Projenin Amacı

En az 30.000 kelimelik Türkçe kelimeler sözlüğü için tahmin edilen kelimeyi kullanıcının verdiği ipuçları ile bulmak.

Giriş

Problemde kullanıcının sözlük içinden seçtiği kelimenin bilgisayar tarafından bilinmesi gerekmektedir. Bunun için kullanıcın, bilgisayarın tahmin olarak verdiği kelime ile ortak harf sayısının normal ve Kartezyen karşılaştırma sonuçlarının bilgisayara verilmesi beklenmektedir. Aynı şekilde Kullanıcı-Bilgisayar olabileceği gibi Bilgisayar-Kullanıcı da olabilmektedir. Problem daha önceden Donalt Knuth tarafından ele alınmış olan Mastermind problemi ile doğrudan alakalıdır. Mastermind probleminin çözümü için iki algoritma ele alınmış olup bunlar gelen ipucuna göre örnek uzay oluşturup buradan tahmin etmek ve bütün olasılıkların arasından eleyerek sonuca ulaşmaktır.

Gelişme

# Mastermind

Edindiğimiz bilgiler ışığında elimizde sözlük olduğunu düşünürsek ikinci algoritmayı kullanmak daha mantıklı bir seçim olacaktır. Algoritmanın amacı örnek uzayın yani sözlükteki kelimelerin sayısını bire düşürmek olduğundan en çok kelime eleme ihtimali olan kelimenin sormak için en iyi olduğunu bulabiliriz. Knuth algoritması olası sonuçlar içinden kelimeleri elemek için başlangıçta seçilecek en iyi kelimelerin aabb, ccdd gibi tekrarlı harfler içeren kelimeler olduğunu öne sürmekte. Ancak sözlükteki kelime sayımız azaldığından bu yöntemin çok fazla efektif olmadığı anlaşılabilir.

# Shannon’s Enthropy

Ardından karşımıza çıkan bir kelimenin, sözlükteki hiçbir diğer kelimeyi elemediğini gördük. Bunun üstüne bütün kelimeleri diğerleriyle karşılaştırıp hangi kelimelerin sormak için daha iyi olacağını gördük \*¹. Bu işlemi bilgisayarda formülize için “Shannon’s Enthropy” \*² teorisini kullandık. Daha yüksek entropi daha fazla kelime eleme anlamına geldiği için formülden yararlanarak, tüm kelimeleri diğer tüm kelimeler ile karşılaştırıp entropisi en yüksek olan kelimeyi sormak istedik. Fakat bu çok fazla işlem yükü gerektirdiğinden son bin kelime için bu algoritmayı kullandık. Zaten Knuth başlangıç için en iyi kelimelerin tekrarlı kelimeler olduğunu söylemişti.

# Anagram kelimeler

Başka bir problem olan anagram kelimeler sorununa eğildiğimizde gördük ki anagram kelimelerin birbirlerine göre tek fark harflerinin yerleridir. Buradan hareketle normal veya Kartezyen karşılaştırma ile anagram kelimelerin herhangi birini eleyemezdik. Bunun için kelimenin momenti dediğimiz bir formül oluşturduk \*³. Fakat kullanıcıdan moment ile ilgili herhangi bir bilgi alamayacağımızdan sezgisel bir yöntem oluşturmaya çalıştık(momentlerin ortalamasını sor, en büyüğünü sor, ya da en küçüğünü sor gibi). Yine de tam olarak başarıya ulaşamadığımız için moment fikrinden vazgeçtik.

Sonuç

Son olarak belirginleşen genel algoritma şu şekilde: ilk tahminlerde Donald Knuth’un önerdiği gibi başlangıç için tekrarlı ve uzun kelimeler en iyi seçimdir. Ardından tabii ki bilgisayar hızına göre kalan kelime sayısı belli bir sınırın altına düştüğünde Shannon’s Enthropy algoritması ile en iyi seçim olan kelimeyi bulur. Burada dikkat edilmesi gereken faktör Shannon’s Enthropy algoritmasının bulduğu kelimeler her zaman en iyi çözüm değildir fakat en optimum çözümdür. Tutarlılığı arttırır. Rastgele seçilen bir kelime daha iyi sonuç verebilir fakat çok kötü sonuç da verebilir.

Problem çözümümüzün başarılı olup olmadığını görmek için kısa, uzun, tekrarlı ve tekrarsız kelimeleri 1,140,208 kelimelik Türkçe sözlük ile deneyip sonuçları yorumlayabiliriz.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1.Deneme | 2. Deneme | 3. Deneme | 4. Deneme | 5. Deneme | ortalama |
| Çare | 8 | 7 | 8 | 6 | 5 | 6,8 |
| Sevince | 5 | 10 | 7 | 9 | 10 | 8,2 |
| Sokağınızda | 5 | 5 | 6 | 5 | 6 | 5,4 |
| Özgüleşir | 5 | 4 | 4 | 5 | 5 | 4,6 |
| Ağ | 5 | 5 | 7 | 5 | 6 | 5,6 |
| Uyu | 6 | 6 | 6 | 7 | 6 | 6,2 |
| Pohpohlamak | 4 | 6 | 6 | 6 | 5 | 5,4 |

Opimizasyon

Shannon’s Enthropy gibi işlem yükü gerektiren algoritma kullandığımız için optimize kod daha hızlı çalışacağından kodu olabildiğince optimize etmeye çalıştık. Karşılaştırmayı, enthropy hesaplamayı, kelimenin sözlük içinde olup olmaması gibi fonksiyonları en sade haliyle yazdık. Dosya ismi gibi sabit değişkenleri ön işlemci komutları ile tanımlayıp, döngü içinde bir fonksiyon çağırmak yerine fonksiyonu bir değişkene atayıp onun üstünden işlem yaparak derleme-çalıştırma süresinden tasarruf etmiş olduk. Bunun haricinde Shannon’s Enthropy algoritmasını uygularken her kelimeyi diğerleriyle karşılaştırmak O(n2) yerine iki boyutlu bir dizi içinde karşılaştırmalar hafızada tutulup O(n2/2) toplam karşılaştırma sayısının yarısından kaçınabilirdik. Fakat gerek süre gerek kod yoğunluğu gerekte ne kadar süre tasarrufu edeceğimizi bilemediğimizden bu fikri gerçekleştirmedik.

# Kaynaklar

1,140,208 kelimelik Türkçe sözlük: <https://github.com/ncarkaci/TDKDictionaryCrawler/blob/master/Zemberek_S%C3%B6zl%C3%BCk_Kelime_Listesi.txt>

180011 kelimelik Türkçe sözlük: <http://ceng.ktu.edu.tr/dsy/sozluk.txt>

Prof. Dr. Vasif NABİYEV, 2016, “Teoriden Uygulamalara Algoritmalar” Seçkin Yayınevi sayfa 207-216

Raymond W. Yeung, MAY 1991, “IEEE TRANSACTIONS ON INFORMATION THEORY”, VOL. 37, NO. 3

E.T.Jaynes,1957, “Information Theory And Statistical Mechanics”,VOL 106,NO 4

Settings

Guesslong

Start

0

1

Settings==1

T=Randomword()

İ=0;i<=guesslong;i++

İ=0;i<=guesslong;i++

T=Getlong()

1

addword()

T==0

Enter your guess

Return 1;

0

C1=Randomword()

word

Computer guess is C1

That is a right word

1

T==word

Please enter comparenor and comparekar

0

C1=Randomword()

Return 1;

Scorenor

Scorekar

C1

Scorenor==-1 or scorekar==-1

C2

C2=Randomword()

1

Return 1;

0

Cleartxt()

Ek\*¹

|  |
| --- |
| blimp |
| blip | 4 |
| dimly | 3 | 2 |
| flip | 3 | 3 | 2 |
| girl | 2 | 2 | 2 | 2 |
| idly | 2 | 2 | 4 | 2 | 2 |
| limp | 4 | 3 | 3 | 3 | 2 | 2 |
| lip | 3 | 3 | 2 | 3 | 2 | 2 | 3 |
| lucid | 2 | 2 | 3 | 2 | 2 | 3 | 2 | 2 |
| ovoid | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 |
|  | blimp | blip | dimly | flip | girl | idly | limp | lip | lucid | ovoid |

Aşağıdaki şekle bakarsak “ovoid” kelimesini seçilecek en kötü kelimeden birisi olduğunu görebiliriz. Diğer kelimeler ile ortak harf sayısı paterninin en dağınık olduğu “blimp” kelimesi ise en iyi kelimedir. Kullanıcıdan ortak sayının ne geleceğini bilemeyiz. Fakat en kötü ihtimal altında dahi sözlükten 6 kelime elenir. “ovoid” e baktığımızda ise en iyi ihtimal ile 6 kelime eleneceğini görürüz.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 |
| blimp | 0 | 1 | 3 | 3 | 2 | 0 |
| blip | 0 | 1 | 4 | 3 | 1 | 0 |
| dimly | 0 | 0 | 5 | 3 | 1 | 0 |
| flip | 0 | 1 | 4 | 4 | 0 | 0 |
| girl | 0 | 1 | 8 | 0 | 0 | 0 |
| idly | 0 | 0 | 7 | 1 | 1 | 0 |
| limp | 0 | 1 | 3 | 4 | 1 | 0 |
| lip | 0 | 1 | 4 | 4 | 0 | 0 |
| lucid | 0 | 0 | 7 | 2 | 0 | 0 |
| ovoid | 0 | 6 | 3 | 0 | 0 | 0 |

Ek\*²

# Shannon’s Enthropy

Shannon’s Enthropy teorisi olasılık kuramından yararlanarak bir sorunun cevabından edinebileceğimiz bilginin matematiksel olarak formülize edilmiş halidir. Örnek vermemiz gerekirse bir dizi harfin içinden tahmin etme oyununu ele alalım. AAAAAAAA, AAAABBCD, AABBCCDD dizilerinin önce entropilerini sıralayalım. Ardından hangisinin içinden tutulan bir harfi bulma kolaylığı açısından sıralayalım. Oluşturduğumuz tablodan da görebiliriz ki düzensizlik arttıkça bilgiye ulaşma yolumuz uzar. Kısaca Enthropy bilgi edinmek için sorulması gereken sorudur.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Enthropy | Tahmin zorluğu | Enthropy |
| AAAAAAAA | Düşük | Kolay | 0 |
| AAAABBCD | Orta | Orta | 1,75 |
| AABBCCDD | Yüksek | Zor | 2 |

Ek\*³

Ek\*² de görebildiğimiz gibi kalan bütün kelimeler birbirleri ile karşılaştırıldığında sonuçlar eşittir. Bu durumda kullanıcıdan aldığımız karşılaştırma sonuçları kelime elemek için yeterli değildir. Öyleyse bu gibi durumlarda ek bir bilgi daha edinmeliyiz. Her kelime için (alfabedeki yeri)\*(kelimedeki sırası) formülünden yararlanarak kelimeleri diğerlerinden ayırabileceğimiz yeni bir bilgi edinmiş olduk.

|  |  |
| --- | --- |
| Atlas | 168 |
| Salt | 132 |
| Last | 144 |
| Lsat | 127 |
| Tasl | 123 |
| Tals | 129 |
| Satl | 125 |

Fakat bu bilgiyi kullanıcıdan almamız mümkün olmadığı için sezgisel bir yöntem izlemek zorunda kaldık. Birkaç grup anagram kelime için, momentlerin ortalamasına en yakın kelimeyi sor, momenti en büyük olanı sor, momenti en küçük olanı sor gibi sezgisel bir algoritma geliştirmeye çalıştık fakat hiçbir sezgisel algoritmanın başarı oranı %50nin üzerinde olmadığından bir algoritmada karar kılamadık.