PENDEKATAN CONDITIONAL RANDOM FIELD PADA INDONESIAN NAMED ENTITY RECOGNITION

AESTHETICS:

11S18022 - DEVI WAHYUNI SILITONGA
11S18037 - HISAR HARYANTO SINAGA
11S18046 - SIMSON FRANSISCO PANJAITAN
11S18058 - LEONARDO ROBINSAR AGUSTINUS SIANTURI
11S18066 - JUMADI HERYANTO DAMANIK

NAMED ENTITY RECOGNITION

Tujuan: Untuk mengidentifikasi atau mengklasifikasi sebuah entitas misalnya nama orang, organisasi, lokasi, waktu dan entitas lain dalam sebuat teks yang sangat berguna dalam kasus ektraksi informasi.

PENDEKATAN CONDITIONAL RANDOM FIELD PADA INDONESIAN NAMED ENTITY RECOGNITION

Penelitian kami bertujuan untuk membuat model *Indonesian Named-Entity Recognition* yang dapat digunakan untuk melakukan tag entitas (nama orang, nama lokasi, dan nama organisasi) dalam bahasa Indonesia.

BATASAN

1. Dataset: Singgalang
2. Entitas yang di klasifikasi adalah nama orang, nama lokasi, dan nama organisasi

TAHAPAN

Pemrosesan

Model Klasifikasi

Evaluasi Model

PEMROSESAN

1. Lowercase

Mengubah seluruh data menjadi huruf kecil

```
# Lowercase
dataset_prep["Word"] = dataset_prep["Word"].str.lower()
dataset_prep.head()
```

	Word	Tag	kalimat
0	ia	O	1
1	menjabat	O	1
2	sebagai	O	1
3	presiden	O	1
4	ketiga	O	1

PEMROSESAN

2. BIO Annotation

Memberi tag BIO

- B – {CHUNK_TYPE} – Chunk di bagian awal (Beginning)
- I – {CHUNK_TYPE} – Chunk di bagian dalam (Inside)
- O – Chunk di bagian luar (Outside)

```
# BIO Annotation

bio_tag = []
prev_tag = "O"
for tag in dataset_prep['Tag']:
        if tag == "O": #O
            bio_tag.append((tag))
            prev_tag = tag
            continue
        if tag != "O" and prev_tag == "O": # Begin NE
            bio_tag.append(("B-"+tag))
            prev_tag = tag
        elif prev_tag != "O" and prev_tag == tag: # Inside NE
            bio_tag.append(("I-"+tag))
            prev_tag = tag
        elif prev_tag != "O" and prev_tag != tag: # NE yang berdekatan
            bio_tag.append(( "B-"+tag))
            prev_tag = tag
```

```
dataset_prep['bio_tag'] = bio_tag
dataset_prep.iloc[:2000]
```

	Word	Tag	kalimat	bio_tag
0	ia	O	1	O
1	menjabat	O	1	O
2	sebagai	O	1	O
3	presiden	O	1	O
4	ketiga	O	1	O
...	...	...	...	...
1995	kerajaan	Place	1	B-Place
1996	majapahit	Place	1	I-Place
1997	di	O	1	O
1998	bawah	O	1	O
1999	raja	O	1	O

2000 rows × 4 columns

MODEL KLASIFIKASI

Random Forest Classifier

Random Forest Classifier merupakan algoritma yang digunakan pada klasifikasi data dalam jumlah yang besar. Klasifikasi random forest dilakukan melalui penggabungan pohon (tree) dengan melakukan training pada sampel data yang dimiliki.

```
# Using Random Forest Classifier
def feature_map(word):
    return np.array([word.istitle(), word.islower(), word.isupper(), len(word),
                     word.isdigit(),  word.isalpha()])
```

```
words = [feature_map(w) for w in dataset["Word"].values.tolist()]
tags = dataset["Tag"].values.tolist()
```

```
print(words[:5])
```

```
[array([1, 0, 0, 2, 0, 1]), array([0, 1, 0, 8, 0, 1]), array([0, 1, 0, 7, 0,
1]), array([1, 0, 0, 8, 0, 1]), array([0, 1, 0, 6, 0, 1])]
```

EVALUASI MODEL

Hasil evaluasi menggunakan Random Forest Classifier diperoleh akurasi sebesar 91%

```
# Evaluate Model from Random Forest

from sklearn.metrics import classification_report
report = classification_report(y_pred=pred, y_true=tags)
print(report)
```

	precision	recall	f1-score	support
	1.00	1.00	1.00	33453
O	0.92	0.99	0.95	921226
Organisation	0.00	0.00	0.00	9211
Person	0.00	0.00	0.00	28028
Place	0.47	0.19	0.27	59994
accuracy			0.91	1051912
macro avg	0.48	0.44	0.44	1051912
weighted avg	0.86	0.91	0.88	1051912

MODEL KLASIFIKASI

Conditional Random Fields

Conditional Random Fields kelas metode pemodelan statistik yang sering diterapkan dalam pengenalan pola dan pembelajaran mesin dan digunakan untuk prediksi terstruktur .

```
#Creating the train and test set
X = [sent2features(s) for s in kalimat]
y = [sent2labels(s) for s in kalimat]
```

```
#Creating the CRF model
crf = CRF(algorithm='lbfgs',
          c1=0.1,
          c2=0.1,
          max_iterations=100,
          all_possible_transitions=False)
```

```
#We predcit using the same 5 fold cross validation
pred = cross_val_predict(estimator=crf, X=X, y=y, cv=5)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:209: FutureWarning: From version 0.24, get_params will raise an Attr
ibuteError if a parameter cannot be retrieved as an instance attribute. Previously it would return None.
  warnings.warn('From version 0.24, get_params will raise an '
```

EVALUASI MODEL

Hasil evaluasi menggunakan Conditional Random Fields diperoleh akurasi sebesar 100%

```
#Lets evaluate the model
report = flat_classification_report(y_pred=pred, y_true=y)
print(report)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:68: FutureWarning: Pass labels=None as keyword args. Fro
m version 0.25 passing these as positional arguments will result in an error
  warnings.warn("Pass {} as keyword args. From version 0.25 "
```

	precision	recall	f1-score	support
	1.00	1.00	1.00	33453
O	1.00	1.00	1.00	921226
Organisation	1.00	1.00	1.00	9211
Person	1.00	1.00	1.00	28028
Place	1.00	1.00	1.00	59994
accuracy			1.00	1051912
macro avg	1.00	1.00	1.00	1051912
weighted avg	1.00	1.00	1.00	1051912

TERIMA KASIH