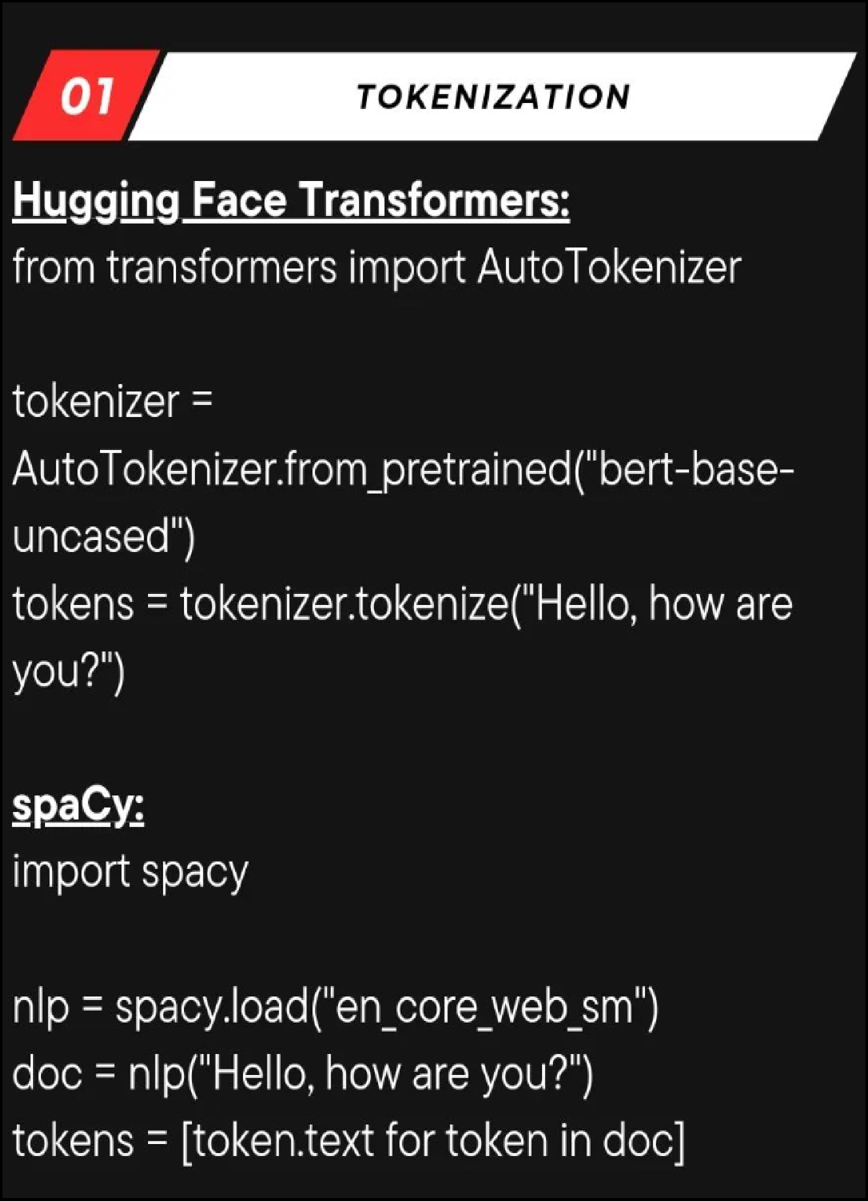
ALGORITHMS

CHEAT SHEET

for Natural Language Processing



01 TOKENIZATION

Hugging Face Transformers:

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
tokens = tokenizer.tokenize("Hello, how are you?")
```

spaCy:

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Hello, how are you?")
tokens = [token.text for token in doc]
```

02 NAMED ENTITY RECOGNITION

Hugging Face Transformers:

```
from transformers import pipeline

ner_pipeline = pipeline("ner", model="dbmdz/bert-large-cased-finetuned-conll03-english")
entities = ner_pipeline("Hugging Face is a great NLP library.")
```

spaCy:

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Hugging Face is a great NLP library.")
entities = [(ent.text, ent.label_) for ent in doc.ents]
```

03 TEXT CLASSIFICATION

Hugging Face Transformers:

```
from transformers import pipeline

classifier = pipeline("sentiment-analysis")
result = classifier("I love using Hugging Face Transformers!")
```

scikit-learn:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline

model = make_pipeline(CountVectorizer(), MultinomialNB())
model.fit(X_train, y_train)
result = model.predict(["I love using scikit-learn!"])
```

04 PART-OF-SPEECH TAGGING

Hugging Face Transformers:

(Note: Hugging Face doesn't have a specific pre-trained model for POS tagging as of my knowledge cutoff in January 2022)

spaCy:

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Hugging Face is a great NLP library.")
pos_tags = [(token.text, token.pos_) for token in doc]
```

05 WORD EMBEDDINGS

Hugging Face Transformers:

```
from transformers import AutoModel, AutoTokenizer
import torch

model_name = "bert-base-uncased"
model = AutoModel.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name)

text = "Hugging Face is awesome!"
inputs = tokenizer(text, return_tensors="pt")
embeddings = model(**inputs).last_hidden_state
```

Word2Vec with NLTK:

```
from nltk.tokenize import word_tokenize
from gensim.models import Word2Vec

text = "Hugging Face is awesome!"
tokens = word_tokenize(text)
model = Word2Vec([tokens], vector_size=100, window=5,
min_count=1, workers=4)
embeddings = model.wv["Hugging"]
```