Doğal Dil İşlemeye Giriş
Ders Notu 3

3. HAFTA

ZEYNEP ALTAN

Düzgün İfadeler: Bağlayıcılar (Anchors) ^ $

Pattern	Matches
^[A-Z]	Palo Alto
^[^A-Za-z]	1 "Hello"
\.$	The end.
.$	The end? The end!

- Carat/şapka imi, ^ , satır başı ile eşleşir.
  - Carat köşeli parantez dışında ise **satır başı**, köşeli parantez içinde ise **olumsuzluk** anlamına gelir.
- $ ise **satır sonu** ile eşleşir.
- Nokta imi, . , varsa herhangi bir karakter yer alabilir
- Ters çizgi komutu ile kullanıldığında aynen (literally) uygulanan bir periyottur.
  - \.$ satır sonundaki bir periyot ile eşleşir.
    - Satır sonunda herhangi bir karakter olabilir.

Python Dilinde Düzgün İfadeler

- Regex ve Python özel karakterler için ters çizgi backslash "\" kullanır.
- Ek bir ters çizgi (backslashes) ile yazılmalıdır.
  - "\\d+" 1 veya daha fazla dijit aranmasıdır
  - "\n" Python'da "newline" yeni satır karakteridir.
    - ✓ not a "slash" followed by an "n".
  - "\\n" ile, iki karakter için n ifade edilir.
  - Bunun yerine: Python dilinin regex için **kaba dizgi notasyonu** r"[tT]he" ve r"\d+" yazılır. Burada bir veya daha fazla dijit eşleşir.
  - Bu gösterim \\d+ yerine yapılmıştır.

regex Python kod örneği -1

```
import re
#Check if the string starts with "The" and ends with "Spain":
txt = "The rain in Spain"
x = re.search("^The.*Spain$",  txt)
if x:
  print("YES! We have a match!")
else:
  print("No match")
```

YES! We have a match!

Metacharacters

Özel anlamı olan karakterlerdir.

[] Karakterlerin bir dizilişini simgeler. Örnek: "[a-m]"

Python kodu

```
import re
txt = "The rain in Spain"
#Find all lower case characters alphabetically between "a" and "m":
x = re.findall("[a-m]", txt)
print(x)
```

Output: ['h', 'e', 'a', 'i', 'i', 'a', 'i']

regex Python kod örneği -2

```
import re

txt = "hello planet"

#Search for a sequence that starts with "he",
followed by two (any) characters, and an "o":

x = re.findall("he..o", txt)
print(x)
  output: ['hello']
```

regex Python kod örneği -3

```
import re
txt = "The rain in Spain"
#Return a match at every NON word character (characters NOT
between a and Z. Like "!", "?" white-space etc.):
x = re.findall("\W", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output: [' ', ' ', ' ']
Yes, there is at least one match!

regex Python kod örneği - 4

```
import re
txt = "That will be 59 dollars"

#Find all digit characters:

x = re.findall("\d", txt)
print(x)
```

Output: ['5', '9']

regex Python kod örneği - 5

```
import re
txt = "The rain in Spain"
#Return a match at every NON white-space character:
x = re.findall("\S", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output: ['T', 'h', 'e', 'r', 'a', 'i', 'n', 'i', 'n', 'S', 'p', 'a', 'i', 'n']

Yes, there is at least one match!

regex Python kod örneği - 6

```
import re
txt = "8 times before 11:45 AM"
#Check if the string has any two-digit numbers, from 00 to 59:
x = re.findall("[0-5][0-9]", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output: ['11', '45']

Yes, there is at least one match!

Split() fonksiyonu
regex Python kod örneği - 7

split() fonksiyonu, her eşleşmede dizginin bölündüğü bir listeyi döndürür

```
import re
#Split the string at every white-space character:
txt = "The rain in Spain"
x = re.split("\s", txt)
print(x)
```

Output:['The', 'rain', 'in', 'Spain']

Sub() fonksiyonu
regex Python kod örneği - 8

sub() fonksiyonu metni seçilen ile yer değiştirir.

```
import re
#Replace all white-space characters with the digit "9":
txt = "The rain in Spain"
x = re.sub("\s", "9", txt)
print(x)
```

Output: The9rain9in9Spain

Search() fonksiyonu
regex Python kod örneği - 9

```
import re

#Search for an upper case "S" character in the beginning of a word, and print its position:

txt = "The rain in Spain"

x = re.search(r"\bS\w+", txt)

print(x.span())
```

Output: (12, 17)

regex Python kod örneği - 10

```
import re
#The string property returns the search string:
txt = "The rain in Spain"
x = re.search(r"\bS\w+", txt)
print(x.string)
```

Output: The rain in Spain

regex Python kod örneği - 11

```
import re
txt = "8 times before 11:45 AM"
#Check if the string has any digits:
x = re.findall("[0-9]", txt)
print(x)
if x:
  print("Yes, there is at least one match!")
else:
  print("No match")
```

Output: ['8', '1', '1', '4', '5']

Yes, there is at least one match!

Herhangi bir Örnekte Hata Yakalama

- « the» sözcüklerinin geçtiği bir düzgün ifade yazmak istiyoruz.
  - Düzgün ifadeyi /the/ olarak betimlemek doğru olmaz.
  - /[tT]he/ bir diğer gösterim şeklidir.
    - Burada da diğer sözcüklere gömülü metinler yanlış bir şekilde döndürülecektir.
  - Çıkarmak istediğimiz örneklerin her iki tarafına bir sınır (boundary) koyabiliriz: /\b[tT]he\b/
    - Eğer aynı işlemi /\b/ kullanmadan yazmak istersek, alt çizgiler ve sayılar kelime sınırları olarak ele alınmayacaktır
  - Bir aramada, , alt çizgiler veya sayılar olabilecek bir bağlamda «the» bulunmak istenebilir (the _ ya da the25).
    - Bu nedenle aranan terimin her iki tarafında da alfabetik harf olmayan örnekler istediğimizi belirtmemiz gerekir:

/[^a-zA-Z][tT]he[^a-zA-Z]/

Herhangi bir Örnekte Hata Yakalama

- Bu örnekte sorun devam eder: Yeni bir satıra başladığında «the» kelimesi bulunamayacaktır.
  - Çünkü «the» nın gömülü örneklerini önlemek için kullanılan [^a-zA-Z] düzgün ifadesi, «the» dan önce tek bir karakter olması gerektiğini ifade eder.
  - «the» dan önce satırın başlangıcını veya alfabetik olmayan bir karakteri ve satırın sonunda da aynısını belirtmek gerekir.

/(^|[^a-zA-Z])[tT]he([^a-zA-Z]|$)/

«false positive» ve «false negative» ne işe yarar?

Verilen örnekte yapılan farklı işlemler iki tür hatayı düzeltmeyi gerektir.

False Positives: Dizgiler yanlış olarak diğer dizgilerle eşleşir (there, others)

False Negatives :Eşleştirilmesi gereken şeyleri eşleştirememek (The)

Bu iki tür hatayla konuşma ve dil işleme sistemlerini uygularken çok karşılaşılır.

Bir uygulama için genel hata oranını azaltmak, bu nedenle iki karşıt çaba gerektirir.

Kesinliği /doğruluğu (accuracy /artırma (precision) (yanlış pozitifleri en aza indirme)

Geri çağırmayı artırma /Kapsamı /coverage) artırma (recall) (yanlış negatifleri en aza indirme)

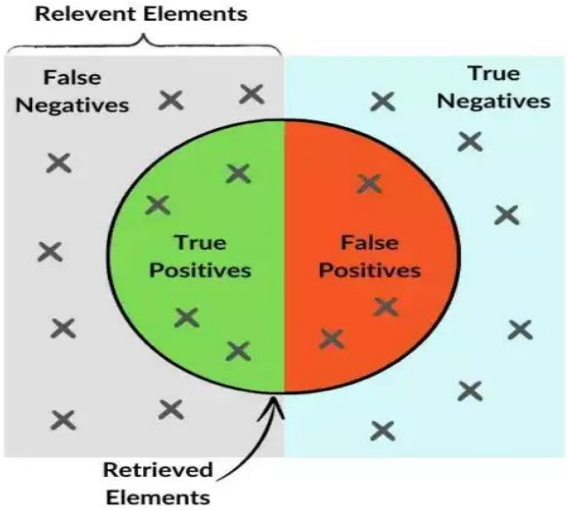
DDİ için Kullanımı

DDİ ‘de bu tür hatalarla her zaman karşılaşılabilir.

Bir uygulama için hata oranını azaltmak genellikle iki karşıt çaba içerir:

- Kapsamı (coverage) artırma (geri çağırmayı /recall)
  - (yanlış negatifleri en aza indirmek).
- Doğruluğu (accuracy) artırma (kesinlik /precision)
  - (yanlış pozitifleri en aza indirme)

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative



How many retrieved elements are relevant?

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

How many relevant elements are retrieved?

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

$$\text{False Negative Rate} \Rightarrow \frac{FN}{FN+TP}$$

Basic Text Processing

More Regular Expressions: Substitutions and ELIZA

Substitutions

```
s/regexp1/pattern/
```

e.g.:

```
s/colour/color/
```

Python "S" komutunu, regex tarafından eşleştirilen bir dizgeyi, başka bir dizgeyle değiştirmek için kullanır.

Yakalama Grupları / Capture Groups

- Tüm sayıların baş ve sonuna < > konulmak istenebilir.
  the 35 boxes → *the <35> boxes*
- () kullanılarak sayılardan oluşan bir kayıta ait örnek (1, 2, 3…) yakalanabilir
- \1 kullanılarak kaydın içeriklerine atıf yapılır.
- `s/([0-9]+)/<\1>/`
- İlk örnekle *eşleşen dizgenin belirli bir alt bölümüne* erişmek gerekebilir.
  - Bunun için, örneğin bir bölümünü bir register üzerine kaydetmenin bir yolu olan "*yakalama grupları*" kullanılabilir.
    - ✓ Böylece daha sonra yerine geçme dizgesinde kullanılabilir.

Yakalama grupları: çoklu kayıtlar / Capture groups: multiple registers

```
/the (.*)er they (.*), the \1er we \2/
```

the faster they ran, the faster we ran

cümlesi ile eşleşir. Fakat;

the faster they ran, the faster we ate

cümlesi ile eşleşmez.

Herhangi bir grup için yakalanma (capturing) istenmiyorsa !!

Parantezlerin iki farklı işlevleri vardır: *Grouping Terms* ve C*apturing*

Grubun yakanmasını önlemek için parantezden sonra ? kullanılır.

```
/(?:some|a few) (people|cats) like some \1/
```

düzgün ifadesi

- `some cats like some cats`

cümlesi ile eşleşir. Ama,

- `some cats like some some`

cümlesi ile eşleşmez.

- Bu örnek sadece *gruplama* içerir, yakalama içermez.

İleriye Yönelik İddialar / Lookahead Assertions

`(?= pattern)` örnek eşleşirse doğrudur.

`(?! pattern)` örnek eşleşmezse doğrudur.

```
/^(?!Volcano)[A-Za-z]+/
```

Düzgün ifadesi ile bir satırın başında, "Volcano" ile başlamayan herhangi bir tek kelimeyle eşleşme gerçekleşir

ELIZA /Joseph Weizenbaum, 1966

Substitutions ve *capture groups* sohbet robotlarını uygulamada oldukça başarılıdır.

Uses pattern matching to match

"I need X"

and translates them into

"What would it mean to you if you got X?

- X'e ihtiyacım var" gibi ifadeleri tanımak ve bunları "X'e sahip olsaydın bu senin için ne anlama gelirdi?" gibi uygun çıktılara çevirmek için desen eşleştirmeyi kullanan basit bir programdır.
  - Bu basit yöntem başarılıdır.
    - ELIZA'nın bir psikoterapisti taklit etmek için hiçbir şey bilmesine gerek yoktur
    - Bu, dinleyicilerin hiçbir şey bilmiyormuş gibi davranabilecekleri birkaç diyalog türünden biridir.

Simple Application: ELIZA

Men are all alike.
IN WHAT WAY

They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE

Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE

He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED

ELIZA Nasıl Çalışır?

s/.* I'M (depressed|sad) .*/I AM SORRY TO HEAR YOU ARE \1/

s/.* I AM (depressed|sad) .*/WHY DO YOU THINK YOU ARE \1/

s/.* all .*/IN WHAT WAY?/

s/.* always .*/CAN YOU THINK OF A SPECIFIC EXAMPLE?/

- ELIZA bir dizi substitution (yerine geçen) desenden oluşur.
- Hangi desenin seçileceği, daha üst düzeyde diyalog yapısını araştırır.
  - Sıfatları yakalayacak gruplar araştırılıyor.
  - Kullanıcının kendisini tanımlamak için yazdığı sıfatları yakalamak için *yakalama gruplarının* örneklerini ve kullanıcı "all" veya "always" içeren genel ifadeler kullandığında daha fazla ayrıntı sormak için kalıplar görülüyor.

Weinbaum'un ELIZA için akıllı tanımı

Aşağıdaki kullanıcı cümlesi için:

I know everybody laughed at me

I ile başlayan bir cümle olduğu için şu kuralla eşleşmelidir. keyword is I:

I (.*) -> You say you \1

producing:

YOU SAY YOU KNOW EVERYBODY LAUGHED AT YOU

Bir başka ELIZA örneği

(.*) YOU (.*) ME -> WHAT MAKES YOU THINK I \2 YOU

Bir kullanıcı cümlesini «You hate me» gibi bir sistem cevabına dönüştürmek için yukarıdaki düzgün ifade kullanıldığında sonuç:

WHAT MAKES YOU THINK I HATE YOU

olacaktır.

Basic Text Processing

Words and Corpora

How Many Words in a Sentence?

- *"I do uh main-mainly business data processing"*

Bu cümlede kaç kelime var? "Uh" bir kelime midir? «mainly" öncesindeki "main" kesmesi ne olacaktır?

"main" gibi bildirimlere *parça* denir; "uh" ve "um" ise *duraklamalardır.*

Konuşma uygulamaları gibi belirli uygulamalarda bunların sayılması istenebilir.

- "*Seuss's cat in the hat is different from other cats!"*
  - **Lemma**: aynı köke (*stem*), aynı sözcük türüne (*POS*), aynı anlama(*word sense*) sahiptir.
    - ✓ cat ve cats = same lemma
  - **Wordform**: tüm çekimleri veya son ekleriyle kelimenin tam yüzey (surface) biçimidir.
    cat ve cats = farklı sözcük formları

How Many Words in a Sentence?

They lay back on the San Francisco grass and looked at the stars and their

Type: kelime dağarcığının bir öğesi

Token (Belirteç /Andaçlama): bu türün metindeki bir örneği

Kaç tane "token" ve "type" içerilir?

15 tokens (or 14)

13 types (or 12) (or 11?)

- Kelime türleri (types), cümlede geçen tekil (unique) kelime sayısı olarak seçilebilir.
  - Bu sayımda, iki kez görünse bile, «the» ve «and» bir kez sayılır.
- Kelime belirteçleri (tokens), sayfadaki her kelime belirtecini sayabilir; bu yüzden iki "the" iki kez sayılır.
  - San Francisco? bir kelime mi yoksa iki kelime mi?
  - "they ve their " için nasıl kararlar alınabilir?
- Farklı kelime biçimleri ve aynı önermenin belirlenme şekli, hedefe bağlıdır ve kelime sayıları bildirildiğinde bu açıkça belirtilmelidir.

word_form Python örneği

```
from word_forms.word_forms import get_word_forms
word_forms = get_word_forms("president")
print(word_forms)
```

Output:

```
{
   'n': {'presidents', 'presidentships', 'presidencies', 'presidentship',
                                                    'president', 'presidency'},
   'a': {'presidential'},
   'v': {'preside', 'presided', 'presiding', 'presides'},
   'r': {'presidentially'}
}
```

Derlemin Boyuta göre Değişmesi

- **Language**: Dünyada 7097 dil vardır. Bu nedenle derlem oluşturmak için belli koşyllara uygmak zorunludur.
- **Variety** *(Çeşitlilik): Örneğin African American Language (AEE) bir* çeşittir.
  - AAE Twitter postları "*iont" (I don't)* gibi formlar içerecektir.
- **Code switching***(Kod değiştirme): Spanish(S) /English(E) , Hindi(H)/English (E)* dilleri arasında bir örnek aşağıdadırçç

S/E: Por primera vez veo a @username actually being hateful! It was beautiful:)

[For the first time I get to see @username actually being hateful! it was beautiful:)]

şeklinde değiştirilir

Dünyadaki Dil Sayısı (Unesco)

https://en.wal.unesco.org/world-atlas-languages

- Dünya Dil Atlası, hükümetler, kamu kurumları ve akademik topluluklar tarafından belgelenen 8324 konuşulan veya imzalanan dil olduğunu belirlemiştir.
- 8324'ten yaklaşık 7000 dilin hala kullanıldığı belirtilmektedir.
- Bu atlasta her bir dil, türüne, yapısına ve bağlılığına, durumuna, statüsüne, işlevlerine, kullanıcılarına ve kullanımına göre işaretlenmiştir.

Bir derlemde /corpus kaç sözcük vardır?

N = number of tokens / andaç (belirteç) sayısı

V= Tüm kelimelerin kümesi V

V'nin kardinalitesi , |V|, kelime dağarcığının büyüklüğü, kelime türlerinin sayısıdır.

Heaps Law = Herdan's Law = $|V| = kN^{\beta}$ $.67 < \beta < .75$

Sözcük büyüklüğü kare kökten büyük büyüklükte büyür

	Tokens = N	Types = |V|
Switchboard phone conversations	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
COCA	440 million	2 million
Google N-grams	1 trillion	13+ million

Yorum

Google n- gramlarında on üç milyondan fazla tip bulunmaktadır.

Bazıları muhtemelen URL'ler ve e-posta adresleridir. Bunların hepsi kaldırılsa bile, bir dildeki kelime sayısı çok fazladır.

Muhtemelen bir milyon İngilizce kelime vardır.

Corpus /Derlem

Herhangi bir metin:

- Belli bir kişi tarafından,
- Belli bir zamanda
- Belli bir çeşitlilikte,
- Belirli bir dilde,
- Belirli bir amaç için hazırlanır.

Derlemin Boyuta göre Değişmesi

- **Language**: Dünyada 7097 dil vardır. Bı nedenle derlem oluşturmak için belli koşyllara uygmak zorunludur.
- **Variety** *(Çeşitlilik): Örneğin African American Language (AEE) bir* çeşittir.
  - AAE Twitter postları "*iont" (I don't)* gibi formlar içerecektir.
- **Code switching***(Kod değiştirme): Spanish(S) /English(E) , Hindi(H)/English (E)* dilleri arasında bir örnek aşağıdadırçç

S/E: Por primera vez veo a @username actually being hateful! It was beautiful:)

[For the first time I get to see @username actually being hateful! it was beautiful:)]

şeklinde değiştirilir

Dünyadaki Dil Sayısı (Unesco)

https://en.wal.unesco.org/world-atlas-languages

- Dünya Dil Atlası, hükümetler, kamu kurumları ve akademik topluluklar tarafından belgelenen 8324 konuşulan veya imzalanan dil olduğunu belirlemiştir.
- 8324'ten yaklaşık 7000 dilin hala kullanıldığı belirtilmektedir.
- Bu atlasta her bir dil, türüne, yapısına ve bağlılığına, durumuna, statüsüne, işlevlerine, kullanıcılarına ve kullanımına göre işaretlenmiştir.

Derlemin Boyuta göre Değişmesi

H/E: dost tha or ra- hega ... dont wory ... but dherya rakhe

["he was and will remain a friend ... don't worry ... but have faith"]

şeklinde değiştirilir.

KESİNLİKLE makine çevirisi değildir.

Derlem oluşturulurken aşağıdakiler de göz önüne alınır:

Tür (Genre) : haber (newswire), kurgu (fiction), bilimsel makaleler (scientific articles), Wikipedia

Yazar Demografisi: yazarın yaşı, cinsiyeti, etnik kökeni,

Örneğin, metin hangi dildedir?

Kelime belirteçlemesi (tagging), bir kelimenin ne olarak sayılacağını belirtir, bu belirtim farklı dillerde farklı olabilir.