# Metin Madenciliği ile Soru Cevaplama Sistemi

*Sevinç İlhan[1], Nevcihan Duru[2], Şenol Karagöz[3], Merve Sağır[4]*

[1]Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
silhan@kocaeli.edu.tr, nduru@kocaeli.edu.tr, senol.karagoz@hotmail.com, mervesagir@gmail.com

## Özet

*Baş döndürücü hızla büyüyen bilgi teknolojileri alanında giderek artan verilerden kullanışlı bilgi elde etmek önemi giderek artan bir konu olarak karşımıza çıkmaktadır. Bilgiyi elde etmede en az maliyetli, en iyi sonucu veren metotlar tercih edilmeye başlanmıştır. Bu bağlamda doğal dil işleme, metin madenciliği için önemli bir disiplin haline gelmiştir.*

*Doğal dil işlemenin çalışma alanlarından biri olan soru cevap sistemlerinin gerek dünyada gerek ülkemizde giderek popülerliği artmaya başlamıştır. Ancak bugün gelinen nokta bile Doğal dil işleme alanında daha çok kat edecek yolun olduğunu göstermektedir. Türkçe'nin sondan eklemeli bir dil olması da dil işlemeyi güçleştirmektedir.*

*Bu çalışma kapsamında, doğal dil işleme ve metin madenciliği teknikleri kullanılarak kullanıcıdan alınan soruya en uygun cevabı içeren metin keşfedilmeye çalışılmıştır. Kullanıcıdan alınan soru, veri madenciliğinin bir aşaması olan ön işlemeden geçirilip anahtar sözcükler belirlenmekte ve her anahtar sözcüğün metin içindeki önemine göre uygun cevap bulunmaya çalışılmaktadır.*

## Abstract

*In Information Technologies area, which are rising increasingly, extracting useful information from the increasing data is became the vital issue. The methods that gives less cost, better performance and the best results are preferred for extracting the information. In this context, natural language processing is became the important discipline for text mining.*

*Question answering systems, which is the one of the areas of natural language processing, are increasingly becoming popular in the world and in our country. However, today there are much things to do in natural language processing. Natural processing becomes hard, because of Turkish is an agglutinative language.*

*In this paper, the most appropriate text to the questioner is trying to be discovered as an answer by using natural language processing and text mining techniques. The question has taken from the user is preprocessed. The key words are become definite and according to the importance degree of key words in the texts; the appropriate answer is to be shearched.*

## 1.Giriş

Veri madenciliği, eldeki verilerden çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılması yaklaşımıdır. Veri madenciliğinin alt dalı olarak ele alınan metin madenciliği ise yazılmış farklı dokümanlardan yeni, önceden bilinmeyen bilgilerin bilgisayar tarafından otomatik bir şekilde keşfedilmesidir. Metin madenciliğini veri madenciliğinden ayıran en büyük fark metin madenciliğinde kalıpların düzgün veritabanlarından çok, doğal dil metinlerinden çıkarılmasıdır.

Metin madenciliğinde, veriden bilgi çıkarma yöntemlerinden biri olan doğal dil işleme disiplini ile bilgi çıkarımında daha anlamlı sonuçlar elde edilmeye başlanmıştır. Doğal Dil İşleme (DDİ), ana işlevi bir doğal dili çözümleme, anlama, yorumlama ve üretme olan bilgisayar sistemlerinin tasarımını konu alan bir mühendislik alanıdır [1]. Doğal dil işleme çalışmaları sayesinde insan-bilgisayar etkileşiminin arttırılması başarılmıştır.

Soru cevaplama sistemleri, bilgiye ulaşma ihtiyacı ile ortaya çıkmış olan ve genelde bilgisayar destekli yapılardır. Soru-cevap benzerliklerini karşılaştırılarak ya da varolan kaynaklar üzerinde yapay zeka gibi insan türevi teknikler uygulanarak, sorulara yeni cevaplar üretmeye çalışan sistemler geliştirilmiştir.

Soru-cevaplama sistemleri incelenirken kendi içinde ikiye ayrılması gerekmektedir. Biçimsel ve anlamsal yönden karşılaştırma yapılarak soru cevap metinleri arasında benzerlik aranmaktadır.

Bu bildiride Türkçe için Doğal Dil İşleme disiplininde biçimsel analiz yöntemine göre kullanıcıdan alınan soru işlenmiştir. Kullanıcıdan alınan soru, vektör uzay modelinde gösterilerek veritabanındaki cevaplar ile karşılaştırılması yapılmış ve kosinüs benzerliği teoremine göre benzerlik oranı hesaplanmıştır.

## 2. Ön İşleme

Veri madenciliğinde analiz edilecek giriş verilerinin belirli bir formata sahip olması ayrıca bozuk veya gereksiz verilerden temizlenmiş olması gerekmektedir. Metin madenciliğinin en büyük sorunu, işleyeceği veri kümesinin yapısal olmamasıdır. Genellikle doğal dil kullanılarak yazılmış dokümanlar üzerinde çalışılan metin madenciliği alanında ön işleme aşaması, veri temizlemenin yanında veriyi uygun formata getirme işlemini de gerçekleştirmektedir [5].

Çalışma kapsamında gerçekleştirilen ön işleme adımlarında; kullanılan soru ve cevap metinleri, sözcüklerine ayrılarak ön işlemenin formatlama aşaması gerçeklenmiştir. Ön işlemenin temizleme aşamasında ise, sözcüklerine ayrılan metinlerdeki noktalama işaretleri, edatlar, bağlaçlar, zamirler ve fiiller gibi Türkçe' de sık kullanılan sözcüklerin çıkarılması işlemleri gerçekleştirilmiştir.

Metnin sözcüklere ayrılma aşamasında Java dilinin StringTokenizer sınıfı kullanılmıştır. Bu sınıf kullanılırken ayıraç olarak noktalama işaretleri alınmıştır, temizleme aşaması için XML dosya yapısında metinden çıkarılması düşünülen sözcükler tutulmuştur. Bu dosyadaki sözcükler ile metindeki her sözcük karşılaştırılarak metin içinde ve XML dosyasında bulunan sözcükler değerlendirmeye alınmamıştır. Fiil tipindeki sözcükler ise Zemberek [2] DDİ kütüphanesinin sağladığı sözcük türü bulma özelliği ile tespit edilmiş ve metinden çıkarılmıştır.

## 3. Vektör Uzay Modeli Kullanımı

Vektör uzay modelinde her nesne, vektör yapısında tanımlanmaktadır. Nesnelerin sahip olduğu farklı özellikler, vektör uzayının eksenlerini oluşturmakta ve her nesne sahip olduğu özelliklere göre vektör uzayında belli bir konuma sahip olmaktadır.

### 3.1 Vektör Oluşturma

Vektör uzay modeli, çalışma içerisinde, doğal dil kullanılarak yazılmış metinlere uygulanmış ve yapısal olmayan bu nesneler yapısal hale getirilmiştir. Üzerinde çalışılan soru ve cevap metinleri, seçilen anahtar sözcükler kullanılarak vektör olarak tanımlanmıştır.

Bir metnin, vektör olarak ifade edilebilmesi için 3 farklı yöntem yer almaktadır. [3].

Yöntemler ile ilgili örnekler tablo 1'deki metinlere göre verilmiştir.

*Tablo 1*: Örnek Metinler

| Id | Metinler |
|---|---|
| 1 | Gribe yakalanan hasta grip olduğunu anlamamıştı. İlacını almamıştı. |
| 2 | İlacını aksatanlar hastalığa davetiye çıkarırlar. |
| 3 | Yıllık enflasyon oranı bu senede yükselişte |
| 4 | Tarımla uğraşanlar bu yıl tarımdan zarar edecekler. |
| 5 | Hakemin gözü önünde olmasına rağmen hakem penaltı çalmadı. (Spor) |
| 6 | Taraftarlara erken gelen gol ilaç gibi geldi ve taraftarlar golden sonra hiç susmadı. (Spor) |

#### 3.1.1. Bitsel Tanımlama

Anahtar sözcük sözlüğünde yer alan sözcüklerin metinde yer alıp almadığı gösteren vektörel bir gösterge oluşturulmaktadır. Yukarıdaki örnek metinler için oluşturulmuş olan sözlük ve metinlerin anahtar kelimelere göre bitsel tanımlamaları aşağıda görülmektedir.

Sözlük={enflasyon, grip, hakem, ilaç, taraftar, tarım}

D1=(0,1,0,1,0,0)
D2=(0,0,0,1,0,0)
D3=(1,0,0,0,0,0)
D4=(0,0,0,0,0,1)
D5=(0,0,1,0,0,0)
D6=(0,0,0,1,1,0)

#### 3.1.2. Frekansa Göre Tanımlama

Sözcüklerin metinlerde kaç defa kullanıldığına dayanan bir yöntemdir. Örnek metinler için oluşturulmuş olan sözlük ve metinlerin anahtar kelimelere göre frekans tanımlamaları aşağıda görülmektedir.

Sözlük={enflasyon, grip, hakem, ilaç, taraftar, tarım}

D1=(0,2,0,1,0,0)
D2=(0,0,0,1,0,0)
D3=(1,0,0,0,0,0)
D4=(0,0,0,0,0,2)
D5=(0,0,2,0,0,0)
D6=(0,0,0,1,2,0)

#### 3.1.1. Tf-Idf Ağırlıklandırma Yöntemine Göre Tanımlama

Tf-Idf ağırlıklandırmasında her bir dokümandaki sözcüklerin frekansı rol oynamaktadır. Böylece dokümanda daha fazla geçen (Tf değeri büyük sözcükler) o doküman için daha değerli olmaktadır. Ayrıca Idf tüm dokümanlarda seyrek geçen sözcükler ile ilgili bir ölçü vermektedir. Bu değer tüm eğitim dokümanları ele alınarak hesaplanmaktadır. Bu yüzden eğer bir sözcük dokümanlarda sık geçiyorsa, o doküman için belirleyici olmadığı düşünülebilir. Eğer sözcük dokümanlarda çok sık geçmiyorsa o sözcüğün o doküman için belirleyici özelliği olduğu kabul edilebilir.

### 3.2 Anahtar Sözcük Seçimi ve Ağırlıklandırma

Dokümanlar arasında kullanımı az olan ve dokümanların ayırt edilebilmesini sağlayacak özellikteki sözcükler, anahtar sözcük olarak nitelendirilmektedirler. Dokümanlar arasında sadece bir dokümanda geçen sözcük veya sözcükler, o doküman için en verimli anahtar sözcükler olarak kabul edilir. Bu anahtar sözcükler kullanılarak yapılan sorgu işlemleri, elemanı oldukları dokümana ulaşmanın en güçlü yolu olacaktır.

Anahtar sözcük seçimi çalışmanın en önemli noktasını oluşturmaktadır. Anahtar sözcük seçimi süreci; ön işleme aşamasındaki anahtar sözcük olamayacak belirli sözcüklerin metinden çıkarılması ile başlamakta, vektör oluşturma aşamasındaki gövde bulma ve aynı gövdeye sahip olan sözcüklerin aynı anahtar sözcük olarak kabul edilmesiyle devam etmektedir.

Ön işleme aşamasında metinden çıkarılan, ayırt edilicilik özelliği olmayan sözcüklerin tespitinde, sözcüklerin Türkçe dil yapısına göre cümlede kullanılma olasılıkları göz önünde bulundurulmuştur. Çıkarılacak sözcük sınıfları olarak edatlar,

bağlaçlar, zamirler ve fiiller seçilmiştir. Edat, bağlaç ve zamir sözcük türlerinin, soru ve cevap metinlerinde sıkça kullanılabileceği ve metinler için ayırt edici olamayacakları düşünülmüştür. Fillerin ise, soru cümlelerinde nadir kullanıldığı; soru cümlelerinin genellikle nedir, kimdir gibi soru sözcükleri ile oluşturulduğu göz önünde bulundurularak temizlenmesine karar verilmiştir. Cevap metninde olup soru metninde olmayan fiiller, cevap vektörünün uzunluğunu arttırmakta ve benzerlik oranını düşürmektedir.

Vektör oluşturma aşamasında ise ön işleme aşamasından sonra elde edilen sözcük listeleri kullanılmaktadır. Bu aşamada sözcükler tekrar işlenmekte ve sözcüklerin gövdelerine ulaşılmaya çalışılmaktadır. Sözcük gövdelerine ulaşabilmek için TÜBİTAK'ın Zemberek [2] isimli DDİ kütüphanesi kullanılmıştır.

Zemberek kütüphanesi ile her sözcüğün gövdeleri elde edilmekte ve bu gövdeler arasında en uzun olanı sözcüğün kullanılacak gerçek gövdesi olarak seçilmektedir.

gözlükçüler → göz-lük-çü-ler

Gözlükçüler sözcüğü, anahtar kelime havuzuna gözlükçü şeklinde eklenmiştir.

Literatürde incelenen çalışmalarda, anahtar sözcük seçiminin statik olarak yapıldığı görülmüştür. Fakat sistem içerisindeki çok sayıda soru ve cevap metinleri için anahtar sözcük havuzunun statik bir şekilde oluşturulması hem zahmetli hem de verimsiz olmaktadır. Bu nedenle dinamik bir yapı oluşturulmaya çalışılmış ve veritabanındaki tüm cevaplar işlenip, cevaplar içerisinde anahtar sözcük özelliği taşıyan sözcükler seçilmiştir. Bu işlem sisteme soru sorulması aşamasında değil cevap ekleme aşamasında gerçekleştirildiği için sistemin soru cevaplama performansını etkilememiştir.

Biçimsel olarak karşılaştırılan soru ve cevap metinlerinin yakınlığının daha hassas bir şekilde incelenebilmesi için kullanılan ağırlık mantığına göre, metinlerde geçen her sözcük aynı değerde değildir. Birkaç cevapta geçen bir sözcük, sadece bir cevapta geçen sözcükten daha az öneme sahiptir. Bu nedenle cevaplara özel sözcüklerin daha büyük ağırlık değeri taşıması gerekmektedir.

$$w_i = tf_i * IDF_i \quad (1)$$

$$IDF_i = \log\left(D / df_i\right) \quad (2)$$

(1) ve (2) formülleri anahtar kelimenin ağırlığının hesaplanmasında kullanılmaktadır.

Ağırlıkların hesaplanmasında sözcüklerin metinler içindeki geçiş durumu ön plandadır. Sözcüğün kaç adet metinde geçtiği bilgisi $\mathbf{df_i}$, toplam metin sayısı **D,** arasındaki ayırt edici özelliğini ortaya koymaktadır. Bu değerin yüksek olması az sayıda metinde geçtiğini göstermektedir ve hesaplamada daha büyük öneme sahip olmasını sağlamaktadır. Az sayıda olursa da birçok metinde geçtiği anlaşılmaktadır. Sıfır çıkması ise bütün metinlerde geçtiğini ve metinleri ayırt etme aşamasında hiç bir etkisinin olmadığını göstermektedir.

Sözcüğün aynı metin içindeki geçiş sayısı $\mathbf{tf_i}$ ise, oluşturulan vektörler arasındaki uzaklığın hesaplanması aşamasında kullanılmaktadır. Metinler arasındaki ayırt edicilik değeri $\mathbf{IDF_i}$, anahtar sözcüğün aynı metin içinde kaç defa geçtiğini gösteren değer ile çarpılarak, sözcüğün metin için toplam ağırlığı ($\mathbf{w_i}$) bulunmaktadır. Bu değer vektörün o sözcük boyutundaki değerini vermektedir. Karesi alınarak vektör uzunluğu değerine ve soru vektöründeki aynı sözcük boyutunun değeri ile çarpılarak da vektörlerin iç çarpımı değerine eklenmektedir.

Bu yöntem ile tablo 2'deki cevap metinlerinde geçen sözcüklerin ağırlıklandırılması gerçekleştirilmekte ve sonuçlar tablo 3'de gösterilmektedir.

*Tablo 2*: Örnek Cevap Metinleri

| Id | Cevaplar |
|---|---|
| C1 | Türkiye' nin başkenti Ankara' dır. |
| C2 | Ankara, Türkiye' nin başkentidir ve İç Anadolu Bölgesi' nde bulunmaktadır. |
| C3 | Türkiye' nin başkenti olan Ankara, İç Anadolu Bölgesi' nin en kalabalık kentidir. |

*Tablo 3*: Sözcük Ağırlıkları

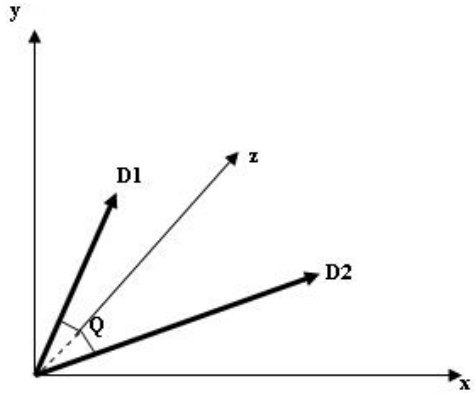
| Sözcük | Ağırlık |
|---|---|
| Bölge | 0.1761 |
| Kent | 0.4771 |
| Ankara | 0 |
| Türkiye | 0 |
| Başkent | 0 |
| O | 0.4771 |
| En | 0.4771 |
| İç | 0.1761 |
| Anadolu | 0.1761 |
| Kalabalık | 0.4771 |
| Bölge | 0.1761 |
| Kent | 0.4771 |
| Ankara | 0 |

## 4. Benzerlik Hesaplanması

Vektör uzayında tanımlanan nesneler belirli konumlara sahiptir. Nesnelerin konumlarını gösteren vektörler kullanılarak, nesnelerin birbirlerine yakınlıkları hesaplanabilmektedir. Vektörler arasındaki yakınlığı hesaplamak için kosinüs benzerliği yöntemi kullanılmıştır. Bu yöntem ile vektörler arasındaki açının, kosinüs değeri hesaplanmakta ve bu değer vektörler arasındaki yakınlık değeri olarak alınmaktadır.

Çalışma kapsamındaki soru ve cevap metinlerinin de vektör uzayına taşınmış olması aralarındaki benzerliğin hesaplanmasında kosinüs benzerliği yönteminin kullanılabilmesini sağlamıştır. Bu yöntem ile karşılaştırılacak

her bir soru ve cevap vektörü arasındaki açının kosinüs değeri hesaplanmaktadır.




*Şekil 1*: Cevap Vektörleri.

$$Cos(D1, D2) = Cos(Q) \quad (3)$$

D1 ve D2 iki vektörsel doküman olmak üzere,

$$Cos(D1, D2) = D1 \bullet D2 \Big/ \|D1\| * \|D2\| \quad (4)$$

Formül 4, vektörler arasındaki açının kosinüs değerini vermektedir. Her vektör ikilisi için bulunan açı değerleri karşılaştırılarak en yakın dokümanlar belirlenmektedir [4].

Formülü 4'ün matematiksel yapısı incelendiğinde, soru ve cevap vektörleri arasındaki iç çarpım değeri ve her bir vektörün uzunluk değerinin, vektörler arasındaki benzerlik sonucunu doğrudan etkilediği anlaşılmaktadır. İç çarpım değeri sonuç ile doğru orantılı, vektörlerin uzunluk değerlerinin çarpımı ise ters orantılıdır. Bu yüzden vektörler arasındaki ortak sözcük sayısının arttırılması, vektör uzunluklarını etkileyen yani iki vektör arasında ortak olmayan sözcüklerin ise azaltılması gerekmektedir.

Benzerlik sonucunun verimli olabilmesi adına anahtar sözcük sözlüğüne cevaplar içerisinde geçen her sözcük alınmamıştır. Her bir cevap için bulunan anahtar sözcükler, anahtar sözcük sözlüğünde toplanmaktadır. Kullanıcıdan alınan sorular bu sözlükteki anahtar sözcüklere göre vektör haline getirilmektedir. Vektör oluşturma aşamasında, aynı köke sahip sözcükler için ortak olan en uzun gövde seçilerek hepsi için tek bir anahtar sözcük veritabanına yazılmaktadır. Bu aşamalardan geçen anahtar sözcük sözlüğü ile daha hassas hesaplamalar yapılmıştır.

Soru: Türkiye' nin başkenti neresidir?

*Tablo 4*: Türkiye'nin Başkenti Neresidir? Metninin Cevapları

| Id | Cevaplar |
|---|---|
| C1 | Türkiye' nin başkenti Ankara' dır. |
| C2 | Ankara, Türkiye' nin başkentidir ve İç Anadolu Bölgesi' nde bulunmaktadır. |
| C3 | Türkiye' nin başkenti olan Ankara, İç Anadolu Bölgesi' nin en kalabalık kentidir. |

Cos(S,C1) = S•C1 / ||S|| * ||C1||
= ((1*0) * (1*0) + ((1*0) * (1*0)) / ((1*0) + (1*0)) ((1*0) + (1*0) + (1*0))
= 0 / ( 0 * 0)
= 0

Cos(S,C2) = S•C2 / ||S|| * ||C2||
= ((1*0) * (1*0) + ((1*0) * (1*0)) / ((1*0) + (1*0)) * ((1*0) + (1*0) + (1*0) + (1*0.1761) + (1*0.1761) + (1*0.1761))
= 0 / ( 0 * 0. 0.305)
= 0

Cos(S,C3) = S•C1 / ||S|| * ||C1||
= ((1*0)*(1*0) + ((1*0) * (1*0)) / ((1*0) + (1*0)) * ((1*0) + (1*0) + (1*0) + (1*0.1761) + (1*0.1761) + (1*0.1761) + (1*0.4771) + (1*0.4771) + (1*0.4771))
= 0 / ( 0 * 1.1096)
= 0

Yukarıdaki hesaplama adımlarında vektörler arasındaki açının kosinüs değeri hesaplanmıştır. Vektörleri oluşturan sözcüklerin frekans ve ağırlıkları da hesaplamaya katılmıştır. Fakat Türkiye, başkent ve Ankara sözcüklerinin ağırlıkları sıfır olduğundan sonuç sıfır bulunmuştur. Bu sözcüklerin ağırlıklarının sıfır oluşu bir önceki ağırlıklandırma bölümünde anlatılmıştır. Türkiye, başkent ve Ankara sözcükleri 3 cevapta da kullanıldıkları için ağırlık hesabında sıfır değerini almışlardır.

Bu durumun düzeltilebilmesi için veri tabanına yeni bir cevap daha eklenmiştir. Yeni cevap tablo 5'de görülmektedir.

*Tablo 5*: Türkiye'nin Başkenti Neresidir? Metninin Yeni Cevapları

| Id | Cevaplar |
|---|---|
| C1 | Türkiye' nin başkenti Ankara' dır. |
| C2 | Ankara, Türkiye' nin başkentidir ve İç Anadolu Bölgesi' nde bulunmaktadır. |
| C3 | Türkiye' nin başkenti olan Ankara, İç Anadolu Bölgesi' nin en kalabalık kentidir. |
| C4 | İstanbul nüfusu en yüksek kentimizdir. |

Son eklenen cevaptan (C4) sonra sözcüklerin ağırlıkları değişmiştir. Sözcük ağırlıklarının son hali tablo 6'da verilmiştir.

*Tablo 6*: Sözcük Ağırlıkları

| Sözcük | Ağırlık |
|---|---|
| Bölge | 0.301 |
| Kent | 0.301 |
| Ankara | 0.1249 |
| Türkiye | 0.1249 |
| Başkent | 0.1249 |
| Olan | 0.6021 |
| En | 0.301 |
| İç | 0.301 |
| Anadolu | 0.301 |

| | |
|---|---|
| Kalabalık | 0.6021 |
| istanbul | 0.6021 |
| nüfus | 0.6021 |
| yüksek | 0.6021 |

Yeniden hesaplanan ağırlık değerlerinden sonra benzerlik hesaplaması aşağıdaki sonuçları vermektedir.

S•C1 / ||S|| * ||C1|| = ((1*0.1249) * (1*0.1249) + ((1*0.1249) * (1*0.1249)) / ((1*0.1249) + (1*0.1249)) * ((1*0.1249) + (1*0.1249) + (1*0.1249))
= 0.0312 / ( 0.1766 * 0.2163 )
= 0.8168

S•C2 / ||S|| * ||C2|| = ((1*0.1249) * (1*0.1249) + ((1*0.1249) * (1*0.1249)) / ((1*0.1249) + (1*0.1249)) * ((1*0.1249) + (1*0.1249) + (1*0.1249) + (1*0.301) + (1*0.301) + (1*0.301))
= 0.0312 / ( 0.1766 * 0.5644 )
= 0.313

S•C3 / ||S|| * ||C3|| = ((1*0.1249) * (1*0.1249) + ((1*0.1249) * (1*0.1249)) / ((1*0.1249) + (1*0.1249)) * ((1*0.1249) + (1*0.1249) + (1*0.1249) + (1*0.301) + (1*0.301) + (1*0.301) + (1*0.301) + (1*0.6021) + (1*0.301))
= 0.0312 / ( 0.1766 * 1.1067 )
= 0.1596

S•C1 / ||S|| * ||C1|| = 0.0 / ((1*0.1249) + (1*0.1249)) * ((1*0.6021) + (1*0.6021) + (1*0.301) + (1*0.6021) + (1*0.301))
= 0.0 / ( 0.1766 * 1.1264 )
= 0.0

Yeni hesaplamalar sonucunda "Türkiye'nin başkenti neresidir?" sorusuna en yakın cevap olarak "Türkiye'nin başkenti Ankara'dır" şeklindeki 1. cevap bulunmuştur.

Soru ile cevap metinleri arasındaki ortak sözcük sayısının ilk 3 cevapta aynı olmasına karşılık, vektör uzunluklarının farklı olması sonucu değiştirmiştir. Son cevap da ise ortak hiçbir sözcük olmadığından sıfır sonucu dönmüştür.

Yukarıdaki hesaplamalar göz önünde bulundurulduğunda, bir soruya verilen doğru cevaplar arasında detayı en az olan cevap doğru cevap olarak getirilecektir.

## 5.Sonuçlar

Bu makalede, Doğal Dil İşleme disiplini altında yer alan biçimsel analiz yöntemine göre kullanıcıdan alınan soru metni işlenmektedir. Kullanıcıdan alınan soru metni için anahtar sözcükler belirlenip bu sözcüklerin her metin için ağırlığı belirlenmektedir. Bu ağırlık vektör uzay modelinde gösterilmektedir. Vektör uzay modeli bilgi çıkarımı, bilgi filtreleme, indeksleme gibi alanlarda kullanılan cebirsel bir modeldir. Doğal dil belgelerinin çok boyutlu uzayda özel bir anlamını simgelemektedir [3].

Cevap metinleri veritabanında vektörsel halde tutulmaktadır. Bu sayede hesaplamada sadece bu vektörsel yapılar katılmaktadır. Fakat soru metinleri kullanıcıdan çalışma anında alınan veriler olduğundan, her soru hesaplamaya katılmadan önce vektör olarak tanımlanmaktadır. Cevap metinlerinin veritabanında vektörler halinde tutulması ve sorulan sorunun bir kez vektöre çevrilip her cevapla karşılaştırmada bu vektörün kullanılması performans açısından önemlidir.

Sistem içerisinde kullanıcıdan alınan her soru, vektör haline getirildikten sonra tüm cevaplar ile karşılaştırılmaktadır. Sistemde anahtar sözcük seçim işlemi dinamik olarak yapılmakta ve veritabanına yazılmaktadır. Yapılan her sorguda, veritabanında hazır bulunan anahtar sözcükler ile vektör uzayında gösterilen sorgu karşılaştırılmaktadır. Veritabanında anahtar sözcüklerin hazır olarak bulundurulması performansı artırmaktadır.

Bu çalışma, ileride yapılacak soru cevaplama sistemleri için dokümanlar arası benzerlik arama gibi çalışmalara bir kaynak niteliği taşımaktadır. Biçimsel analiz yönünden kullanılabilecek bu sistem, anlamsal yönden desteklendiğinde daha doğru sonuçların çıkarılacağı düşünülmektedir.

### 6.Kaynaklar


[1] Rich E. Artificial Intelligence, McGraw Hill Inc., Second Edition, Newyork, 1991.

[2]https://zemberek.dev.java.net/surumler/v04/zemberek_0.4.0.html

[3] Pilavcılar İ.F., "Metin Madenciliği ile Metin Sınıflandırma", Yıldız Teknik Üniv. FBE, Yüksek Lisans Tezi, 2007.

[4] http://www.miislita.com/term-vector/term-vector-3.html

[5] Feldman, R., Sanger, J., 2007. The Text Mining HandBook Advanced Approaches in Advanced Approaches in Analyzing Unstructured Data.