Tugas 1 IF4072 Natural Language Processing

Deskripsi :

Buatlah kode program penggunaan Spacy untuk

- Sentence splitter
- Tokenization
- Stemming
- Lemmatization
- Entity Masking
- POS Tagger
- Phrase Chunking

Lengkapi kode program tersebut dengan definisi dari setiap NLP tools dalam file python-nya

Dibuat dalam file .ipynb atau .py dimana setiap kode program perlu diberi komentar berupa penjelasan kode program tersebut, nama file adalah nim mahasiswa

Library

```
pip install -U spacy-cleaner
```

```
Requirement already satisfied: spacy-cleaner in /usr/local/lib/python3.10/dist-packages (3.1.3)
Requirement already satisfied: spacy<3.5.0,>=3.4.1 in /usr/local/lib/python3.10/dist-packages (from spacy-cleaner) (3.4.4)
Requirement already satisfied: spacy-lookups-data<1.1.0,>=1.0.3 in /usr/local/lib/python3.10/dist-packages (from spacy-cleaner) (1.0.5)
Requirement already satisfied: tqdm<4.65.0,>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from spacy-cleaner) (4.64.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.10 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (1.0.4)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (1.0.9)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (2.0.7)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (3.0.8)
Requirement already satisfied: thinc<8.2.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (8.1.12)
Requirement already satisfied: wasabi<1.1.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (0.10.1)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (2.4.7)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (2.0.9)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (0.7.0)
Requirement already satisfied: pathy>=0.3.5 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (0.10.2)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (6.3.0)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (1.23.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (1.10.12)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (3.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (67.7.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (23.1)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.5.0,>=3.4.1->spacy-cleaner) (3.3.0)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4->spacy<3.5.0,>=3.4.1->spacy-clear
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.5.0,>=3.4.1->spacy-cleaner) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.5.0,>=3.4.1->spacy-cleaner) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.5.0,>=3.4.1->spacy-cleaner) (2.0.4)
```

```
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.5.0,>=3.4.1->spacy-cleaner) (2023.7.22)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.0->spacy<3.5.0,>=3.4.1->spacy-cleaner) (0.7.10)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.0->spacy<3.5.0,>=3.4.1->spacy-cleaner) (0.1.1)
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.10/dist-packages (from typer<0.8.0,>=0.3.0->spacy<3.5.0,>=3.4.1->spacy-cleaner) (8.1.7)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->spacy<3.5.0,>=3.4.1->spacy-cleaner) (2.1.3)
```

```
import spacy
import spacy_cleaner
from spacy_cleaner.processing import removers, replacers, mutators
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
True
```

Text

```
paragraph = 'Natural language processing (NLP) is a branch of artificial intelligence (AI) that enables computers to comprehend, generate, and manipulate human language. Natural lan
sentence = 'ChatGPT, which stands for Chat Generative Pre-trained Transformer, is a large language model-based chatbot developed by OpenAI and launched on November 30, 2022, notable

print(f'Text Paragraf : \n{paragraph}\n')
print(f'Text Kalimat : \n{sentence}\n')
```

```
Text Paragraf :
Natural language processing (NLP) is a branch of artificial intelligence (AI) that enables computers to comprehend, generate, and manipulate human language. Natural language pr

Text Kalimat :
ChatGPT, which stands for Chat Generative Pre-trained Transformer, is a large language model-based chatbot developed by OpenAI and launched on November 30, 2022, notable for er
```

Sentence Splitter

```
def sentence_splitter(text) :
  nlp = spacy.load("en_core_web_sm")
  doc = nlp(text)

  # Menyimpan split sentence
  res = []
  for sent in doc.sents :
    res.append(sent)
  return res

result = sentence_splitter(paragraph)
i = 1
for sent in result:
  print(f'Kalimat ke-{i} : {sent}')
  i += 1
```

```
Kalimat ke-1 : Natural language processing (NLP) is a branch of artificial intelligence (AI) that enables computers to comprehend, generate, and manipulate human language.
Kalimat ke-2 : Natural language processing has the ability to interrogate the data with natural language text or voice.
Kalimat ke-3 : This is also called “language in.”
Kalimat ke-4 : Most consumers have probably interacted with NLP without realizing it.
Kalimat ke-5 : For instance, NLP is the core technology behind virtual assistants, such as the Oracle Digital Assistant (ODA), Siri, Cortana, or Alexa.
Kalimat ke-6 : When we ask questions of these virtual assistants, NLP is what enables them to not only understand the user’s request, but to also respond in natural language.
Kalimat ke-7 : NLP applies both to written text and speech, and can be applied to all human languages.
Kalimat ke-8 : Other examples of tools powered by NLP include web search, email spam filtering, automatic translation of text or speech, document summarization, sentiment analy
Kalimat ke-9 : For example, some email programs can automatically suggest an appropriate reply to a message based on its content—these programs use NLP to read, analyze, and re
```

Tokenization

```
def tokenize(text) :
  nlp = spacy.load("en_core_web_sm")
  doc = nlp(text)

  # Menyimpan tokenize word
  res = []
  for token in doc:
    res.append(token)
  return res

result = tokenize(sentence)
for word in result :
  print(word.text)
```

```
ChatGPT
,
which
stands
for
Chat
Generative
Pre
-
trained
Transformer
,
is
a
large
language
model
-
based
chatbot
developed
by
OpenAI
and
launched
on
November
30
,
```

```
2022
,
notable
for
enabling
users
to
refine
and
steer
a
conversation
towards
a
desired
length
,
format
,
style
,
level
of
detail
,
and
language
used
.
```

Remove punctuation

```
def cleaner(text) :
  model = spacy.load("en_core_web_sm")
  tes = spacy_cleaner.Cleaner(model, True, True)
  return tes.clean(text)
```

Stemming

spaCy tidak menyediakan function untuk melakukan stemming

```
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize

def stemming(text) :
  # remove puncuation dan stop-words
  clean = cleaner([text])[0]
  words = tokenize(clean)

  # melakukan stemming dengan nltk
  print('Hasil Stemming :\n')
  ps = PorterStemmer()
  for w in words:
```

```
    print(w.text, " --> ", ps.stem(w.text))

stemming(sentence)
```

```
     Cleaning Progress: 100%|██████████| 1/1 [00:00<00:00, 46.71it/s]
     Hasil Stemming :

     chatgpt  -->  chatgpt
     stands  -->  stand
     chat  -->  chat
     generative  -->  gener
     pre  -->  pre
     trained  -->  train
     transformer  -->  transform
     large  -->  larg
     language  -->  languag
     model  -->  model
     based  -->  base
     chatbot  -->  chatbot
     developed  -->  develop
     openai  -->  openai
     launched  -->  launch
     november  -->  novemb
     notable  -->  notabl
     enabling  -->  enabl
     users  -->  user
     refine  -->  refin
     steer  -->  steer
     conversation  -->  convers
     desired  -->  desir
     length  -->  length
     format  -->  format
     style  -->  style
     level  -->  level
     detail  -->  detail
     language  -->  languag
```

Lemmatization

```
def lemmatization(text) :
  # remove puncuation dan stop-words
  clean = cleaner([text])[0]

  # melakukan lemmatization dengan spaCy
  print('Hasil Lemmatization :\n')
  nlp = spacy.load('en_core_web_sm')
  doc = nlp(clean)
  for token in doc:
      print(token.text, "-->", token.lemma_)

lemmatization(sentence)
```

```
     Cleaning Progress: 100%|██████████| 1/1 [00:00<00:00, 47.27it/s]
     Hasil Lemmatization :

     chatgpt --> chatgpt
```

```
stands --> stand
chat --> chat
generative --> generative
pre --> pre
trained --> train
transformer --> transformer
large --> large
language --> language
model --> model
based --> base
chatbot --> chatbot
developed --> develop
openai --> openai
launched --> launch
november --> november
notable --> notable
enabling --> enable
users --> user
refine --> refine
steer --> steer
conversation --> conversation
desired --> desire
length --> length
format --> format
style --> style
level --> level
detail --> detail
language --> language
```

Entity Masking

```
def entity_masking(text) :
  # remove puncuation dan stop-words
  clean = cleaner([text])[0]

  # entity masking (named entity recognition)
  nlp = spacy.load('en_core_web_sm')
  doc = nlp(clean)
  for w in doc:
    print((w, w.ent_iob_, w.ent_type_))

entity_masking(sentence)
```

```
Cleaning Progress: 100%|██████████| 1/1 [00:00<00:00, 44.90it/s]
(chatgpt, 'O', '')
(stands, 'O', '')
(chat, 'O', '')
(generative, 'O', '')
(pre, 'O', '')
(trained, 'O', '')
(transformer, 'O', '')
(large, 'O', '')
(language, 'O', '')
(model, 'O', '')
(based, 'O', '')
(chatbot, 'O', '')
(developed, 'O', '')
```

```
(openai, 'O', '')
(launched, 'O', '')
(november, 'B', 'DATE')
(notable, 'O', '')
(enabling, 'O', '')
(users, 'O', '')
(refine, 'O', '')
(steer, 'O', '')
(conversation, 'O', '')
(desired, 'O', '')
(length, 'O', '')
(format, 'O', '')
(style, 'O', '')
(level, 'O', '')
(detail, 'O', '')
(language, 'O', '')
```

POS Tagger

```
def pos_tagger(text) :
  # POS Tagger
  nlp = spacy.load("en_core_web_sm")
  doc = nlp(text)
  for word in doc:
    print(word, ':', word.pos_) # Mengambil word.pos_ untuk melihat POS tagger dari kata/token tsb

pos_tagger(sentence)
```

```
ChatGPT : PROPN
, : PUNCT
which : PRON
stands : VERB
for : ADP
Chat : PROPN
Generative : PROPN
Pre : PROPN
- : PUNCT
trained : VERB
Transformer : PROPN
, : PUNCT
is : AUX
a : DET
large : ADJ
language : NOUN
model : NOUN
- : PUNCT
based : VERB
chatbot : NOUN
developed : VERB
by : ADP
OpenAI : PROPN
and : CCONJ
launched : VERB
on : ADP
November : PROPN
30 : NUM
, : PUNCT
```

```
2022 : NUM
, : PUNCT
notable : ADJ
for : ADP
enabling : VERB
users : NOUN
to : PART
refine : VERB
and : CCONJ
steer : VERB
a : DET
conversation : NOUN
towards : ADP
a : DET
desired : VERB
length : NOUN
, : PUNCT
format : NOUN
, : PUNCT
style : NOUN
, : PUNCT
level : NOUN
of : ADP
detail : NOUN
, : PUNCT
and : CCONJ
language : NOUN
used : VERB
. : PUNCT
```

Phrase Chunking

```
def phrase_chunking(text) :
  # Phrase Noun Chunking
  nlp = spacy.load("en_core_web_sm")
  doc = nlp(text)
  for chunk in doc.noun_chunks:
    print(chunk.text)

phrase_chunking(sentence)
```

```
ChatGPT
which
Chat Generative Pre-trained Transformer
a large language model-based chatbot
OpenAI
November
users
a conversation
a desired length, format, style, level
detail
language
```

Kesimpulan

Sebenarnya dengan menggunakan library dari spaCy ini yang sangat simple dengan hanya memanggil `spacy.load("en_core_web_sm")(text)`, spaCy sudah menyediakan berbagai dasar operasi NLP. Selanjutnya kita hanya perlu memanggilnya saja sesuai yang kita perlukan.