Metin Oluşturu Model İnşa Etme

Halihazırda öğrenmiş olduğunuz tekniklerin çoğunu kullanarak, belirli bir ana kelimeyi takip eden bir sonraki kelimeyi tahmin ederek yeni metin oluşturmak artık mümkün. Bu yöntemi uygulamak için Kaggle Şarkı Sözleri Veri Kümesini kullanacağız.

TensorFlow ve Gereli Fonksiyonların İçeri Aktarılması ¶

In [1]:

```
import tensorflow as tf

from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Verilerin işlenmesi için diğer içeri aktarmalar
import string
import numpy as np
import pandas as pd
```

Veri Setinin Alınması

Yukarıda belirttiğimiz gibi bu colab dosyasında model geliştirmek için şarkı sözleri veri setini kullanacağız.

In [2]:

```
!wget --no-check-certificate \
    https://drive.google.com/uc?id=1LiJFZd41ofrWoBtW-pMYsfz1w8Ny0Bj8 \
    -O /tmp/songdata.csv
```

```
--2021-07-28 20:25:55--  https://drive.google.com/uc?id=1LiJFZd41ofrWoBtW-pM
Ysfz1w8Ny0Bj8 (https://drive.google.com/uc?id=1LiJFZd41ofrWoBtW-pMYsfz1w8Ny0
Bj8)
Resolving drive.google.com (drive.google.com)... 172.217.9.206, 2607:f8b0:40
04:806::200e
Connecting to drive.google.com (drive.google.com)|172.217.9.206|:443... conn
ected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://doc-04-ak-docs.googleusercontent.com/docs/securesc/ha0ro93
7gcuc7l7deffksulhg5h7mbp1/f58oiassacf7bh3bmkqa7c7dtaf96pt3/1627503900000/111
18900490791463723/*/1LiJFZd41ofrWoBtW-pMYsfz1w8Ny0Bj8 (https://doc-04-ak-doc
s.googleusercontent.com/docs/securesc/ha0ro937gcuc7l7deffksulhg5h7mbp1/f58oi
assacf7bh3bmkqa7c7dtaf96pt3/1627503900000/11118900490791463723/*/1LiJFZd41of
rWoBtW-pMYsfz1w8Ny0Bj8) [following]
Warning: wildcards not supported in HTTP.
--2021-07-28 20:25:57--  https://doc-04-ak-docs.googleusercontent.com/docs/s
ecuresc/ha0ro937gcuc7l7deffksulhg5h7mbp1/f58oiassacf7bh3bmkqa7c7dtaf96pt3/16
27503900000/11118900490791463723/*/1LiJFZd41ofrWoBtW-pMYsfz1w8Ny0Bj8 (http
s://doc-04-ak-docs.googleusercontent.com/docs/securesc/ha0ro937gcuc7l7deffks
ulhg5h7mbp1/f58oiassacf7bh3bmkqa7c7dtaf96pt3/1627503900000/11118900490791463
723/*/1LiJFZd41ofrWoBtW-pMYsfz1w8Ny0Bj8)
Resolving doc-04-ak-docs.googleusercontent.com (doc-04-ak-docs.googleusercon
tent.com)... 172.217.164.129, 2607:f8b0:4004:814::2001
Connecting to doc-04-ak-docs.googleusercontent.com (doc-04-ak-docs.googleuse
rcontent.com)|172.217.164.129|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: ‘/tmp/songdata.csv’

/tmp/songdata.csv       [    <=>             ]  69.08M   122MB/s    in 0.6s

2021-07-28 20:25:58 (122 MB/s) - ‘/tmp/songdata.csv’ saved [72436445]
```

Önce veri setinden sadece 10 şarkıya bakalım ve işlerin nasıl performans gösterdiğini görelim.

Ön İşleme Yapılması

Noktalama işaretlerinden kurtulmak ve her şeyi küçük harf yapmak için bazı temel önişlemeler yapalım. Daha sonra sözleri satırlara bölüp şarkı sözlerini belirteceğiz.

In [3]:

```
def tokenize_corpus(corpus, num_words=-1):
  # Tokenize ediciyi yerleştirin
  if num_words > -1:
    tokenizer = Tokenizer(num_words=num_words)
  else:
    tokenizer = Tokenizer()
  tokenizer.fit_on_texts(corpus)
  return tokenizer

def create_lyrics_corpus(dataset, field):
  # Diğer tüm noktalama işaretlerini kaldır
  dataset[field] = dataset[field].str.replace('[{}]'.format(string.punctuation), '')
  # küçük harf yap
  dataset[field] = dataset[field].str.lower()
  # Satıra göre bölmek için uzun bir dize yapın
  lyrics = dataset[field].str.cat()
  corpus = lyrics.split('\n')
  # Sondaki boşlukları kaldırın
  for l in range(len(corpus)):
    corpus[l] = corpus[l].rstrip()
  # Boş satırları kaldırın
  corpus = [l for l in corpus if l != '']

  return corpus
```

In [4]:

```
# Veri kümesini csv'den okuyun - şimdilik sadece ilk 10 şarkı
dataset = pd.read_csv('/tmp/songdata.csv', dtype=str)[:10]
# Şarkı sözlerini içeren 'metin' sütununu kullanarak derlemi oluşturun
corpus = create_lyrics_corpus(dataset, 'text')
# Tokenize edin
tokenizer = tokenize_corpus(corpus)

total_words = len(tokenizer.word_index) + 1

print(tokenizer.word_index)
print(total_words)
```

```
{'you': 1, 'i': 2, 'and': 3, 'a': 4, 'me': 5, 'the': 6, 'is': 7, 'my': 8, 't
o': 9, 'ma': 10, 'it': 11, 'of': 12, 'im': 13, 'your': 14, 'love': 15, 'so':
16, 'as': 17, 'that': 18, 'in': 19, 'andante': 20, 'boomaboomerang': 21, 'ma
ke': 22, 'on': 23, 'oh': 24, 'for': 25, 'but': 26, 'new': 27, 'bang': 28, 'i
ts': 29, 'be': 30, 'like': 31, 'know': 32, 'now': 33, 'how': 34, 'could': 3
5, 'youre': 36, 'sing': 37, 'never': 38, 'no': 39, 'chiquitita': 40, 'can':
41, 'we': 42, 'song': 43, 'had': 44, 'good': 45, 'youll': 46, 'she': 47, 'ju
st': 48, 'girl': 49, 'again': 50, 'will': 51, 'take': 52, 'please': 53, 'le
t': 54, 'am': 55, 'eyes': 56, 'was': 57, 'always': 58, 'cassandra': 59, 'blu
e': 60, 'time': 61, 'dont': 62, 'were': 63, 'return': 64, 'once': 65, 'the
n': 66, 'sorry': 67, 'cryin': 68, 'over': 69, 'feel': 70, 'ever': 71, 'belie
ve': 72, 'what': 73, 'do': 74, 'go': 75, 'all': 76, 'out': 77, 'think': 78,
'every': 79, 'leave': 80, 'look': 81, 'at': 82, 'way': 83, 'one': 84, 'musi
c': 85, 'down': 86, 'our': 87, 'give': 88, 'learn': 89, 'more': 90, 'us': 9
1, 'would': 92, 'there': 93, 'before': 94, 'when': 95, 'with': 96, 'feelin
g': 97, 'play': 98, 'cause': 99, 'away': 100, 'here': 101, 'have': 102, 'ye
s': 103, 'baby': 104, 'get': 105, 'didnt': 106, 'see': 107, 'did': 108, 'clo
sed': 109, 'realized': 110, 'crazy': 111, 'world': 112, 'lord': 113, 'shes':
114, 'kind': 115, 'without': 116, 'if': 117, 'touch': 118, 'strong': 119, 'm
aking': 120, 'such': 121, 'found': 122, 'true': 123, 'stay': 124, 'togethe
r': 125, 'thought': 126, 'come': 127, 'they': 128, 'sweet': 129, 'tender': 1
30, 'sender': 131, 'tune': 132, 'humdehumhum': 133, 'gonna': 134, 'last': 13
5, 'leaving': 136, 'sleep': 137, 'only': 138, 'saw': 139, 'tell': 140, 'he
s': 141, 'her': 142, 'sound': 143, 'tread': 144, 'lightly': 145, 'ground': 1
46, 'ill': 147, 'show': 148, 'life': 149, 'too': 150, 'used': 151, 'darlin
g': 152, 'meant': 153, 'break': 154, 'end': 155, 'yourself': 156, 'little':
157, 'dumbedumdum': 158, 'bedumbedumdum': 159, 'youve': 160, 'dumbbedumbdum
b': 161, 'bedumbbedumbdumb': 162, 'by': 163, 'theyre': 164, 'alone': 165, 'm
isunderstood': 166, 'day': 167, 'dawning': 168, 'some': 169, 'wanted': 170,
'none': 171, 'listen': 172, 'words': 173, 'warning': 174, 'darkest': 175, 'n
ights': 176, 'nobody': 177, 'knew': 178, 'fight': 179, 'caught': 180, 'reall
y': 181, 'power': 182, 'dreams': 183, 'weave': 184, 'until': 185, 'final': 1
86, 'hour': 187, 'morning': 188, 'ship': 189, 'gone': 190, 'grieving': 191,
'still': 192, 'pain': 193, 'cry': 194, 'sun': 195, 'try': 196, 'face': 197,
'something': 198, 'sees': 199, 'makes': 200, 'fine': 201, 'who': 202, 'min
e': 203, 'leaves': 204, 'walk': 205, 'hand': 206, 'well': 207, 'about': 208,
'things': 209, 'slow': 210, 'theres': 211, 'talk': 212, 'why': 213, 'up': 21
4, 'lousy': 215, 'packing': 216, 'ive': 217, 'gotta': 218, 'near': 219, 'kee
ping': 220, 'intention': 221, 'growing': 222, 'taking': 223, 'dimension': 22
4, 'even': 225, 'better': 226, 'thank': 227, 'god': 228, 'not': 229, 'somebo
dy': 230, 'happy': 231, 'question': 232, 'smile': 233, 'mean': 234, 'much':
235, 'kisses': 236, 'around': 237, 'anywhere': 238, 'advice': 239, 'care': 2
40, 'use': 241, 'selfish': 242, 'tool': 243, 'fool': 244, 'showing': 245, 'b
oomerang': 246, 'throwing': 247, 'warm': 248, 'kiss': 249, 'surrender': 250,
'giving': 251, 'been': 252, 'door': 253, 'burning': 254, 'bridges': 255, 'be
ing': 256, 'moving': 257, 'though': 258, 'behind': 259, 'are': 260, 'must':
261, 'sure': 262, 'stood': 263, 'hope': 264, 'this': 265, 'deny': 266, 'sa
```

```
d': 267, 'quiet': 268, 'truth': 269, 'heartaches': 270, 'scars': 271, 'danci
ng': 272, 'sky': 273, 'shining': 274, 'above': 275, 'hear': 276, 'came': 27
7, 'couldnt': 278, 'everything': 279, 'back': 280, 'long': 281, 'waitin': 28
2, 'cold': 283, 'chills': 284, 'bone': 285, 'youd': 286, 'wonderful': 287,
'means': 288, 'special': 289, 'smiles': 290, 'lucky': 291, 'fellow': 292, 'p
ark': 293, 'holds': 294, 'squeezes': 295, 'walking': 296, 'hours': 297, 'tal
king': 298, 'plan': 299, 'easy': 300, 'gently': 301, 'summer': 302, 'evenin
g': 303, 'breeze': 304, 'grow': 305, 'fingers': 306, 'soft': 307, 'light': 3
08, 'body': 309, 'velvet': 310, 'night': 311, 'soul': 312, 'slowly': 313, 's
himmer': 314, 'thousand': 315, 'butterflies': 316, 'float': 317, 'put': 318,
'rotten': 319, 'boy': 320, 'tough': 321, 'stuff': 322, 'saying': 323, 'nee
d': 324, 'anymore': 325, 'enough': 326, 'standing': 327, 'creep': 328, 'fel
t': 329, 'cheap': 330, 'notion': 331, 'deep': 332, 'dumb': 333, 'mistake': 3
34, 'entitled': 335, 'another': 336, 'beg': 337, 'forgive': 338, 'an': 339,
'feels': 340, 'hoot': 341, 'holler': 342, 'mad': 343, 'under': 344, 'heel':
345, 'holy': 346, 'christ': 347, 'deal': 348, 'sick': 349, 'tired': 350, 'te
dious': 351, 'ways': 352, 'aint': 353, 'walkin': 354, 'cutting': 355, 'tie':
356, 'wanna': 357, 'into': 358, 'eye': 359, 'myself': 360, 'counting': 361,
'pride': 362, 'unright': 363, 'neighbours': 364, 'ride': 365, 'burying': 36
6, 'past': 367, 'peace': 368, 'free': 369, 'sucker': 370, 'street': 371, 'si
nging': 372, 'shouting': 373, 'staying': 374, 'alive': 375, 'city': 376, 'de
ad': 377, 'hiding': 378, 'their': 379, 'shame': 380, 'hollow': 381, 'laughte
r': 382, 'while': 383, 'crying': 384, 'bed': 385, 'pity': 386, 'believed': 3
87, 'lost': 388, 'from': 389, 'start': 390, 'suffer': 391, 'sell': 392, 'sec
rets': 393, 'bargain': 394, 'playing': 395, 'smart': 396, 'aching': 397, 'he
arts': 398, 'sailing': 399, 'father': 400, 'sister': 401, 'reason': 402, 'li
nger': 403, 'deeply': 404, 'future': 405, 'casting': 406, 'shadow': 407, 'el
se': 408, 'fate': 409, 'bags': 410, 'thorough': 411, 'knowing': 412, 'late':
413, 'wait': 414, 'watched': 415, 'harbor': 416, 'sunrise': 417, 'sails': 41
8, 'almost': 419, 'slack': 420, 'cool': 421, 'rain': 422, 'deck': 423, 'tin
y': 424, 'figure': 425, 'rigid': 426, 'restrained': 427, 'filled': 428, 'wha
ts': 429, 'wrong': 430, 'enchained': 431, 'own': 432, 'sorrow': 433, 'tomorr
ow': 434, 'hate': 435, 'shoulder': 436, 'best': 437, 'friend': 438, 'rely':
439, 'broken': 440, 'feather': 441, 'patch': 442, 'walls': 443, 'tumbling':
444, 'loves': 445, 'blown': 446, 'candle': 447, 'seems': 448, 'hard': 449,
'handle': 450, 'id': 451, 'thinking': 452, 'went': 453, 'house': 454, 'hardl
y': 455, 'guy': 456, 'closing': 457, 'front': 458, 'emptiness': 459, 'he': 4
60, 'disapeared': 461, 'his': 462, 'car': 463, 'stunned': 464, 'dreamed': 46
5, 'lifes': 466, 'part': 467, 'move': 468, 'feet': 469, 'pavement': 470, 'ac
ted': 471, 'told': 472, 'lies': 473, 'meet': 474, 'other': 475, 'guys': 476,
'stupid': 477, 'blind': 478, 'smiled': 479, 'took': 480, 'said': 481, 'may':
482, 'couple': 483, 'men': 484, 'them': 485, 'brother': 486, 'joe': 487, 'se
eing': 488, 'lot': 489, 'him': 490, 'nice': 491, 'sitting': 492, 'sittin': 4
93, 'memories': 494}
495


/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:12: SettingWith
CopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  if sys.path[0] == '':
```

Dizilerin ve Etiketlerin Oluşturulması

Ön işlemeden sonra, sıralar ve etiketler oluşturmamız gerekiyor. Dizileri kendileri oluşturmak (`texts_to_sequences` ile öncekine benzer) ancak N-Grams kullanımını da içerir; etiketlerin oluşturulması artık bu dizileri kullanacak ve tüm potansiyel çıktı sözcükleri üzerinde `one-hot` kodlamayı kullanacak.

In [5]:

```
sequences = []
for line in corpus:
    token_list = tokenizer.texts_to_sequences([line])[0]
    for i in range(1, len(token_list)):
        n_gram_sequence = token_list[:i+1]
        sequences.append(n_gram_sequence)

# Eşit giriş uzunluğu için dolgu dizileri
max_sequence_len = max([len(seq) for seq in sequences])
sequences = np.array(pad_sequences(sequences, maxlen=max_sequence_len, padding='pre'))

# "Giriş" dizisi ve "çıkış" tahmin edilen kelime arasında dizileri bölme
input_sequences, labels = sequences[:,:-1], sequences[:,-1]
# Etiketlere one-hot kodlama
one_hot_labels = tf.keras.utils.to_categorical(labels, num_classes=total_words)
```

In [6]:

```
# Bazı verilerimizin nasıl saklandığını kontrol edin
# Tokenizer, kelime başına yalnızca tek bir dizine sahiptir
print(tokenizer.word_index['know'])
print(tokenizer.word_index['feeling'])
# Giriş dizileri birden çok dizine sahip olacak
print(input_sequences[5])
print(input_sequences[6])
# Ve one-hot etiket, tokenize edilmiş kelimelerin tam olarak yayılması kadar uzun olacaktır
print(one_hot_labels[5])
print(one_hot_labels[6])
```

```
32
97
[  0   0   0   0   0   0   0   0   0   0   0   0   0  81  82 142 197  29
   4]
[  0   0   0   0   0   0   0   0   0   0   0   0  81  82 142 197  29   4
 287]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

Metin Oluşturucu Modelin Eğitilmesi

Metin oluşturma modelimizi eğitmek için bir RNN oluşturmak, daha önce oluşturduğunuz duygu modellerine çok benzer olacaktır. Gerekli olan tek gerçek değişiklik, kayıp işlevi olarak İkili Çapraz Entropi (Binary Cross Entropy) yerine Kategorik'i (Categorical) kullandığınızdan emin olmaktır - daha önce duygu yalnızca 0 veya 1 olduğundan Binary'yi kullanabilirdik, ancak şimdi yüzlerce kategori var.

Metin oluşturmanın yakınsaması duygu analizine göre biraz daha uzun sürebileceğinden ve henüz o kadar çok veriyle çalışmıyoruz, çünkü bu noktadan sonra, öncekinden daha fazla dönem kullanmayı düşünmeliyiz. Veri kümesinin yalnızca bir kısmını kullandığımız için burada 200 çağa ayarlıyoruz ve eğitim bu kadar çok çağdan biraz daha geride kalacak.

In [7]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional

model = Sequential()
model.add(Embedding(total_words, 64, input_length=max_sequence_len-1))
model.add(Bidirectional(LSTM(20)))
model.add(Dense(total_words, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
history = model.fit(input_sequences, one_hot_labels, epochs=200, verbose=1)
```

```
Epoch 1/200
62/62 [==============================] - 10s 8ms/step - loss: 5.9906 - acc
uracy: 0.0283
Epoch 2/200
62/62 [==============================] - 1s 8ms/step - loss: 5.4349 - accu
racy: 0.0358
Epoch 3/200
62/62 [==============================] - 1s 8ms/step - loss: 5.3726 - accu
racy: 0.0399
Epoch 4/200
62/62 [==============================] - 0s 8ms/step - loss: 5.3119 - accu
racy: 0.0399
Epoch 5/200
62/62 [==============================] - 1s 9ms/step - loss: 5.2440 - accu
racy: 0.0318
Epoch 6/200
62/62 [==============================] - 1s 9ms/step - loss: 5.1822 - accu
racy: 0.0394
Epoch 7/200
```

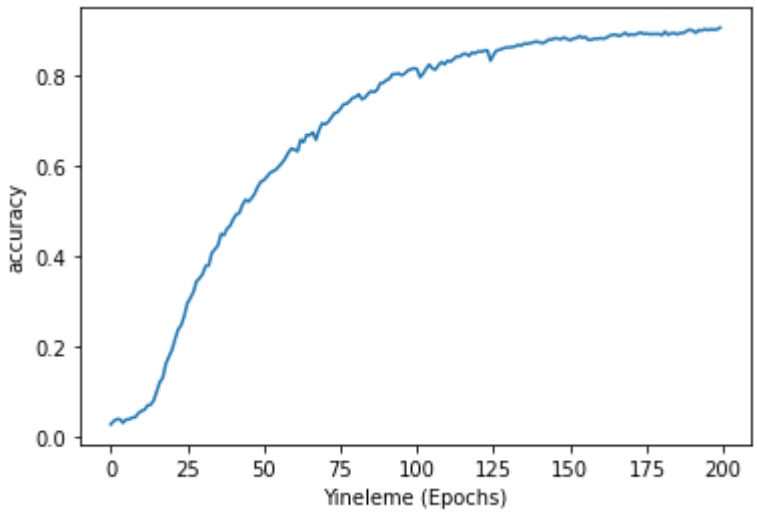
Eğitim Grafiğinin Oluşturulması

In [8]:

```
import matplotlib.pyplot as plt

def plot_graphs(history, string):
  plt.plot(history.history[string])
  plt.xlabel("Yineleme (Epochs)")
  plt.ylabel(string)
  plt.show()

plot_graphs(history, 'accuracy')
```



Yeni Şarkı Sözleri Oluşturalım

Sonunda, eğitilmiş modelden yeni sözler üretmenin ve ne elde ettiğimizi görmenin zamanı geldi. Bunu yapmak için, modelin başlaması için bir miktar "tohum metni" (seed text) veya bir giriş sırası sağlayacağız. Ayrıca bir çıktı dizisinin ne kadar uzun olmasını istediğimize de karar vereceğiz - girdi artı önceki çıktı sürekli olarak yeni bir çıktı kelimesi için besleneceğinden (en azından maksimum dizi uzunluğumuza kadar) bu aslında sonsuz olabilir.

In [9]:

```
seed_text = "im feeling chills"
next_words = 100

for _ in range(next_words):
    token_list = tokenizer.texts_to_sequences([seed_text])[0]
    token_list = pad_sequences([token_list], maxlen=max_sequence_len-1, padding='pre')
    predicted = np.argmax(model.predict(token_list), axis=-1)
    output_word = ""
    for word, index in tokenizer.word_index.items():
        if index == predicted:
            output_word = word
            break
    seed_text += " " + output_word
print(seed_text)
```

```
im feeling chills me to me every little touch touch lousy night night night
night night night dont surrender surrender dont would surrender sky and with
again and you gently more you leave me girl life us would feet feet night li
fe sun have fellow be bone learn figure tiny leaving feeling cry words more
of life of yourself once more more you leave more you leave talk feel talk n
ight night night question of girl surrender surrender surrender surrender su
rrender surrender surrender surrender surrender surrender surrender would an
dante behind dont surrender surrender surrender surrender would front feet n
ight night night night night
```