Web Madenciliği
(Web Mining)

Hazırlayan: M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Bing Liu, Springer, 2011." kitabı kullanılarak hazırlanmıştır.

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

Giriş

- Web aramanın temeli, bilgi erişim (information retrieval - IR) yöntemlerine dayanmaktadır.
- **Klasik IR sistemleri, temel bilgi birimini doküman olarak alır, Web üzerinde ise temel bilgi birimi Web sayfalarıdır.**
- Bilgiye erişim, kullanıcı sorgusuyla ilgili bir grup dokümanın bulunmasını ifade eder.
- **En yaygın kullanılan sorgu formatı keyword (term) listesi şeklindedir.**
- IR ile bilgiye erişim, veritabanlarında SQL ile bilgiye erişimden çok farklıdır.
- **Veritabanları yapılandırılmıştır ancak metin içerisindeki bilgi yapılandırılmamıştır.**

3

Giriş

- Web arama, en önemli IR uygulamalarının başında yer alır.
- **Web arama**, IR yöntemlerinin yanı sıra çok sayıda **kendisine özgü yöntemi de birlikte kullanır.**
- **Web aramada etkinlik ve yüksek performans en önemli gereksinimlerdir,** çünkü Web üzerindeki doküman sayısı çok fazladır.
- Ancak, klasik IR sistemlerinde doküman sayısı daha az olduğundan etkinlik en önemli gereksinim değildir.
- **Web kullanıcıları arama sonuçlarına etkin ve çok hızlı yanıt almak istemektedir.**

4

Giriş

- **Web sayfaları klasik metin dokümanlarına göre çok farklı yapıya sahiptir.**
- Web sayfaları hyperlinklere ve anchor metinlere sahiptir, klasik dokümanlarda linkler yer almaz.
- **Hyperlinkler**, Web arama algoritmalarında ve elde edilen sonuçların sıralanması için kullanılan **ranking algoritmalarında kullanılan en önemli bileşenlerdir.**
- Hyperlinkler ile ilişkilendirilen anchor metinler, link verilen sayfaya ait bilgiyi içerdiğinden çok önemlidir.
- **Web sayfaları yarı yapılandırılmıştır.** Bir Web sayfasında title, metadata, body gibi alanlar bulunmaktadır.
- **Bazı alanlardaki bilgiler diğerlerinden daha fazla öneme sahiptir.**

5

Giriş

- **Web sayfaları çok sayıda farklı yapıya sahip bloklardan oluşur.**
- Bu bloklardan **bazıları çok önemlidir** (menü alanları, başlık bilgileri), **bazıları ise önemli değildir** (reklamlar, copyright bilgileri, gizlilik bildirimleri) ve Web sayfasından çıkartılmalıdır.
- Bu blokların çok iyi analiz edilmesi ve faydalı olanların seçilmesi, faydalı olmayanların silinmesi gereklidir.
- **Web dokümanlarında spamming çok önemlidir.** Ancak **klasik IR dokümanlarında spamming yapılmaz.**
- Spamming ile Web sayfasının arama sonuçlarında elde edilen listede daha üst sıralarda yer alınması sağlanabilir.
- Bunun sonucunda kullanıcı sorgusuyla daha ilgili Web sayfaları alt sıralarda olduğundan kullanıcı tarafından görülemez.

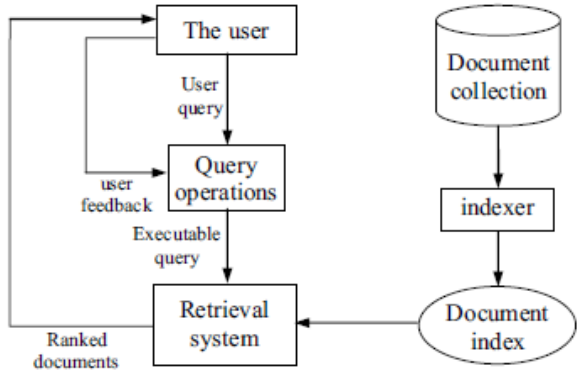
6

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

Bilgi Erişiminde Temel Yaklaşımlar

- IR ile kullanıcının ihtiyaç duyduğu bilgilerle uyumlu bilgilerin bulunması amaçlanır.
- IR, **bilginin toplanması, organize edilmesi, depolanması, geri kazanımı ve dağıtılması** konularıyla ilgilenir.



Bilgi Erişiminde Temel Yaklaşımlar

- Kullanıcı sorguları, **keyword sorguları**, **Boolean sorguları**, **phrase sorguları**, **proximity sorguları**, **full doküman sorguları** ve **doğal dil sorguları** şeklinde olabilir.

Keyword sorguları

- **Kullanıcı ihtiyaç duyduğu bilgiyi keyword listesi olarak verir.**
- Keyword listesindeki **tüm terimlerin birbirine AND ile bağlandığı varsayılır.**
- Elde edilen listedeki tüm dokümanların keyword listesindeki tüm terimleri bulundurması zorunlu olmayabilir.

9

Bilgi Erişiminde Temel Yaklaşımlar

Boolean sorguları

- Kullanıcı Boolean ifadeleri ile karmaşık sorgular oluşturabilir.
- Sorgu içerisinde **AND, OR** ve **NOT** gibi Boolean operatörler yer alabilir.
- Arama motorları Boolean sorguların kısıtlı versiyonunu kullanır.

Phrase sorgular

- Bu sorgular sıralı kelimelerden oluşur ve bir ifadeyi gösterir.
- Elde edilen dokümanlar en az bir kez sorgunun tamamını içerisinde bulundurmak zorundadır.
- Arama motorlarında **phrase sorguları çift tırnak içerisinde yer alır.**

10

Bilgi Erişiminde Temel Yaklaşımlar

Proximity sorgular

- Phrase sorgular ve keyword'lerden oluşabilir.
- **Proximity sorgulardaki terimlerin doküman içerisinde birbirine uzaklıkları ranking algoritmalarında kullanılır.**
- Tüm terimleri ve phrase'leri bulunduran dokümanlardan bu **terimler birbirine daha yakın olanlar daha üst sırada yer alır.**
- Bazı sistemler terimler arasındaki **maksimum mesafeyi** de **kısıtlayabilmektedir.**
- Popüler arama motorlarının büyük çoğunluğu terimlerin yakınlığını ve sırasını değerlendirir.

11

Bilgi Erişiminde Temel Yaklaşımlar

Full doküman sorguları

- Bu tür sorgular verilen bir dokümanın benzeri olan dokümanları bulmayı amaçlar.
- **Sorgu sayfasında dokümana ait URL girilir.**

Doğal dil sorguları

- **En karmaşık ancak ideal sorgu türüdür.**
- Kullanıcı isteğini doğal dil ifadesiyle verir.
- Ancak, doğal dili anlamak halen oldukça zordur tam olarak başarı sağlanamamıştır.

12

Bilgi Erişiminde Temel Yaklaşımlar

Sorgu önişlemleri

- Sorgu içerisindeki anlamı **önemli olmayan kısımlar** (the, a, in, ki, de/da, için, ...) **çıkartılır.**
- Daha önce yapılan sorgularda kullanıcıdan alınan **ilgililik geribildirimi** (relevance feedback) gözönüne alınarak **orijinal sorgular yeniden düzenlenir.**
- Indexer modülü ile orijinal dokümanlar hızlı erişimi sağlamak için indekslenir.
- **Informaton Retrieval Sisteminin en önemli görevi** kullanıcı sorgusuyla ilgili olduğu belirlenen **dokümanların sıralanmasıdır.**

13

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

14

Bilgi Erişim Modelleri

- IR modeli, **dokümanların ve sorguların nasıl gösterileceği** ve **dokümanlar ile kullanıcı sorguları arasındaki ilginin nasıl tanımlanacağını** belirler.
- Temel olarak 4 tane IR modeli bulunmaktadır:
  - Boolean modeli
  - Vector space modeli
  - İstatistiksel dil modeli
  - Probabilistic model
- En yaygın kullanılan IR modelleri, Boolean modeli ve vector space modelidir.
- IR modelleri, dokümanları ve sorguları farklı gösterse de, **dokümanlar ve sorgular kelimelerden oluşan küme olarak alınır.**

15

Bilgi Erişim Modelleri

- **IR modellerinde kelimelerin sırası ve cümle içerisindeki konumu önemli değildir.**
- Doküman veya sorgu içerisindeki **tüm kelimeler hesaplanan bir ağırlık değeri ile ilişkilidir.**
- Bir doküman topluluğu D olsun.
- Farklı terimlerden oluşan $V = \{t_1, t_2, ..., t_{|V|}\}$ kümesi bu doküman topluluğunun **sözlüğü (vocabulary)** olarak ifade edilir.
- Burada, $|V|$ sözlüğün boyutudur.
- Bir $t_i \in V$ **kelimesi ile** $d_j \in D$ **dokümanı arasındaki ilişki** $w_{ij} \geq 0$ **ağırlığı ile gösterilir.**

16

Bilgi Erişim Modelleri

- Bir t_i kelimesi $\mathbf{d}_j$ dokümanında bulunmuyorsa w_{ij} = 0 olur.
- Her doküman $\mathbf{d}_j$ aşağıdaki gibi gösterilir.

$$\mathbf{d}_j = (w_{1j}, w_{2j}, \ldots, w_{|V|j}),$$

burada, **her w_{ij} ağırlığı, $t_i \in V$ ile ilişkilidir** ve **t_i kelimesinin $\mathbf{d}_j$ dokümanı için önem seviyesini gösterir** (kelimelerin sırası önemli değildir).

- Doküman topluluğu ilişkisel tablo veya matris şeklinde ifade edilebilir.
- Bu tabloda her kelime (terim) bir niteliği (attribute) ve her ağırlık nitelik değerini (value) gösterir.
- **Farklı IR modelleri w_{ij} ağırlık değerini farklı şekillerde hesaplar.**

17

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

18

Boolean Model

- En eski IR modellerinden birisidir.
- Doküman ve sorgu için **Boolean cebri kullanılır.**
- Her doküman $\mathbf{d}_j$ aşağıdaki gibi gösterilir.

Doküman gösterimi

- **Her kelime için** dokümanda ve sorguda bulunup bulunmadığına bakılır ve $w_{ij} \in \{0, 1\}$ olur.
- Eğer, t_i kelimesi $\mathbf{d}_j$ dokümanı içinde varsa $w_{ij} = 1$, yoksa $w_{ij} = 0$ olur.

$$w_{ij} = \begin{cases} 1 & \text{if } t_i \text{ appears in } \mathbf{d}_j \\ 0 & \text{otherwise.} \end{cases}$$

19

Boolean Model

Boolean sorgular

- Sorgularda **AND**, **OR** ve **NOT** Boolean operatörleri kullanılır.
- Boolean sorgular, ((x **AND** y) **AND** (**NOT** z)) şeklinde kesin anlama sahiptirler (x, y, z terimlerdir).

Doküman erişimi

- Verilen Boolean sorgu için elde edilen dokümanlar sorguyu mantıksal olarak doğru yapan dokümanlardır.
- **Bir doküman ya tam ilgilidir ya da tam ilgisizdir (exact match).**
- Çoğu arama motoru sınırlı Boolean operatör kullanır (+ inclusion, – exclusion).
- **Örnek:** mining –data + “equipment price”

20

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

21

Vector Space Model

- **En yaygın kullanılan IR modelidir.**

Doküman gösterimi

- Vector space modelinde **dokümanlar ağırlık vektörü olarak gösterilir.**
- Ağırlık değerleri **TF (Term Frequency)** ve **TF-IDF (Term Frequency–Inverse Document Frequency)** yöntemleriyle elde edilir.
- $\mathbf{d}_j$ dokümanı içindeki t_i kelimesinin w_{ij} ağırlık değeri hesaplanma yöntemine göre herhangi bir değer olabilir.

22

Vector Space Model

Term Frequency (TF)

- $\mathbf{d}_j$ dokümanı içindeki t_i kelimesinin w_{ij} ağırlık değeri, t_i kelimesinin bulunma sayısıdır ve f_{ij} olarak gösterilir.
- Normalizasyon uygulanmış şekli aşağıdadır:

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$$

- TF yönteminde **dokümanların büyük çoğunluğunda bulunan kelimelerin ayırt edici özelliğinin olmayışı gözönüne alınmaz.**

23

Vector Space Model

Term Frequency–Inverse Document Frequency (TF–IDF)

- Bir doküman topluluğundaki toplam doküman sayısı N olsun.
- f_{ij} ise t_i kelimesinin $\mathbf{d}_j$ dokümanında bulunma sayısı olsun.
- $\mathbf{d}_j$ dokümanında t_i kelimesinin normalize edilmiş TF değeri,

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$$

- Maksimum değer, tüm terimlerin $\mathbf{d}_j$ dokümanında en çok bulunan kelimenin adedine eşittir. $|V|$ sözlük boyutudur.
- Eğer bir t_i kelimesi $\mathbf{d}_j$ dokümanında yoksa $tf_{ij} = 0$ olur.

24

Vector Space Model

Term Frequency–Inverse Document Frequency (TF–IDF)

- $\boldsymbol{df_i}$ ise t_i kelimesinin en az bir kez bulunduğu doküman sayısı olsun.
- t_i kelimesinin inverse document frequency (IDF) değeri,

$$idf_i = \log \frac{N}{df_i}$$

- Bir t_i kelimesi çok sayıda dokümanda varsa ayırt edicilik özelliği ve önemi olmaz.
- TF-IDF ağırlık değeri aşağıdaki gibi hesaplanır.

$$w_{ij} = tf_{ij} \times idf_i$$

25

Vector Space Model

Sorgular

- Bir **q** sorgusu doküman topluluğundaki bir doküman ile aynı şekilde gösterilir.
- **q** sorgusu içindeki her t_i kelimesinin w_{iq} ağırlık değeri dokümanlardaki gibi hesaplanır.
- Salton ve Buckley tarafından w_{iq} ağırlık değeri hesabı için aşağıdaki eşitlik önerilmiştir.

$$w_{iq} = \left(0.5 + \frac{0.5 f_{iq}}{\max\{f_{1q}, f_{2q}, \ldots, f_{|V|q}\}} \right) \times \log \frac{N}{df_i}$$

26

Vector Space Model

Doküman erişimi ve ilgililik sıralaması

- Bir dokümanın bir sorguyla ilgili olup olmadığına karar vermek çok zordur.
- Dokümanlar **sorguya ilgililik derecelerine göre sıralanır.**
- **İlgililik derecesi, $\mathbf{q}$ sorgusu ile $\mathbf{d}_j$ dokümanının benzerliğini hesaplayarak elde edilir.**
- **Benzerliğin hesaplanması için temel yaklaşımda sorgu ile dokümanda bulunan ortak kelimelerin ağırlık değerleridir.**
- Bazı ranking algoritmaları kelimelerin doküman içerisindeki **yakınlıklarını da gözönüne alarak benzerlik** hesaplayabilmektedir.

27

Vector Space Model

Doküman erişimi ve ilgililik sıralaması

- Metin ve doküman kümelemede en yaygın kullanılan benzerlik **cosine similarity**'dir.

$$cosine(\mathbf{d}_j, \mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q} \rangle}{\| \mathbf{d}_j \| \times \| \mathbf{q} \|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}}$$

- w_{ij} , t_i kelimesinin $\mathbf{d}_j$ dokümanındaki ağırlık değeridir.
- w_{iq} , t_i kelimesinin $\mathbf{q}$ sorgusundaki ağırlık değeridir.
- Daha basit benzerlik ölçütü iki vektör ile hesaplanabilir.

$$sim(\mathbf{d}_j, \mathbf{q}) = \langle \mathbf{d}_j \bullet \mathbf{q} \rangle$$

28

Vector Space Model

Doküman erişimi ve ilgililik sıralaması

- **Okapi** metin ve doküman kümelemede popüler yöntemlerdendir. Genellikle kısa sorgularda daha etkindir.

$$okapi(d_j, q) = \sum_{t_i \in q, d_j} \ln \frac{N - df_i + 0.5}{df_i + 0.5} \times \frac{(k_1 + 1) f_{ij}}{k_1 (1 - b + b \frac{dl_j}{avdl}) + f_{ij}} \times \frac{(k_2 + 1) f_{iq}}{k_2 + f_{iq}}$$

Bulunma sıklığı arttıkça önemi artar.

Bulunduğu doküman sayısı arttıkça önemi azalır.

Doküman boyutu arttıkça sık geçmesinin önemi azalır.

f_{ij} , t_i kelimesinin $\mathbf{d}_j$ dokümanındaki bulunma sayısıdır.

f_{iq} , t_i kelimesinin $\mathbf{q}$ sorgusundaki bulunma sayısıdır.

df_i , t_i kelimesini bulunduran doküman sayısıdır.

dl_j , $\mathbf{d}_j$ dokümanının uzunluğudur.

$avdl$, dokümanların ortalama uzunluğudur.

$k_1 = [1{,}0 – 2{,}0]$
$b = 0{,}75$ (genellikle)
$k_2 = [1 – 1000]$

29

Vector Space Model

Doküman erişimi ve ilgililik sıralaması

- **Pivoted normalization weighting** yöntemi ile metin ve dokümanlara skor değeri hesaplanabilir.

$$pnw(d_j, q) = \sum_{t_i \in q, d_j} \frac{1 + \ln(1 + \ln(f_{ij}))}{(1 - s) + s \frac{dl_j}{avdl}} \times f_{iq} \times \ln \frac{N + 1}{df_i}$$

Bulunma sıklığı arttıkça önemi artar.

Bulunduğu doküman sayısı arttıkça önemi azalır.

Doküman boyutu arttıkça sık geçmesinin önemi azalır.

f_{ij} , t_i kelimesinin $\mathbf{d}_j$ dokümanındaki bulunma sayısıdır

f_{iq} , t_i kelimesinin $\mathbf{q}$ sorgusundaki bulunma sayısıdır

df_i , t_i kelimesini bulunduran doküman sayısıdır

dl_j , $\mathbf{d}_j$ dokümanının uzunluğudur

$avdl$, dokümanların ortalama uzunluğudur

$s = 0{,}2$ (genellikle)

30

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

31

İlgililik Geribildirimi

- Bilgi erişiminin etkinliğini artırmak için farklı yöntemler önerilmiştir.
- **Kullanıcı genellikle kısa ve basit sorgular oluşturur**.
- Arama motoru bir doküman kümesi döndürür.
- Kullanıcı bazı dokümanları ilgili, bazı dokümanları ise ilgisiz olarak işaretler (doğrudan veya dolaylı olarak).
- **Arama motoru yeni sorguya ait ağırlık vektörünü belirler.**
- Yeni sorguya ait sonuç listesini döndürür.
- **Yeni** sorgunun **recall değerinin daha iyi olması beklenir.**
- **Sorgu iyileştirme işlemleri, sunulan sonuç listesinden kullanıcı memnun oluncaya kadar devam edebilir.**

32

İlgililik Geribildirimi

- **İlgililik geribildirimi (relevance feedback),** kullanıcının ilgili ve ilgili olmayan dokümanları belirlemesini ve sözlükten yeni kelimeler ekleyerek (w_{iq} değeri 0 olanların yeni ağırlık değeri $w_{iq} > 0$) sorguyu genişletmesini (iyileştirmesini) sağlar.
- Yeni oluşturulan sorgu ile doküman listesi yeniden elde edilir.
- İlgililik geribildirimi yönteminde kullanıcı elde edilen listeden memnun oluncaya kadar tekrar yapılabilir.
- Yaygın kullanılan ilgililik geribildirimi yöntemleri:
  - **Rocchio yöntemi**
  - **Makine öğrenmesi yöntemi**

33

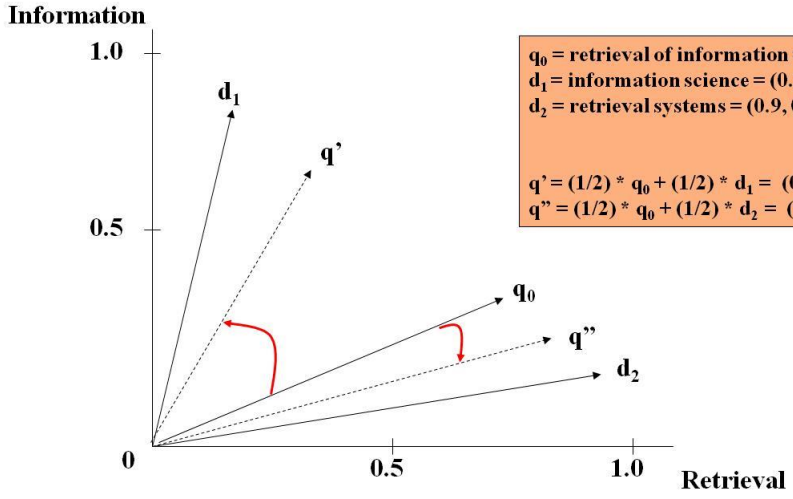
Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

34

Rocchio yöntemi

Vektör gösterimi



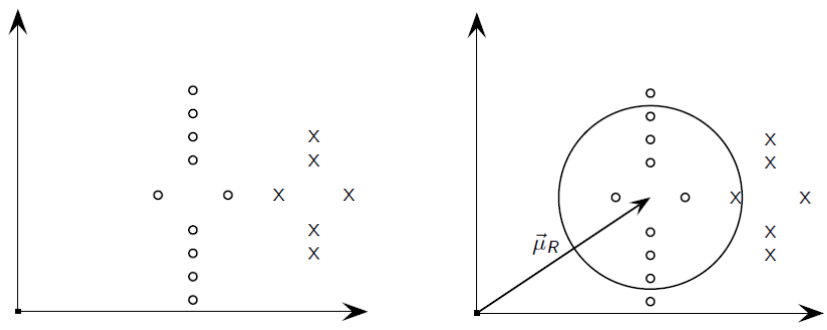
q_0 = retrieval of information = (0.7, 0.3)
d_1 = information science = (0.2, 0.8)
d_2 = retrieval systems = (0.9, 0.1)

q' = (1/2) * q_0 + (1/2) * d_1 = (0.45, 0.55)
q" = (1/2) * q_0 + (1/2) * d_2 = (0.80, 0.20)

35

Rocchio yöntemi

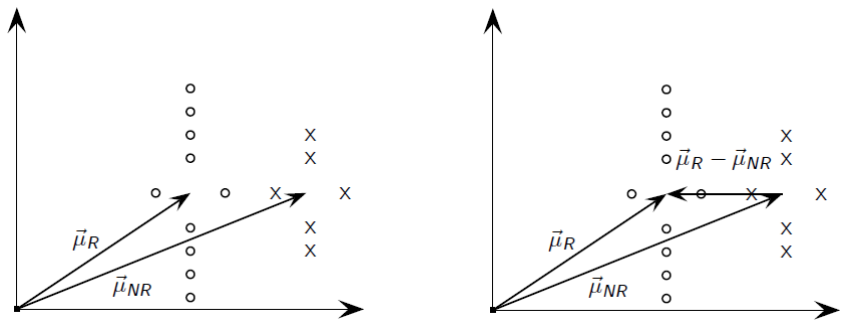
- Şekilde, “o” ilgili dokümanlar, “x” ilgili olmayan dokümanlardır.



36

Rocchio yöntemi

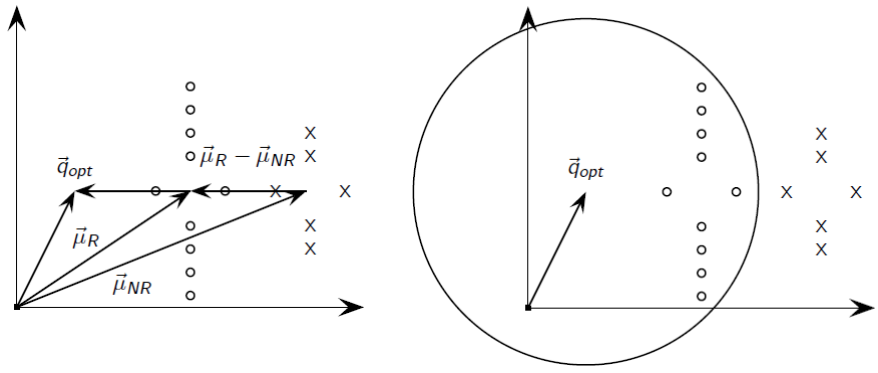
- Şekilde kullanıcının ilgili ve ilgisiz olarak işaretlediği dokümanlara ait merkez nokta vektörleri ile fark vektörleri görülmektedir.



37

Rocchio yöntemi

- İyileştirilmiş sorgu için vektörü ile kümelerin ayrımı görülmektedir.
- Yeni sorgunun ağırlık vektörü ilgili olmayan dokümanlardan uzaklaşmaktadır.



38

Rocchio yöntemi

- **İlk oluşturulan listede kullanıcının belirlediği ilgili ve ilgili olmayan dokümanları kullanarak sorguyu genişletir.**
- Yeni sorgu tekrar kullanılarak doküman listesi elde edilir.

$$\mathbf{q}_e = \alpha\mathbf{q} + \frac{\beta}{|D_r|}\sum_{\mathbf{d}_r \in D_r}\mathbf{d}_r - \frac{\gamma}{|D_{ir}|}\sum_{\mathbf{d}_{ir} \in D_{ir}}\mathbf{d}_{ir}$$

$\mathbf{q}_e$, genişletilmiş sorgu vektörü (ağırlık değerleri)

$\mathbf{q}$, orijinal sorgu vektörü (ağırlık değerleri)

D_r , ilgili dokümanlar kümesi

D_{ir} , ilgili olmayan dokümanlar kümesi

α, β, γ, katsayılar (genellikle $\alpha = 1$, $\beta = 0.75$, $\gamma = 0.25$)

Negatif geribildirim alan dokümanlar

Pozitif geribildirim alan dokümanlar

39

Rocchio yöntemi

- **Pozitif geribildirimler negatif geribildirimlerden daha önemlidir ($\beta = 0.75$, $\gamma = 0.25$).**
- Bazı sistemler sadece pozitif geribildirimlere izin verir.

Negatif yönleri

- Kullanıcı sorgusu ile sözlükteki kelimeler arasında uyumsuzluk olabilir **(Kullanıcı sözlükte olmayan kelime kullanabilir).**
- Kullanıcının işaretlediği dokümanlar ilgili olmayabilir.
- Kullanıcı geribildirim için isteksiz olabilir.
- Sorgu oluşturma maliyeti yüksektir ve değiştirilmiş birden fazla sorgu oluşturabilir.
- Daha uzun sorgular oluşabilir ve sorgu işleme süresi artar.

40

Rocchio yöntemi

Örnek

- Sözlük 9 kelimeden oluşmaktadır.

$$d_{r1} = (0.030, 0.00, 0.00, 0.025, 0.025, 0.050, 0.00, 0.00, 0.120)$$

$$d_{r2} = (0.020, 0.009, 0.020, 0.002, 0.050, 0.025, 0.100, 0.100, 0.120)$$

$$d_{ir1} = (0.030, 0.010, 0.020, 0.00, 0.005, 0.025, 0.00, 0.020, 0.00)$$

$$q = (0.00, 0.00, 0.00, 0.00, 0.500, 0.00, 0.450, 0.00, 0.950)$$

$$\alpha = 1$$

$$\beta = 0.75$$

$$\gamma = 0.25$$

$$q_{yeni} = \alpha \times q + \left(\frac{\beta}{2} \times (d_{r1} + d_{r2})\right) - \left(\frac{\gamma}{1} \times d_{ir1}\right)$$

$$q_{yeni} = (0.011, 0.000875, 0.002, 0.01, 0.527, 0.022, 0.488, 0.033, 1.04)$$

Pozitif geribildirim alan dokümanlar

Negatif geribildirim alan doküman

Orijinal sorgu

İyileştirilmiş sorgu

41

Rocchio yöntemi

Örnek

- Sözlük 6 kelimeden oluşmaktadır.

new query vector = α · original query vector +
β · relevant document vectors -
γ · non-relevant document vectors

0	4	0	8	0	0	α = 1		0	4	0	8	0	0
2	4	8	0	0	2	β = 0.5	+	1	2	4	0	0	1
8	0	4	4	0	16	γ = 0.25	-	2	0	1	1	0	4
								-1	6	3	7	0	-3
								0	6	3	7	0	0

Typically β > γ, since positive feedback is more meaningful.

Negative term weights become 0.

Yeni eklenen kelime

42

Rocchio yöntemi

Örnek

- Sözlük 5 kelimeden oluşmaktadır.

- vocabulary: {*run, lion, cat, dog, program*}
  - original query: q_0 = [1,0,1,0,0] = [1*run, 1*cat]
  - relevant document: d_R=[2,2,1,0,0] = [2*run, 2*lion, 1*cat]
  - non-relevant doc: d_N=[2,0,1,0,3] = [2*run, 1*cat, 3*program]

- $$\vec{q}_1 = \vec{q}_0 + \frac{\alpha}{|R|}\sum_{d\in R}\vec{d} - \frac{\beta}{|NR|}\sum_{d\in NR}\vec{d}$$ α=1.0, β=0.5

- $q_1 = q0 + 1.0\ d_R - 0.5\ d_N$
  = [1,0,1,0,0] + 1.0 [2,2,1,0,0] - 0.5 [2,0,1,0,3]
  = [2, 2,1.5, 0, -1.5]
  = [2*run, 2*lion, 1.5*cat]

43

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

44

Makine öğrenmesi yöntemi

- **Kullanıcı tarafından işaretlenen ilgili ve ilgili olmayan dokümanlar kullanılarak bir sınıflandırma modeli oluşturulabilir.**
- Böylelikle, relevance feedback problemi **öğrenme problemi** şekline dönüştürülür.
- Bu aşamadan sonra herhangi bir öğrenme metodu kullanılabilir.
- **Öğrenme problemi** olarak ifade edildiğinde, **orijinal sorgu ile benzerlik karşılaştırması yapmaya gerek kalmaz.**
- **Rocchio metodu sınıflandırıcı olarak kullanılabilir.**

45

Makine öğrenmesi yöntemi

- Rocchio sınıflandırıcı oluşturmak için her sınıf için (ilgili ve ilgili olmayan dokümanlar) bir protoip $\mathbf{c}_i$ sınıfı aşağıdaki gibi oluşturulur.
- Her test dokümanının $\mathbf{d}_t$ her bir prototip $\mathbf{c}_i$ sınıfı ile benzerliği hesaplanır (cosine similarity veya başka yöntem kullanılabilir).
- **Test dokümanı hangi sınıfa daha çok benzer ise o sınıfa atanır.**

$$\mathbf{c}_i = \frac{\alpha}{|D_i|}\sum_{\mathbf{d}\in D_i}\frac{\mathbf{d}}{\|\mathbf{d}\|} - \frac{\beta}{|D-D_i|}\sum_{\mathbf{d}\in D-D_i}\frac{\mathbf{d}}{\|\mathbf{d}\|}$$

D, tüm dokümanlar kümesi,

D_i, sınıfa ait dokümanlar kümesi

$\mathbf{d}$, bir dokümanın ağırlık vektörü (ağırlık değerleri)

α, β, katsayılar (TF-IDF ağırlıklandırma için genellikle $\alpha = 16$, $\beta = 4$)

46

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

47

Pseudo relevance feedback

- **Web aramada relevance feedback çok sınırlı kullanıma sahiptir.**
- **Excite arama motoru** başlangıçta relevance feedback kullanmaktaydı. Ancak, Web kullanıcıları tarafından kullanılmadığı için kaldırılmıştır.
- Web kullanıcılarının çoğu aramalarını tek sorgu girişi ile yaparlar ve relevance feedback için istekli değillerdir.
- Bu yüzden dolaylı bir şekilde relevance feedback alınmasına yönelik yöntemler geliştirilmiştir.
- Bunlardan yaygın kullanılanlar:
  - Pseudo relevance feedback
  - Indirect relevance feedback

48

Pseudo relevance feedback

Pseudo relevance feedback

- **Sorgu sonucunda elde edilen doküman listesindeki k-tane üstteki ilgili doküman olarak işaretlenir.**
- Seçilen dokümanlara göre sorgu iyileştirmesi veya prototip sınıflar oluşturulur.
- Tüm dokümanlar tekrar yeni sorguya göre değerlendirilir veya prototip sınıflara göre benzerliklerine değerlendirilir.
- Bu işlem tekrarlı bir şekilde yapılabilir.
- **İlk gelen listede recall değeri düşükse sonraki listelerde de recall değerinin giderek düşmesi olasılığı vardır.**

49

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

50

Indirect relevance feedback

- Pseudo relevance feedback yöntemine göre daha başarılı sonuçlar elde edilir.
- Web arama motorlarında kullanımı yaygındır.
- **Kullanıcının click yaptığı dokümanların ilgili diğerlerinin ilgili olmadığı varsayılır.**
- Clickstream verilerine göre sorgular iyileştirilir veya prototip sınıflar yeniden oluşturulur.
- Dokümanlar sorguya benzerliklerine göre yeniden değerlendirilir veya prototip sınıflara benzerliklerine göre sınıflandırılırlar.

51

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

52

Değerlendirme Ölçütleri

- **Web aramada dokümanların kullanıcı sorgusuyla ilgili olup olmadığına yönelik bir karar verilmez.**
- Bunun yerine, **kullanıcı için dokümanların rank değerleri hesaplanır** ve sıralama yapılır.
- Veritabanındaki dokümanların kümesi D, doküman sayısı N olsun.
- Verilen bir $\mathbf{q}$ sorgusu için retrieval algoritmaları D içerisindeki her doküman için ilgililik skoru hesaplar.
- Ardından, **ilgililik skorlarına göre tüm dokümanlar için R_q rank değerleri oluşturulur.**

$$R_q : < \mathbf{d}_1^q, \mathbf{d}_2^q, \ldots, \mathbf{d}_N^q >$$

- Burada, $\mathbf{d}_1{}^q \in D$ kullanıcı sorgusuna en ilgili eleman, $\mathbf{d}_N{}^q \in D$ *ise* en ilgisiz elemandır.

53

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

54

Precision ve recall

- $D_q \subseteq D$, kullanıcı sorgusu $\mathbf{q}$ ile gerçekten ilgili dokümanlar kümesi ise **precision ve recall değerleri hesaplanabilir.**

Recall

- Recall değeri, listede i.sıradaki bir doküman $\mathbf{d}_i^q$ için, $\mathbf{d}_1^q$ ve $\mathbf{d}_i^q$ arasındaki **ilgili dokümanların tüm ilgili dokümanlara oranıdır.**
- D_q içerisinde $\mathbf{d}_1^q$ ve $\mathbf{d}_i^q$ arasındaki ilgili dokümanların sayısı s_i ise, recall değeri aşağıdaki gibi hesaplanır:

$$r(i) = \frac{s_i}{|D_q|}$$

55

Precision ve recall

Precision

- Precision değeri, listede i.sıradaki bir doküman $\mathbf{d}_i^q$ için, $\mathbf{d}_1^q$ ve $\mathbf{d}_i^q$ arasındaki ilgili dokümanların i sayısına oranıdır.

$$p(i) = \frac{s_i}{i}$$

56

Precision ve recall

Örnek

- D doküman kümesinde 20 doküman olsun.
- Kullanıcı sorgusu $\mathbf{q}$ ile gerçekten ilgili doküman sayısı 8 olsun.
- Retrieval algoritması tablodaki rank değerlerini oluştursun.
- Precision ve recall değerleri tablodaki gibi hesaplanabilir.
- Tabloda, "+" ilgili, "-" ilgili olmayan dokümanları göstermektedir.

Rank i	+/–	$p(i)$	$r(i)$
1	+	1/1 = 100%	1/8 = 13%
2	+	2/2 = 100%	2/8 = 25%
3	+	3/3 = 100%	3/8 = 38%
4	–	3/4 = 75%	3/8 = 38%
5	+	4/5 = 80%	4/8 = 50%
6	–	4/6 = 67%	4/8 = 50%
7	+	5/7 = 71%	5/8 = 63%
8	–	5/8 = 63%	5/8 = 63%
9	+	6/9 = 67%	6/8 = 75%
10	+	7/10 = 70%	7/8 = 88%
11	–	7/11 = 63%	7/8 = 88%
12	–	7/12 = 58%	7/8 = 88%
13	+	8/13 = 62%	8/8 = 100%
14	–	8/14 = 57%	8/8 = 100%
15	–	8/15 = 53%	8/8 = 100%
16	–	8/16 = 50%	8/8 = 100%
17	–	8/17 = 53%	8/8 = 100%
18	–	8/18 = 44%	8/8 = 100%
19	–	8/19 = 42%	8/8 = 100%
20	–	8/20 = 40%	8/8 = 100%

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

58

Ortalama precision

- Kullanıcı sorgusu **q** için **farklı retrieval algoritmalarını karşılaştırmak amacıyla ortalama precision değeri kullanılabilir.**
- Ortalama precision değeri tüm ilgili dokümanların precision değerlerinin aritmetik ortalaması hesaplanarak elde edilir.

$$p_{avg} = \frac{\sum_{d_i^q \in D_q} p(i)}{|D_q|}$$

$$p_{avg} = \frac{100\% + 100\% + 100\% + 80\% + 71\% + 67\% + 70\% + 62\%}{8} = 81\%$$

59

Konular

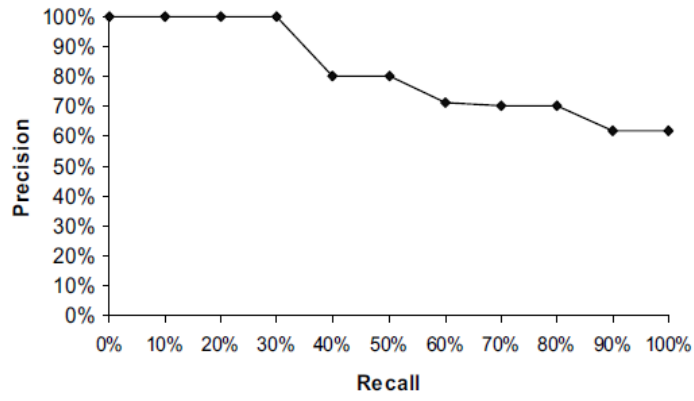
- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

60

Precision recall eğrisi

- Her bir rank pozisyonu için precision ve recall değerlerine göre precision-recall grafiği çizilebilir (x ekseni recall, y ekseni precision).
- Genellikle 11 standart aralık için çizilir (%0, %10, %20, …, %100).

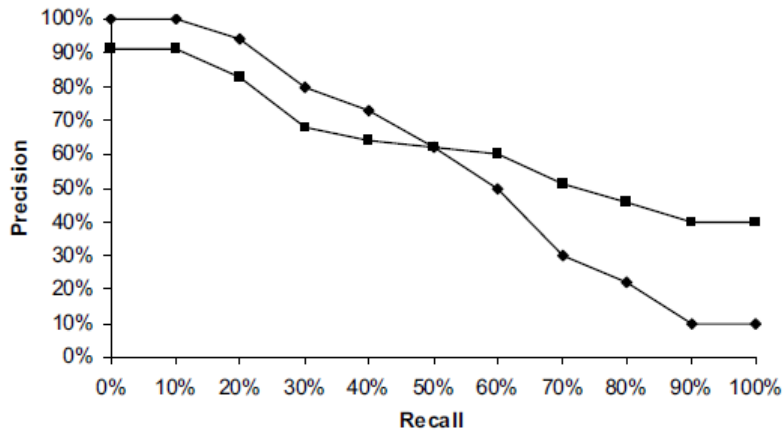
i	$p(r_i)$	r_i
0	100%	0%
1	100%	10%
2	100%	20%
3	100%	30%
4	80%	40%
5	80%	50%
6	71%	60%
7	70%	70%
8	70%	80%
9	62%	90%
10	62%	100%



61

Precision recall eğrisi

- Farklı algoritmalar precision-recall grafiğine göre karşılaştırılabilir.
- Birinci algoritmada düşük recall değerleri için precision daha iyi yüksek recall değerlerinde precision daha düşüktür.



62

Konular

- Giriş
- Bilgi Erişiminde Temel Yaklaşımlar
- Bilgi Erişim Modelleri
  - Boolean model
  - Vector space model
- İlgililik Geribildirimi
  - Rocchio yöntemi
  - Makine öğrenmesi yöntemi
  - Pseudo relevance feedback
  - Indirect relevance feedback
- Değerlendirme Ölçütleri
  - Precision ve recall
  - Ortalama precision
  - Precision recall eğrisi
  - F-score

63

F-score

- F-score değeri farklı retrieval algoritmalarının karşılaştırılması için yaygın bir şekilde kullanılmaktadır.
- F-score değeri, **hem precision hem de recall değerlerinin yüksek olduğu değerlerde yüksek değere sahiptir.**

$$F(i) = \frac{2}{\frac{1}{r(i)} + \frac{1}{p(i)}} = \frac{2\,p(i)r(i)}{p(i) + r(i)}$$

64