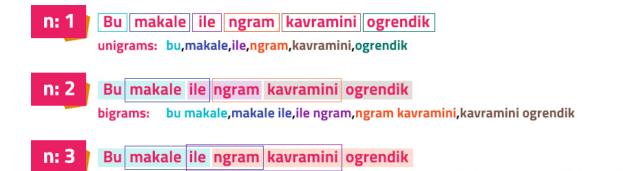
Bil 211 - Lab 3

Bir metindeki N ardışık kelimelerin oluşturduğu gruplara N-gram denir (N > 0). N=1 (unigram), N=2 (bigram), N=3 (trigram) örneklerini aşağıda görebilirsiniz:



Gereklilikler

 Bu labda size input olarak verilen bir dosyadaki n-gramları bulmanız ve bunları yeni bir dosyaya yazmanız beklenmektedir. Bunun için Ngram sınıfı içinde extractNgrams() metodu oluşturunuz.

trigrams: bu makale ile, makale ile ngram, ile ngram kavramini, ngram kavramini ogrendik

- Output dosyasında her satırda bi ngram ve sonrasında parantez içinde o ngramın input dosyasında kaç kere bulunduğu (frekans) yazmalı.
- Dolayısıyla bir ngram input dosyasında birden fazla defa bulunsa bile output dosyasında yalnızca tek bir ngram olarak bulunmalı.
- Ngramlar, input dosyasındaki bulunma sırasına göre output dosyasına yazılmalı. Ngramların output dosyasına eklenme sırası, çıktınızın testlerle birebir örtüşmesi açısından önemli.
- Ardışık kelimelerin aynı n-gram grubu içinde sayılması için aynı cümle içinde yer alması gerekir. Örneğin bir cümlenin son kelimesi ile sonraki cümlenin ilk kelimesi bir bigram oluşturmaz. Nokta karakteri bulunduğunda cümlenin bittiğini varsayabilirsiniz.
- Output dosyasına yeni bir n-gram eklerken kendi yazacağınız updateNgram() fonksiyonunu kullanın.
- Ngram'ları bulurken büyük harfler, küçük harflere çevrilmeli. Örneğin, aynı kelimelerden oluşan n-gram grupları, büyük-küçük harf olarak farklılık gösterseler bile aynı olarak düşünülmeli.
- Ngram'ları tespit ederken noktalama işaretlerini yok sayın. (Bunun için internette bulduğunuz bir yöntem/fonskiyon vs. kullanabilirsiniz)
- Kodunuzda file IO için try/catch bloklarını mutlaka kullanın.
- Kelimeleri tespit ederken space karakterini delimiter olarak kullanın.
- Beklenen çıktıyı daha iyi anlamak için sizinle paylaşılan örnek input ve output dosyalarını inceleyin. Kodunuzun birebir aynı çıktıyı ürettiğinden emin olun.
- Kodunuz değerlendirilirken puan alabilmeniz için beklenen çıktı ile birebir örtüşmelidir.

public void extractNgrams(String input_file, String output_file, int[] N_list)

- input_file: input olarak kullanılacak metin dosyası
- output_file: n-gramların yazılacağı output dosyası
- **N_list:** integer listesi. Fonksiyonunuz bu listedeki her N sayısı için metin dosyasındaki N-gramları bulacak. Örneğin [1,3,5] verildiğinde, oluşacak output dosyasında hem unigram'lar, hem trigramlar, hem five-gram'lar bulunmalı.

Sizinle paylaşılan örnek girdi ve çıktıları inceleyebilirsiniz.

public void updateNgrams(String ngram_str, String output_file)

- Input dosyasında bulduğunuz her n-gram'dan sonra bu fonksiyonu kullanarak output dosyasını güncellemelisiniz.
- Fonksiyon, string olarak verilen bir n-gram'ı output dosyasına ekler. Eğer verilen ngram, daha önce output dosyasına eklenmişse, sadece o satırdaki frekans değerini bir arttırır. Eğer output dosyasında bu ngram bulunmuyorsa, en sona yeni bir satır olarak ekler.
- Bu fonksiyonu, **extractNgrams** içinde çağırarak kullanmalısınız.

Örnek.

```
String ngram_str = "hello from"
Output.txt:
hello from (1)
from the (1)
the other (3)
other side (2)
updateNgram(ngram_str, "output.txt") metodu çağırıldıktan sonra output.txt'nin son hali:
Output.txt:
hello from (2)
from the (1)
the other (3)
other side (2)
String ngram_str = "hello mate"
Output.txt:
hello from (1)
from the (1)
the other (3)
other side (2)
updateNgram (ngram str, "output.txt") metodu çağırıldıktan sonra output.txt'nin son hali:
```

Output.txt: hello from (1) from the (1) the other (3) other side (2) hello mate (1)

Çıktı Formatı:

<token> <token> ... <token> (<frekans>)\n

Output dosyasında her bir satırda bir ngram ve o ngram'ın kaç defa input dosyasında bulunduğu (frekans) yazmalıdır. Ngramı oluşturan token'lar arasında ve frekans yazılmadan önce boşluk bırakılmalıdır.

Örnek:

hello from the (5) from the other (2)