İbrahim Okan Akveç

1238105104

02.04.2024

Doğal Dil İşleme: Metin Önişleme Uygulama Ödevi

Doğal Dil İşleme dersi Metin Önişleme Uygulama Ödevi kapsamında seçilen bir corpus üzerinde Türkçe dilinde aşağıda listelenmiş adımlar taker taker uygulanmıştır.

1. Sentence Segmentation
2. Tokenization
3. Stemming
4. Lemmazation
5. Extract Stopwords
6. Part of Speech
7. Remove Punctuations

Ödev Google Colab üzerinden uygulanmıştır. COLAB LİNK

https://colab.research.google.com/drive/1ZwlGDD4gGToTJ4VY-ybDHhZSjG8ulO7m#scrollTo=3u43-wDekg-u

Metin önişleme işlemleri için Stanford NLP Group tarafından 2020 yılı içerisinde ilk sürümü yayınlanmış olan STANZA kütüphanesi kullanılmıştır.

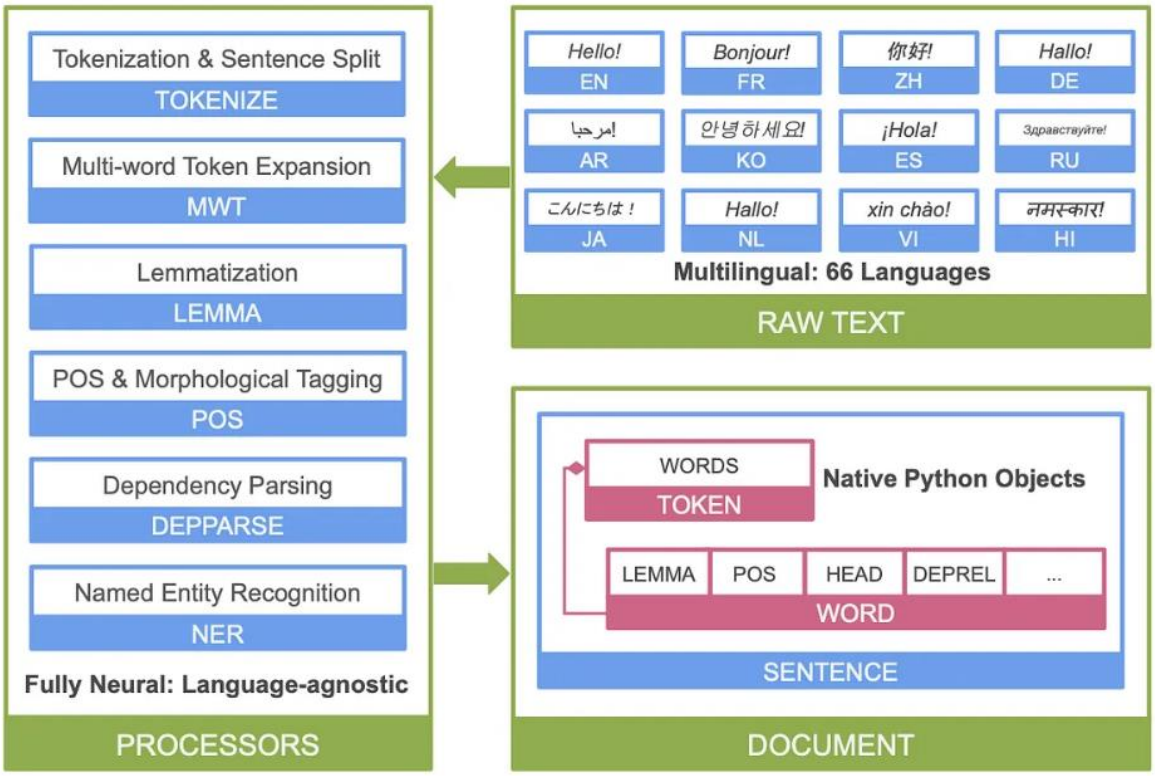
STANZA kütüphanesi PYTHON dili için geliştirilmiş bir uygulama kütüphanesidir. 70 farklı dil ile çalışabilecek şekilde tasarlanmıştır. Desteklediği diller için free-format metinden kelime ve hece sentezleme, morphologic analiz ve (N)ER gibi özellikler sunmaktadır

System	# Human Languages	Programming Language	Raw Text Processing	Fully Neural	Pretrained Models	State-of-the-art Performance
CoreNLP	6	Java	✓		✓	
FLAIR	12	Python		✓	✓	✓
spaCy	10	Python	✓		✓	
UDPipe	60	C++	✓		✓	✓
Stanza	66	Python	✓	✓	✓	✓

STANZA'nın mümkün olan en iyi performansını açığa çıkarabilmesi için, eğitim süreçlerinde yapay sinir ağı tekniklerinden faydalanmaktadır. Eğitim süreçlerinde kullanılan bu modüller Pytorch kütüphanesi üzerine kurulmuştur. Dolayısıyla eğitim sürecinde GPU donanımına sahip bir makinede kullanılırsa çok daha verimli ve yüksek bir performans elde edilebilir.

Ek olarak Stanza, Stanford CoreNLP uygulamasına erişim ve oradaki fonksiyonlarında kullanılabilmesi için bir Python ara katmanı sunmaktadır. Bu sayede Java dilinde olan CoreNLP uygulamasının paketleri ve oradaki tüm methodları kullanılabilir. Örneğin linguistic pattern matching gibi fonksiyonları şuan da Stanza içerisinde barındırmadığı için CoreNLP üzerinden sorunsuz kullanılabilmektedir.

Stanza teknik altyapısı ise aşağıdaki gibidir.



STANZA'nın şuanki sürümü TOKENIZE, LEMMA, POS, NER işlemlerini yerine getirebilmektedir. Bu yüzden ödevin uygulama süreçinde bazı konularda NLTK kütüphanesi de kullanılmıştır.

1) Sentence Segmentation

Cümlelere Ayırma işlevi için STANZA pipeline'ı türkçe dili ile "tokenize" processor kullanılarak inşa edilmiştir. Daha sonra bu pipeline'a corpus verisi geçilmiş, ve dönen document nesnesi üzerinden her bir cümleye erişilmiştir.

```
import stanza
nlp = stanza.Pipeline('tr', processors='tokenize', tokenize_no_ssplit=False)
doc = nlp('Gökyüzü masmaviydi, bulutlar pamuk şeklinde birbirlerine yaklaşıp dağılıyordu. Rüzgar, yapraklar
for sentence in doc.sentences:
  print(sentence.text)
```

```
Gökyüzü masmaviydi, bulutlar pamuk şeklinde birbirlerine yaklaşıp dağılıyordu.
Rüzgar, yaprakları hafifçe sallayarak baharın gelişini müjdeliyordu.
İnsanlar sokaklarda neşeyle dolaşıyor, kuşlar ağaç dallarında cıvıldaşıyordu.
Ancak, küçük bir kasaba olan Ahmetlide, beklenmedik bir olay herkesin dikkatini çekti.
Gece yarısı, gizemli bir şekilde ortadan kaybolan eski çınar ağacı, kasaba halkını endişeye sevk etti.
Kimi insanlar ağacın büyülü olduğunu söylerken, diğerleri mantıklı bir açıklama arıyordu.
Bu esrarengiz olayın ardından, kasabanın genç ve meraklı dedektifi Ayşe, gizemi çözmek için harekete geçti.
İzleri sürerken, eski kasaba efsaneleriyle gerçek dünya arasındaki sınırların bulanıklaştığı bir maceraya atıld
Ancak, Ayşenin fark etmediği şey, gizemin çözümünde asıl ipuçlarının insanların kalplerinde ve geçmişlerinde ol
Bu, Ayşenin sadece gizemi çözmekle kalmayıp aynı zamanda kendi iç dünyasını keşfettiği bir yolculuktu.
Ahmetli kasabasında başlayan bu serüven, aslında insan doğasının derinliklerine yapılan bir yolculuğun başlangı
```

2) Tokenization

Birimlere Ayırma işlevi için cümlelere ayırma işlevinde olduğu gibi, her bir cümlenin tokens fonksiyonu ile erişilmiştir.

```
import stanza

nlp = stanza.Pipeline(lang='tr', processors='tokenize')
doc = nlp('Gökyüzü masmaviydi, bulutlar pamuk şeklinde birbirlerine yaklaşıp dağılıyordu. Rüzgar, yapraklar
for i, sentence in enumerate(doc.sentences):
    print(f'====== Sentence {i+1} tokens =======')
    print(*[f'id: {token.id}\ttext: {token.text}' for token in sentence.tokens], sep='\n')
```

```
====== Sentence 1 tokens =======
id: (1,)        text: Gökyüzü
id: (2, 3)      text: masmaviydi
id: (4,)        text: ,
id: (5,)        text: bulutlar
id: (6,)        text: pamuk
id: (7,)        text: şeklinde
id: (8,)        text: birbirlerine
id: (9,)        text: yaklaşıp
id: (10,)       text: dağılıyordu
```

3) Stemming

STEMMİNG için önce STANZA kütüphanesinden tokenlar alınmış, daha sonra NLTK kütühanesinin PorterStemmer nesnesi ile stem'lere erişilmiştir.

```
import nltk
from nltk.stem import PorterStemmer
nltk.download("punkt")
ps = PorterStemmer()

import stanza
nlp = stanza.Pipeline(lang='en', processors='tokenize')
doc = nlp('The sky was azure, with clouds approaching and dispersing in cotton-like formations. The wind gen
for i, sentence in enumerate(doc.sentences):
    print(f'====== Sentence {i+1} tokens =======')
    for token in sentence.tokens:
      print ("{0:20}{1:20}".format(token.text, ps.stem(token.text)))
```

```
sky                 sky
was                 wa
azure               azur
,                   ,
with                with
clouds              cloud
approaching         approach
and                 and
dispersing          dispers
in                  in
cotton              cotton
-                   -
like                like
formations          format
```

4) Lemmazation

Stanza kütüphanesi ile pipeline oluştururlurken lemma processor'u de kullanılmış, Doc.sentences.words.lemma özelliği ile her bir lemma'ya erişimiştir.

```
import stanza

nlp = stanza.Pipeline(lang='tr', processors='tokenize,mwt,pos,lemma')
doc = nlp('Gökyüzü masmaviydi, bulutlar pamuk şeklinde birbirlerine yaklaşıp dağılıyordu. Rüzgar, yaprakla
print(*[f'word: {word.text+" "}\tlemma: {word.lemma}' for sent in doc.sentences for word in sent.words], s
```

```
word: Gökyuzu   lemma: gökyuzu
word: masmavi   lemma: masmavi
word: ydi       lemma: i
word: ,         lemma: ,
word: bulutlar  lemma: bulut
word: pamuk     lemma: pamuk
word: şeklinde  lemma: şekil
word: birbirlerine      lemma: birbiri
word: yaklaşıp  lemma: yaklaş
```

5) Extract Stopwords

Etkisiz Sözcük çıkarımı için STANZA'nın desteği olmadığından NLTK kütüphanesi kullanılmıştır. NLTK tükrçe stopwords kelimeleri, corpusta tek tek gezilerek corpustan çıkartılmıştır.

```
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')
print(stopwords.words('turkish'))
word_list = 'aslında Gökyüzü masmaviydi, bulutlar pamuk şeklinde birbirlerine yaklaşıp dağı
filtered_words = [word for word in word_list if word not in stopwords.words('turkish')]
print(filtered_words)
```

NLTK kütüphanesinin türkçe için birkaç stopwords örneği:

```
['acaba', 'ama', 'aslında', 'az', 'bazı', 'belki', 'biri', 'birkaç', 'birşey', 'biz', 'bu', 'çok', 'çünkü', 'da
```

6) Part of Speech

Sözcük türü etiketleme işlevi için yine lemmalama bölümündeki gibi her bir kelimeye erişilmiş, fakat bu kez her bir word'un feats özelliği ile part of speech değerlerine erişilmiştir.

```
import stanza

nlp = stanza.Pipeline(lang='tr', processors='tokenize,mwt,pos')
doc = nlp('Gökyüzü masmaviydi, bulutlar pamuk şeklinde birbirlerine yaklaşıp dağılıyordu. Rüzgar, yaprakları
print(*[f'word: {word.text}\tupos: {word.upos}\txpos: {word.xpos}\tfeats: {word.feats if word.feats else "_"
```

```
word: Gökyüzü   upos: NOUN      xpos: Noun      feats: Case=Nom|Number=Sing|Person=3
word: masmavi   upos: ADJ       xpos: NAdj      feats: Case=Nom|Number=Sing|Person=3
word: ydi       upos: AUX       xpos: Zero      feats: Aspect=Perf|Mood=Ind|Number=Sing|Person=3|Tense=Past
word: , upos: PUNCT     xpos: Punc      feats: _
word: bulutlar  upos: NOUN      xpos: Noun      feats: Case=Nom|Number=Plur|Person=3
word: pamuk     upos: NOUN      xpos: Noun      feats: Case=Nom|Number=Sing|Person=3
word: şeklinde  upos: NOUN      xpos: Noun      feats: Case=Loc|Number=Sing|Number[psor]=Sing|Person=3|Person
word: birbirlerine      upos: PRON      xpos: Quant     feats: Case=Dat|Number=Plur|Number[psor]=Plur|Person=
word: yaklaşıp  upos: VERB      xpos: Verb      feats: Aspect=Perf|Mood=Ind|Polarity=Pos|Tense=Pres|VerbForm=
word: dağılıyordu       upos: VERB      xpos: Verb      feats: Aspect=Prog|Mood=Ind|Number=Sing|Person=3|Pola
```

7) Remove Punctuations

Noktalama işaretleri kaldırma işlevi için STANZA kütüphanesi desteklemediğinden dolayı NLTK kütüphanesi kullanılmıştır. Burada yine türkçe stopwords gibi noktalama işaretleri NTLK sözlüğünden alınmış, ve corpus gezilerek tek tek çıkarılmıştır.