VERİ MADENCİLİĞİ
Metin Madenciliği

Yrd. Doç. Dr. Şule Gündüz Öğüdücü

http://www3.itu.edu.tr/~sgunduz/courses/verimaden/

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Terim sayısını azaltma

Metin için Veri Madenciliği

- Metin madenciliği: Veri madenciliği teknikleri ile yazılı belgeler arasındaki (içindeki) ilişkileri, örüntüleri bulmak.
  - doğal dilde yazılmış metinler
  - Aynı konudaki belgeleri bulmak
  - Birbiriyle ilişkili belgeleri bulmak
  - Bulunan belgeleri sıralamak
- Bilgi elde etme sistemleri birleştirilebilir
  - Ortak ön işleme işlemleri: doğal dil işleme yöntemleri ile
  - bilgi elde etme sistemleri metin madenciliği sonuçlarını değerlendirmeli, yorumlamalı

Metin Madenciliğinde Sorunlar

- Metin yapısal değil
  - Sorgulama sonucuyla ilintili metinleri bulmak zor
  - Önemsiz veri (sözcük) çok fazla
- Hatalar var
  - Metin içinde hatalar var
  - Kavram oluşturmak zor: eş anlamlı, eşsesli, üst grup
  - Anlam çıkarmak zor: X'in iyi bir bilgisayar olduğunu düşünüyorum - X'in iyi bir bilgisayar olduğundan şüpheliyim
- Sonuç
  - Belgeler arasındaki ilişkilendirme hatalı

Metin Madenciliği Araçları

- Ticari ürünler
  - http://www.clearforest.com/
  - http://www.trl.ibm.com/projects/textmining/takmi/takmi_e.htm
  - http://www.megaputer.com/
- Metin inceleme
  - http://www.textanalysis.info/
- Problem
  - Dile özel çözümler
  - Sonuçlar yetersiz

Metin Veritabanları & Bilgi Erişim Sistemleri

- Metin Veritabanları (belge veritabanları)
  - Farklı kaynaklardan dokuümanlar: haber, makale, kitap, elektronik kütüphane, elektronik podta, web sayfaları..
  - Veri genelde yapısal değil
  - Bilgi erişim sistemleri büyük miktardaki veri üzerinde başarılı değil
- Bilgi Erişim (Information Retrival) Sistemleri
  - Veritabanları ile birlikte gelişmiş bir araştırma alanı
  - Bilgi belgeler şeklinde yer alıyor

Bilgi Erişim Sistemi

- Kullanıcının ilgi alanına ve isteğine en uygun belgeleri bulma
  - Kullanıcın girdiği bir sorgulamaya göre
  - Kullanıcının ziyaret ettiği sayfalara göre
- Internet ortamında web sayfalarının içeriğinin incelenmesini gerektirir
- Bilgi erişim yönteminde problemler
  - Büyük bir belgeler kümesindeki belgeleri işaretleme
    - erişimin kolay olması için
  - Seçilen belgelerin sıralanması
  - Belgelerin sınıflandırılması: veri madenciliği yöntemleri kullanılabilir

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Terim sayısını azaltma

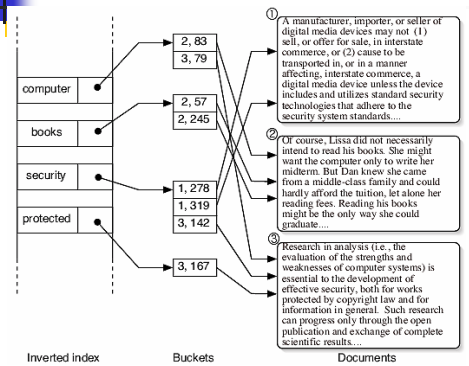
Dizin Oluşturma

- Ters dizin
  - Belgelerden oluşan veri kümesinde her sözcüğün hangi belgelerde görüldüğü işaretlenir
  - Büyük veri kümeleri için etkili
- Terim ω
  - sözcükler veya ifadeler
- Sözcük dağarcığı V
  - Terimlerden oluşan küme

Ters Dizin

- Sözlük
  - her anahtar bir terim $\omega \in V$
  - anahtara ait veriler zincir (atama listesi) $b(\omega)$: ω teriminin, belgeler kümesinde her görüldüğü yeri işaret eden işaretçiler listesi
    - belge kimlik numarası (DID): belgenin küme içinde kaçıncı sırada yer aldığı
    - terimin her görüldüğü yer için ayrı bir işaretçi
      - DID
      - Terimin belge içindeki konumu

Ters Dizin



Ters Dizin Oluşturma

- Belgeler ayrıştırılır
- Terimler bulunur ω_i
  - Eğer ω_i dizinde yer almıyorsa eklenir
- Terimin bulunduğu yer zincire eklenir
- Ters dizin boyutu = $\Omega(|V|)$
- Hash tablosu kullanarak gerçeklenebilir
  - Zincirler bellekte
  - Zincirler diskte
    - Diske erişim süresinden dolayı elverişsiz
      - Özel ikincil depolama yapıları kullanılması gerekir

Zincirlerin Sıkıştırılması

- Zincir için gerekli saklama alanını azaltma
  - her terim için zincir DID'ye göre sıralanır
  - DID'ler arasındaki fark saklanır
- Bellek kullanımı önemli ölçüde azalır
  - Belgeler kümesinde sık yer alan terimlerin DID'leri arasında fark da azdır
  - Küçük sayılar kodlanarak bellekte daha az yer kaplarlar
- Örnek
  - DID listesi: (14, 22, 38, 42, 66, 122, 131, 226)
  - DID'ler arasındaki fark listesi: (14, 8, 16, 4, 24, 56, 9, 95)

Ters Dizin ile Arama

- Bir belgeler kümesi için oluşturulmuş ters dizinde bir terimi ω bulmak için
  - ters dizinde ω terimine ait zincir $b(\omega)$ bulunur
  - zincir taranarak terimin bulunduğu yerlerin listesi elde edilir
- Bir belgeler kümesi için oluşturulmuş ters dizinde k adet terim bulmak için
  - k adet liste oluşturulur
  - küme işlemleri ile listeler birleştirilir

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Terim sayısını azaltma

Metin Ön İşleme

- Belgeler için dizin oluşturmadan önce ön işleme işlemleri
  - İşaretleme
    - Metin içindeki terimleri ayıklama
    - HTML etiketlerinden arındırma
  - Farklı terimleri belirleme
    - Kök bulma: Aynı kökten gelen farklı ek almış sözcüklerin köklerini bulma
    - Sık geçen sözcükleri ayıklama: bağlaçlar, edatlar
    - Terim sayısında %20-30 oranında azalma

Kök Bulma

- Sözcüklerin biçimbirimsel çözümlemesini yaparak terimleri elde etmek
  - Örnek: İçinde *balıkçılık* sözcüğü geçen bir sorgulama için, içinde *balık* ve *balıkçı* geçen belgelerin bulunması
- İngilizce için kök bulma: Porter Stemming Algorithm
  - http://www.tartarus.org/~martin/PorterStemmer/
- Türkçe için kök bulma: Zemberek projesi
  - https://zemberek.dev.java.net/

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Terim sayısını azaltma

Sorgulama Sonuçlarını Sıralama

- Sorgulama içinde geçen terimlerin yer aldığı belgelerin sayısı çok fazla
  - *data mining* için Google arama motorunda dönen sonuç: 68.400.000
- kullanıcı ancak küçük bir kısmını inceleyebilir
  - sorgulama sonuçlarını sıralamak gerekir
    - sorgulamayla daha ilgili olan sonuçların başlarda yer alması

Vektör Uzayı Modeli

- Belgeler çok boyutlu vektör uzayında temsil edilir
- Belgeler terim vektörleri biçiminde
  $d = (\omega(1), \omega(2), \omega(3), ..., \omega(|d|))$
- Belgeler kümesindeki ayrık terim sayısı vektör uzayının boyutunu belirler

Örnek

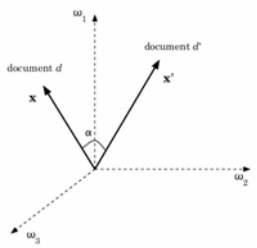
belge	metin	terimler
d_1	*web web graph*	*web graph*
d_2	*graph web net graph net*	*graph web net*
d_3	*page web complex*	*page web complex*

- Belgelerin boolean modeline göre temsil edilmesi:

V = [web, graph, net, page, complex]
V1 = [1 1 0 0 0]
V2 = [1 1 1 0 0]
V3 = [1 0 0 1 1]

Vektör Uzayı Modeli

- x, x': belge vektörleri
- ω_1, ω_2, ω_3:terimler
- Vektör uzayında belgelerin gösterilimi seyrek: |V|>>|d|



Terim Sıklığı (TF)

- Bir belge içinde, diğer terimlere göre daha sık yer alan bir terimin önemi de daha fazladır
- n_{ij} : ω_j teriminin d_i belgesinde yer alma sayısı
- Terim sıklığı (term frequency):

$$TF_{ij} = \frac{n_{ij}}{|d_i|}$$

Devrik Belge Sıklığı (IDF)

- Belgeler kümesinde daha az sayıda belgede görülen bir terimin ayırt edici özelliği daha fazladır
- n_j : belge sıklığı (document frequency: df) - ω_j teriminin geçtiği belge sayısı
- n : belgeler kümesindeki belge sayısı
- ω_j teriminin devrik belge sıklığı (Inverse Document Frequency):

$$IDF_j = \log \frac{n}{n_j}$$

- Belge sıklığı (df) arttıkça, devrik belge sıklığı (idf) azalır

Tam Ağırlıklandırma (TF-IDF)

- Bir belgede çok bulunan ancak diğer belgelerde daha az görülen bir terimin ağırlığı daha fazla
- ω_j teriminin d_i belgesindeki TF-IDF ağırlığı:

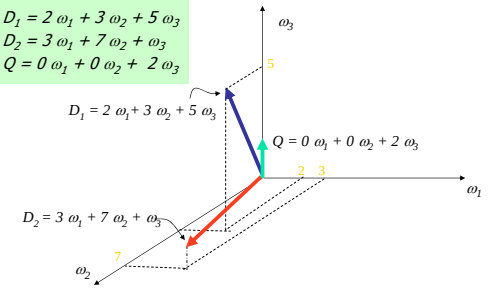
$$x_{ij} = TF_{ij} \times IDF_j$$

Belgeler Arası Benzerlik

- Sorgulama ile herbir belge arasındaki benzerlik hesaplanıp, benzerlik sonucuna göre sıralanır
- Herhangi iki belge arasındaki benzerlik $s(d, d') \in R$
- Vektör uzayı modelinde kosinüs benzerliği belgeler arası benzerliği hesaplamak için kullanılabilir

Grafik Gösterilim

- Örnek:
  $D_1 = 2\,\omega_1 + 3\,\omega_2 + 5\,\omega_3$
  $D_2 = 3\,\omega_1 + 7\,\omega_2 + \omega_3$
  $Q = 0\,\omega_1 + 0\,\omega_2 + 2\,\omega_3$



Kosinüs Benzerliği

- İki vektör arasındaki açının kosinüsü
  $\cos(d_1,d_2) = d_1 \bullet d_2 / ||d_1||\,||d_2||$
  $d_i \bullet d_j$: iki dokümanın vektör çarpımı
  $||d_i||$: d_i dokümanının uzunluğu

- **Örnek**
  d_1 = 3 2 0 5 0 0 0 2 0 0
  d_2 = 1 0 0 0 0 0 0 1 0 2

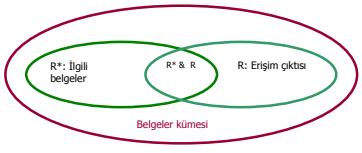
  $d_1 \bullet d_2$ = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5
  $||d_1||$ = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)$^{0.5}$ = (42) $^{0.5}$ = 6.481
  $||d_2||$ = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)$^{0.5}$ = (6) $^{0.5}$ = 2.245

  $\cos(d_1, d_2)$ = .3150

Sonuçların Değerlendirilmesi

- Belgeler kümesindeki belgelerin vektör uzayı modeli bulunur
- Kullanıcı sorgusu Q için vektör uzayı modeli bulunur
- Belgeler kümesindeki her belge için sorgulamayla olan benzerliği hesaplanır $s(d_i, Q),\ i=1,2,\ldots,n$
- En fazla benzerlik değerine sahip olan belgeler kümesi R erişim çıktısı olarak belirlenir
- Belgeler kümesinde sorgulamayla ilgili belgeler kümesi R^* ve R karşılaştırılır.

Sonuçların Değerlendirilmesi



- Duyarlılık (Precision): Erişim çıktısındaki ilgili belge sayısının erişim çıktısındaki belge sayısına oranı

$$precision = \frac{|R^* \cap R|}{R}$$

- Anma (Recall): Erişim çıktısındaki ilgili belge sayısının belgeler kümesinde ilgili belgeler sayısına oranı

$$recall = \frac{|R^* \cap R|}{|R^*|}$$

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Terim sayısını azaltma

Olasılıklı Bilgi Erişim Sistemleri

- Temel varsayım: Kullanıcının sorgulamasına göre sadece ilgili belgelerden oluşan bir belgeler kümesi var (ideal durum)
- Probabilistic Ranking Principle (PRP) (Robertson, 1977)
  - Belgelerin kullanıcının sorgulamasına ilgili olma olasılığına göre sıralanması
  - Olasılıklar eldeki veriye göre mümkün olan en doğru şekilde hesaplanır
  - Eldeki veri ile gerçeklenebilecek en iyi sistem

Olasılıklı Bilgi Erişim Sistemleri

- Sorgu terimlerinin bir belgede bulunabilme olasılığı *P(R | d,q)*
- Belgeler olasılıklara azalacak şekilde sıralanır

$$P(R \mid d, q) \geq P(R \mid d', q)$$

d' : erişim çıktısında yer alamayan belge

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Terim sayısını azaltma

Gizli Anlamsal İnceleme

- Latent Semantic Analysis
- Vektör uzayı modeli sorgulama içinde geçen terimler belge içinde de yer alıyorsa iyi sonuç veriyor
- Doğal dilin zenginliği nedeniyle sorunlar
  - Kullanıcı sorgulamalarında genelde kavramlar yer alıyor
  - eş anlamlı sözcükler: hediye – armağan
    - anma değerini etkiliyor
  - eş sesli sözcükler: çay
    - duyarlılık değerini etkiliyor

Latent Semantic Indexing (LSI)

- Lineer cebirdeki tekil değer ayrışımı (singular value decomposition, SVD) yöntemi kullanılır
  - Veri içindeki gizli yapıyı bulmayı hedefler
  - Sözcükler ve kavramlar arasındaki önemli ilişkileri bulmayı hedefler
- ***D***: terim – belge matrisi $\boldsymbol{D} = [\boldsymbol{d}_1 \ldots \boldsymbol{d}_n]^T$
  - her satır belgelerin vektör uzayı modelindeki gösterilimi
  - her kolon terimin belgede yer alma sayısı
- ***D*** matrisinin tekil değer ayrışım matrisi **Σ** hesaplanır

$$D = U\Sigma V^T$$

- Tekil değer ayrışım matrisindeki en büyük K değer dışındakiler sıfırlanır $\hat{\Sigma}$
- **D** terim belge matrisi yeniden oluşturulur $\hat{D} = U\hat{\Sigma}V^T$

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Terim sayısını azaltma

Nitelik Seçme

- Nitelik -> terim
- Belgeler kümesindeki tüm belgelerdeki ayrık terim sayısı çok fazla
  - belgeler arasında ayırıcılık sağlamayan terimler
  - model öğrenme sırasında zaman karmaşıklığı artıyor