, untuk recall berhasil mendapatkan micro avg 77%, macro avg 55% dan weighted avg 77% yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi, untuk f1-score berhasil mendapatkan micro avg 78%, macro avg 59% dan weighted avg 78% yang mengindikasikan bahwa model klasifikasi kita punya precision dan recall yang baik

3. Percobaan ketiga tanpa fitur akhiran pada *current* kata

Dengan menggunakan metrik evaluasi untuk mendapatkan informasi hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi sebenarnya, untuk precision berhasil mendapatkan micro avg 79%, macro avg 77% dan weighted avg 80% yang menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan, untuk recall berhasil mendapatkan micro avg 78%, macro avg 54% dan weighted avg 78% yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi, untuk f1-score berhasil mendapatkan micro avg 78%, macro avg 58% dan weighted avg 78% yang mengindikasikan bahwa model klasifikasi kita punya precision dan recall yang baik

4. Percobaan keempat dengan semua fitur

Dengan menggunakan metrik evaluasi untuk mendapatkan informasi hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi sebenarnya, untuk precision berhasil mendapatkan micro avg 80%, macro avg 73% dan weighted avg 80% yang menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan, untuk recall berhasil mendapatkan micro avg 78%, macro avg 54% dan weighted avg 78% yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi, untuk f1-score berhasil mendapatkan micro avg 79%, macro avg 58% dan weighted avg 78% yang mengindikasikan bahwa model klasifikasi kita punya precision dan recall yang baik

Berdasarkan keempat percobaan yang telah dilakukan percobaan dengan semua fitur memiliki precision, recall, f1-score yang lebih baik. Karena jika tanpa menggunakan fitur POSTag tidak dapat menemukan tag yang tepat untuk konteks kata di dalam kalimat, tanpa menggunakan fitur dengan fungsi Python `istitle()` untuk mendeteksi semua kata dieja dengan string adalah pertama huruf besar dan huruf lainnya untuk huruf kecil, `isupper()` untuk operasi karakter yang akan bernilai benar(bukan nol) jika karakter merupakan huruf kapital dan `isdigit()` yang berfungsi fungsi operasi karakter yang akan bernilai benar(bukan nol) jika karakter merupakan sebuah angka (digit) dari 0 sampai dengan 9 dan tanpa fitur *current* tidak bisa mengetahui kata mana yang akan digunakan yang sesuai. Oleh karena itu fitur-fitur tersebut membantu untuk dapat mengekstrak informasi terstruktur dari teks yang tidak terstruktur dalam kasus ini mengidentifikasi Named Entity