Bilgiye Erişim Sistemleri
Information Retrieval (IR) Systems

M.Fatih AMASYALI
BLM 5212 Doğal Dil İşlemeye Giriş Ders Notları

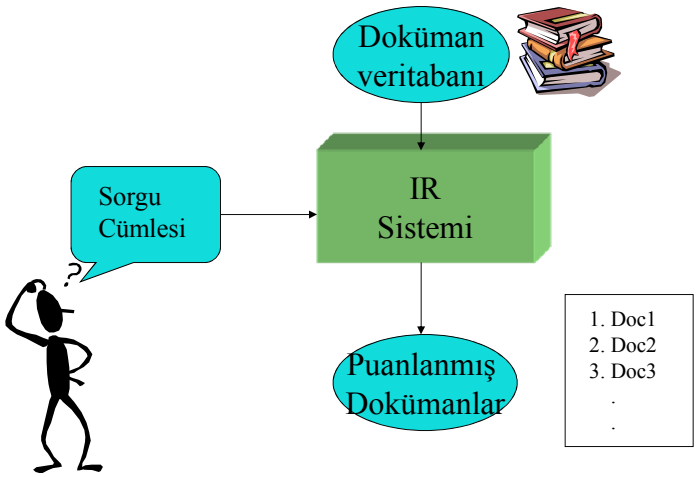
Akış

- Örnek IR Sistemleri
- IR Sistem Mimarisi
- Arama Motoru Mimarisi
- Vektör Uzayı
- Google
- Anahtar Kelime Problemleri
- Zeki IR Teknikleri
- IR Sistemlerinin Değerlendirilmesi

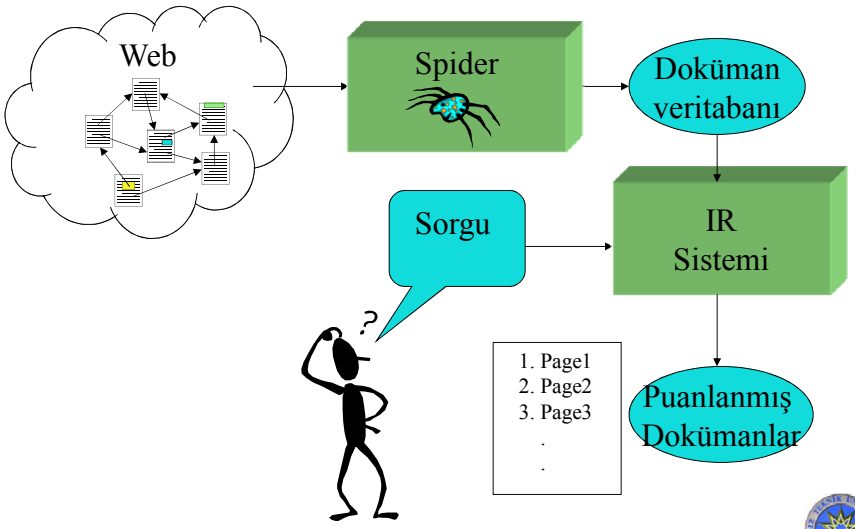
Örnek IR Sistemleri

- Kütüphane veritabanları
  anahtar kelime, başlık, yazar, konu vs. ile büyük veritabanlarında arama (www.library.unt.edu)
- Metin Tabanlı Arama Motorları (Google, Yahoo, Altavista vs).
  Anahtar kelimelerle arama
- Multimedya Arama (QBIC, WebSeek, SaFe)
  Görsel öğelerle arama (şekil, renk vs.)
- Soru Cevaplama Sistemleri (AskJeeves, Answerbus)
  Doğal dille arama

IR Sistem Mimarisi



Arama Motoru Mimarisi



Vektör Uzayı

- Varsayım: Kelimeler birbirinden bağımsızdır

Eldekiler:

N doküman
1 sorgu

$$\begin{array}{c|cccc} & T_1 & T_2 & .... & T_t \\ D_1 & d_{11} & d_{12} & ... & d_{1t} \\ D_2 & d_{21} & d_{22} & ... & d_{2t} \\ : & : & : & & : \\ : & : & : & & : \\ D_n & d_{n1} & d_{n2} & ... & d_{nt} \end{array}$$

$$Q \quad q_1 \quad q_2 \quad \ldots q_t$$

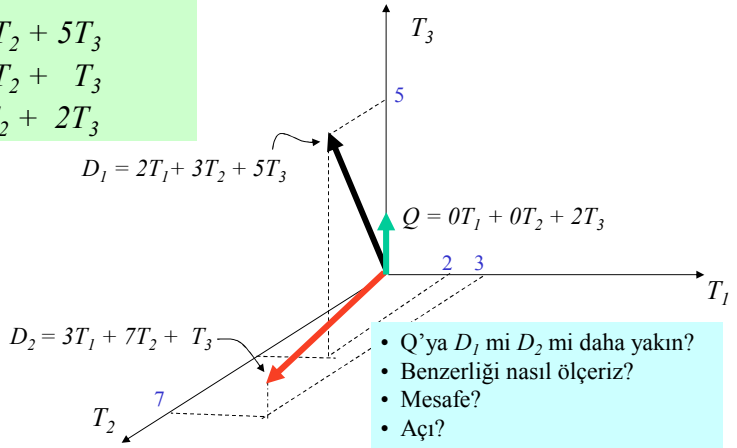
Grafiksel Gösterim

Örnek:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$



- Q'ya D_1 mi D_2 mi daha yakın?
- Benzerliği nasıl ölçeriz?
- Mesafe?
- Açı?
- Projeksiyon?

Benzerlik Ölçümü- Inner Product

$$\text{sim}(D_i, Q) = \sum_{k=1}^{t} (D_i \bullet Q)$$

$$= \sum_{j=1}^{t} d_{ij} * q_j$$

Inner Product - Örnek

Binary:

	retrieval	database	architecture	computer	text	management	information
D =	1,	1,	1,	0,	1,	1,	0
Q =	1,	0 ,	1,	0,	0,	1,	1

- Vektör boyutu = Sözlük boyutu = 7

→ sim(D, Q) = 3

Ağırlıklı

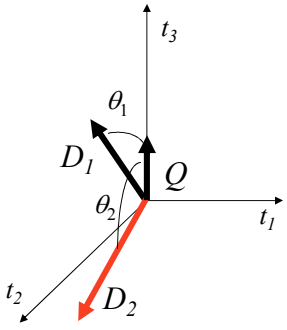
$$D_1 = 2T_1 + 3T_2 + 5T_3$$

$$Q = 0T_1 + 0T_2 + 2T_3$$

$$\text{sim}(D_1, Q) = 2*0 + 3*0 + 5*2 = 10$$

Cosine Benzerlik Ölçümü

- İki vektör arasındaki açının cosinüsü
- Inner product, vektör büyüklükleriyle normalize edilir.



$$\text{CosSim}(D_i, Q) = \frac{\sum_{k=1}^{t} (d_{ik} \bullet q_k)}{\sqrt{\sum_{k=1}^{t} d_{ik}^{\,2} \bullet \sum_{k=1}^{t} q_k^{\,2}}}$$

Cosine Benzerlik: Örnek

$D_1 = 2T_1 + 3T_2 + 5T_3$ CosSim(D_1 , Q) = 0.81
$D_2 = 3T_1 + 7T_2 + T_3$ CosSim(D_2 , Q) = 0.13
$Q = 0T_1 + 0T_2 + 2T_3$

Doküman ve Terim Ağırlıkları

- Ağırlıklar dokümanlardaki frekanslarla (*tf*) ve tüm doküman kütüphanesindeki frekanslarla (*idf*) hesaplanır.

  tf_{ij} = *j*. terimin *i*. dokümandaki frekansı
  df_j = *j*. terimin doküman frekansı
  = *j*. terimi içeren doküman sayısı
  idf_j = *j*. terimin ters doküman frekansı
  = $\log_2 (N/df_j)$ (N: toplam doküman sayısı)

Terim Ağırlıklarının Bulunması

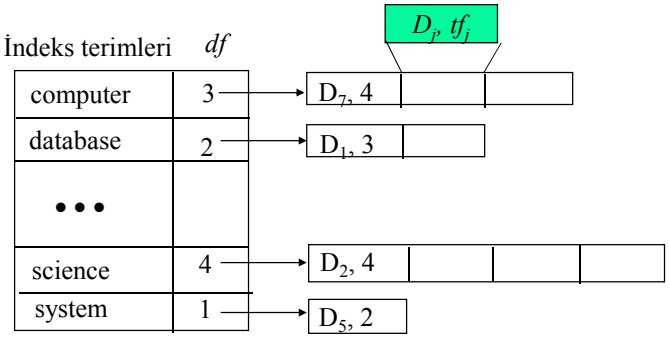
- *j.* terimin *i.* doküman için ağırlığı:

$$d_{ij} = tf_{ij} \bullet idf_j = tf_{ij} \bullet \log_2 (N/df_j)$$

- TF → Terim Frekansı
  - Bir dokümanda sıkça geçen ancak diğer dokümanlarda pek bulunmayan terimin ağırlığı yüksek olur.
  - $max_l\{tf_{li}\}$ = *i.* dokümanda en çok geçen terimin frekansı
  - Normalizasyon: terim frekansı $= tf_{ij}/max_l\{tf_{li}\}$

Terim Frekansı Uygulaması

- Pratikte doküman vektörleri direkt olarak saklanmaz. Hafıza problemlerinden ötürü, arama için aşağıdaki gibi bir yapıda saklanırlar.



Inverted index

- Metinler

*T*0 = "it is what it is"
*T*1 = "what is it"
*T*2 = "it is a banana“

"what", "is" ve "it" kelimeleriyle arama yapılırsa.

$$\{0,1\} \cap \{0,1,2\} \cap \{0,1,2\} = \{0,1\}$$

inverted file index:

- "a": {2}
- "banana": {2}
- "is": {0, 1, 2}
- "it": {0, 1, 2}
- "what": {0, 1}

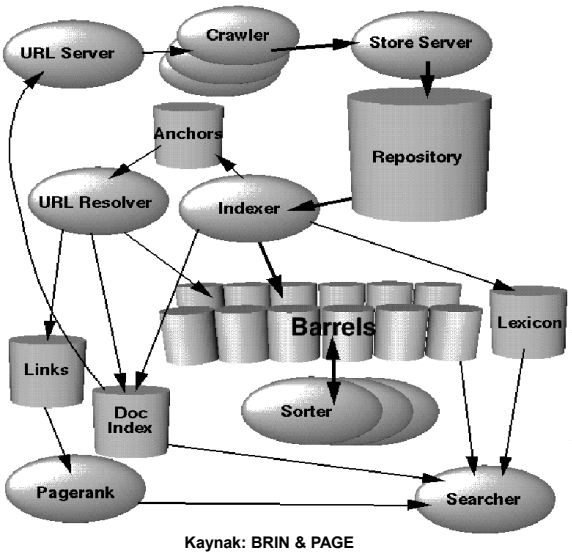
Full inverted file index: (pozisyonları da içerir)

- "a": {(2, 2)}
- "banana": {(2, 3)}
- "is": {(0, 1), (0, 4), (1, 1), (2, 1)}
- "it": {(0, 0), (0, 3), (1, 2), (2, 0)}
- "what": {(0, 2), (1, 0)}

Web Katalogları vs. Arama Motorları

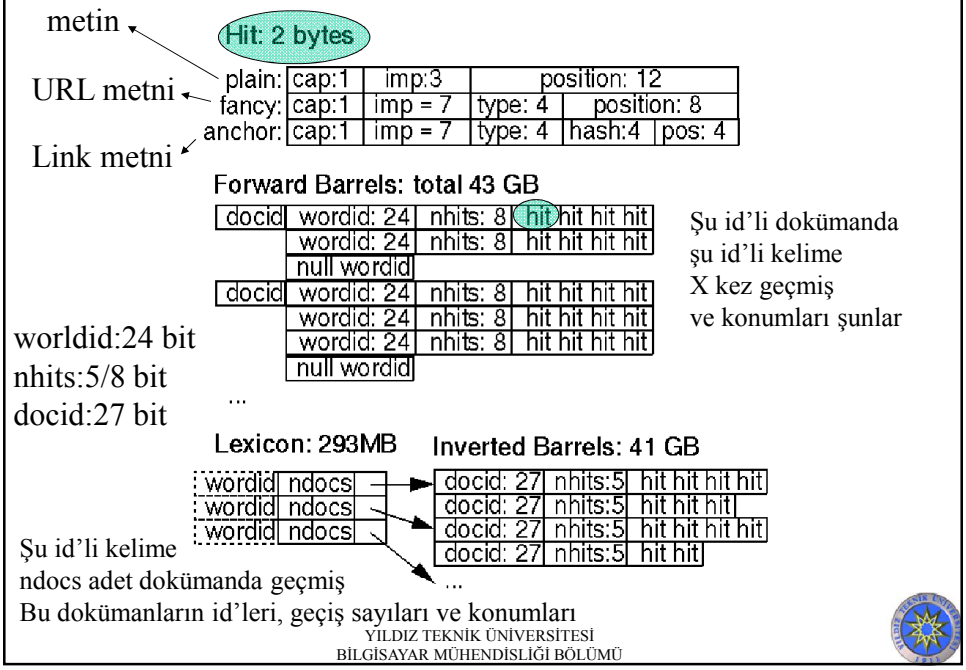
- Web Katalogları
  - Elle seçilmiş siteler
  - Sayfaların içeriğinde değil, tanımlarında arama
  - Hiyerarşik kategorilere atanırlar
- Arama Motorları
  - Tüm sitelerdeki tüm sayfalar
  - Sayfaların içeriğinde arama
  - Sorgu geldikten sonra bulunan skorlara göre sıralanırlar.

Google'ın ilk Mimarisi



Kaynak: BRIN & PAGE

- URL Server- kaydedilecek URLler
- Crawler paralel
- Store Server – crawler'lardan gelen sayfaları sıkıştırıp repository'e kaydeder.
- Repository – sayfaların HTML kodları
- Indexer – forward barrel'leri ve anchors'u olusturur.
- Lexicon – tekil kelime listesi, yer aldığı docID'ler
- Barrels (docID, (wordID, hitList*)*)* ler
- Anchors – sayfalarda bulunan link bilgileri (from, to, anchor text)
- URL Resolver – Anchors'daki rölatif URLleri gercek URL'lere dönüştürür. DocID'lerini verir. Links'i oluşturur. Anchor metinlerini forward barrel'lara ekler
- Sorter – inverted barrel'leri oluşturur.
- Doc Index – her sayfa hakındaki bilgiler (durum, repository'deki pointer vs.)
- Links – docID ikilileri
- Pagerank – her sonuç sayfasının önemini, popülerliğini (links'ten) hesaplar
- Searcher – sorguları cevaplar



Şu id'li dokümanda şu id'li kelime X kez geçmiş ve konumları şunlar

Puanlama Sistemi

- Kriterler
  - Pozisyon, Font Büyüklüğü, Büyük Harfle/ Bold/italik yazılma
  - Sitenin popülerliği (PageRank)
  - Başlık ,link metni (Anchor Text), URL metni vs.

İlk Google'ın depo büyüklükleri

Total Size of Fetched Pages	147.8 GB
Compressed Repository	53.5 GB
Short Inverted Index	4.1 GB
Full Inverted Index	37.2 GB
Lexicon	293 MB
Temporary Anchor Data (not in total)	6.6 GB
Document Index Incl. Variable Width Data	9.7 GB
Links Database	3.9 GB
Total Without Repository	**55.2 GB**
Total With Repository	**108.7 GB**

Web istatistikleri

Number of Web Pages Fetched	24 million
Number of URLs Seen	76.5 million
Number of Email Addresses	1.7 million
Number of 404's	1.6 million

Sistem Performansı

- 26 milyon site 9 günde indirilmiş.(Saniyede 48.5 sayfa)
- Indexer ve Crawler aynı anda çalışıyor
- Indexer saniyede 54 sayfayı indeksliyor
- Sorter'lar 4 makinede paralel çalışarak 24 saatte inverted index'i oluşturuyor

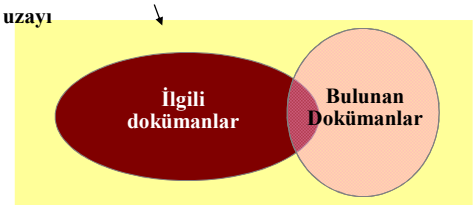
Anahtar Kelime ile Aramada Problemler

- Eşanlamlı kelimeleri içeren dokümanlar bulunamaz.
  - "PRC" vs. "China"
- Eşsesli kelimeler ilgisiz dokümanların bulunmasına sebep olabilir.
  - "bat" (baseball vs. mammal)
  - "Apple" (company vs. fruit)
  - "bit" (unit of data vs. act of eating)

Zeki IR Teknikleri

- Kelimelerin anlamları
- Sorgudaki kelimelerin sırası
- Kullanıcılardan döndürülen sonuçların kalitesiyle ilgili alınan geri bildirimler (sonuçların kaçıncısına tıkladı, kaç sonuç sayfası inceledi vb.)
- Aramayı ilgili kelimelerle genişletmek
- İmla denetimi – kelime önermek
- Kaynakların güvenilirliği
- Kişiselleştirilmiş arama
- Eklemeli dillerde bazı eklerden bağımsızlık

Değerlendirme



relevant / irrelevant	retrieved	not retrieved
retrieved & irrelevant	Not retrieved & irrelevant	
retrieved & relevant	not retrieved but relevant	

$$recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents}$$

$$precision = \frac{Number\ of\ relevant\ documents\ retrieved}{total\ Number\ of\ documents\ retrieved}$$

Kaynaklar

- http://www.cs.huji.ac.il/~sdbi/2000/google/index.htm
- Searching the Web ,Ray Larson & Warren Sack
- Knowledge Management with Documents, Qiang Yang
- Introduction to Information Retrieval, Rada Mihalcea
- Wikipedia
- Introduction to Information Retrieval, Evren Ermis
- The Anatomy of a Large-Scale Hypertextual Web Search Engine, Sergey Brin, Lawrence Page (http://infolab.stanford.edu/~backrub/google.html)