2 BİLGİ ERİŞİM SİSTEMLERİ

Bir bilgi erişim sisteminin temel işlevi, kullanıcıların bilgi ihtiyaçlarını karşılaması muhtemel derlemdeki ilgili (relevant) belgelerin tümüne erişmek, ilgili olmayanları da ayıklamaktır. Bir bilgi erişim sisteminin bazı belgelere erişim sağlayabilmesi için iki koşul yerine getirilmelidir. İlki, derleme eklenen her belgenin temel özellikleri geleneksel veya otomatik olarak gerçekleştirilen dizinleme işlemleri sırasında belirlenmeli ve her belge için ilgili içerik belirteçleri (dizin terimleri) oluşturulmalıdır. Bir belge için oluşturulan söz konusu içerik belirteçleri bilgi erişim sırasında belgenin tamamını temsil etmek üzere (surrogates) kullanılır. İkincisi, kullanıcılar belgelere verilen bu içerik belirteçlerini doğru olarak tahmin edip sorgu cümlelerini ona göre oluşturmalıdırlar. Bir başka deyişle, kullanıcının bilgi ihtiyacını ifade etmek için kullandığı terimlerle belgeyi temsil eden içerik belirteçleri birbiriyle karşılaştırılır ve çakışan belgelere erişilir (Tonta, 1995, 1992). Çakışma "Erişim Kuralı" (Retrieval Rule) olarak adlandırılan kuralı izler. Maron (1984, s.155) bu kuralı şöyle açıklamaktadır: "Herhangi bir resmi (formel) sorgu [cümlesi] için bu arama sorgusunda belirlenen tutanakların (records) alt setinde yer alan dizin tutanaklarının tümüne ve salt bu dizin tutanaklarına erişim sağla." Böylece, bir bilgi erişim sisteminin temel bileşenlerinin: (1) bir belge derlemi (ya da bu belgeleri temsil eden içerik belirteçlerini içeren tutanaklar), (2) kullanıcıların sorgu cümleleri, ve (3) kullanıcıların sorgu cümlelerinde yer alan terimlerle derlemdeki belgelere verilen terimleri karşılaştırarak ilgili belgeleri belirlemek için kullanılan bir erişim kuralından oluştuğu ortaya çıkmaktadır.

Şekil 1'deki işlevsel mimaride de görüleceği üzere, sistemi oluşturan temel bilgi erişim süreçlerini üçer tane ön yüz (front-end) ve arka yüz (back-end) kavramları çerçevesinde tanımlamak mümkündür. Bu şekilde kavramlar dikdörtgen, temel süreçler oval, seçenekli süreçler ise kesikli oval şekillerle gösterilmiştir. Ön yüz kavramları sistemin dış dünyaya yansıyan görünüşünü oluşturmaktadır. Benzer şekilde arka yüz kavramları kullanıcıya saydam olup bilgi erişim süreçleri arasındaki iletişimde kullanılır. Bilgi ihtiyacı, metin nesneleri ve erişim çıktısı ön yüz, sorgular, belgeler ve içerik belirteçleri arka yüz kavramlarını oluşturur.

Bilgi ihtiyacı bir düz metinle (doğal dille) ifade edilebileceği gibi dizin terimleri ve aralarındaki ilişkiler ("ve", "veya", "ve-değil", "ise/eğer", vb.) çerçevesinde de tanımlanabilir. Metin nesneleri arka planda işleyen otomatik dizinleme sürecine giriş oluşturur ve sonuçta belgeler ters dizin kütüğü (inverted file) düzenlemesi içinde içerik belirteçleri ile öznel (subjektif) olarak gösterilirler. Buradaki öznellik metin nesnelerinin içerik belirteçleri ile

gösteriminin ileride de göreceğimiz üzere çeşitlilik göstermesidir.[1] Bunun aksini ise metin yazarı, adı, yayıncı bilgisi, yayın tarihi, türü, gibi nesnel (objektif nitelikler) oluşturur.[2] Erişim çıktısı eldeki sorgu ifadesinin belgeler (ve/veya onların öznel/nesnel nitelikleri) ile eşleştirilmesiyle oluşturulurlar; yani sistemin, belge derlemi (koleksiyonu) içinde sunulan sorgu ifadesi ile ilgili olduğunu "düşündüğü" belgeleri topladığı havuza (formel anlamıyla "küme"ye) erişim çıktısı adını vermekteyiz. Erişim çıktısındaki belgeler kullanıcı bilgi ihtiyacına yakınlık derecesine göre azalan sırada sıralanırlar.[3]

Arka yüz kavramları aslında üç temel sonlu nesne küme notasyonuna karşılık gelirler. Bunlar sırasıyla *belgeler*, *içerik belirteçleri* (*anahtar sözcükler*, *dizin terimleri*[4]*)* ve *sorgulardır*. Kullanılan model ne olursa olsun, sorgular mutlaka belgeler (ya da belgeleri temsil eden içerik belirteçleri) ile eşleştirilmelidir -ki bu eşleştirmeye erişim kuralı (ya da erişim işlevi) denir. Şekil 1'de kümeleme (clustering) süreci bir anlamda aşırı yüklenmiştir. Sorguları, belgeleri ve içerik belirteçlerini tek tek özyineli (recursive) olarak temel alan kümeleme süreçleri, aynı ad ile anılmalarına rağmen amaçları ve/veya uygulanan teknikler açısından birbirlerinden farklılık gösterebilirler.[5] Şöyle ki, içerik belirteçleri eş anlamlılık temelinde kümelendirildiklerinde amaç sorgu genişletebilme ve yerden kazanç sağlama (metin nesnelerinin daha az sayıdaki belirteçler ile gösterilmesi) olmasına rağmen, belgelerin kümelendirilmesinde amaç eşleştirme sürecinin hızlandırılmasıdır. Sorguların kümelendirilmesinde ise, zaman açısından pahalı bir süreç olan geribildirim sürecine olan ihtiyacı azaltma ya da geribildirim sürecini kısa zamanda sonuçlandırma kaygısı olabileceği gibi (Mettrop ve Nieuwenhuysen, 2001)[6], performans etkinliği daha yüksek olan bilgi erişim sistemleri gerçekleştirme hedefi de güdülebilir (Lee, 1995; Belkin, Kantor, Fox ve Shaw,

---

[1] Ne şekilde dizinleme yapılırsa yapılsın, ilgili süreç sonucunda elde edilen gösterim (içerik belirteçleri kümesi) özneldir. Başka bir deyişle, bir belgenin birden fazla (ve doğru) gösterim şekli olabilir. Dizinleme işleminin elle ya da otomatik olarak yapılması bu gerçeği değiştirmez.

[2] Kütüphanecilikte bir bilgi kaynağıyla ilgili nesnel niteliklerin (yazar adı, başlık vs.) belirlenmesine "tanımlayıcı kataloglama", kaynağın hangi konu ya da konular hakkında olduğunun belirlenmesine ise "konu kataloglaması" adı verilmektedir.

[3] Başka bir deyişle, erişim çıktısı erişim fonksiyonunun değişimini oluşturan sıralı belgeler kümesidir.

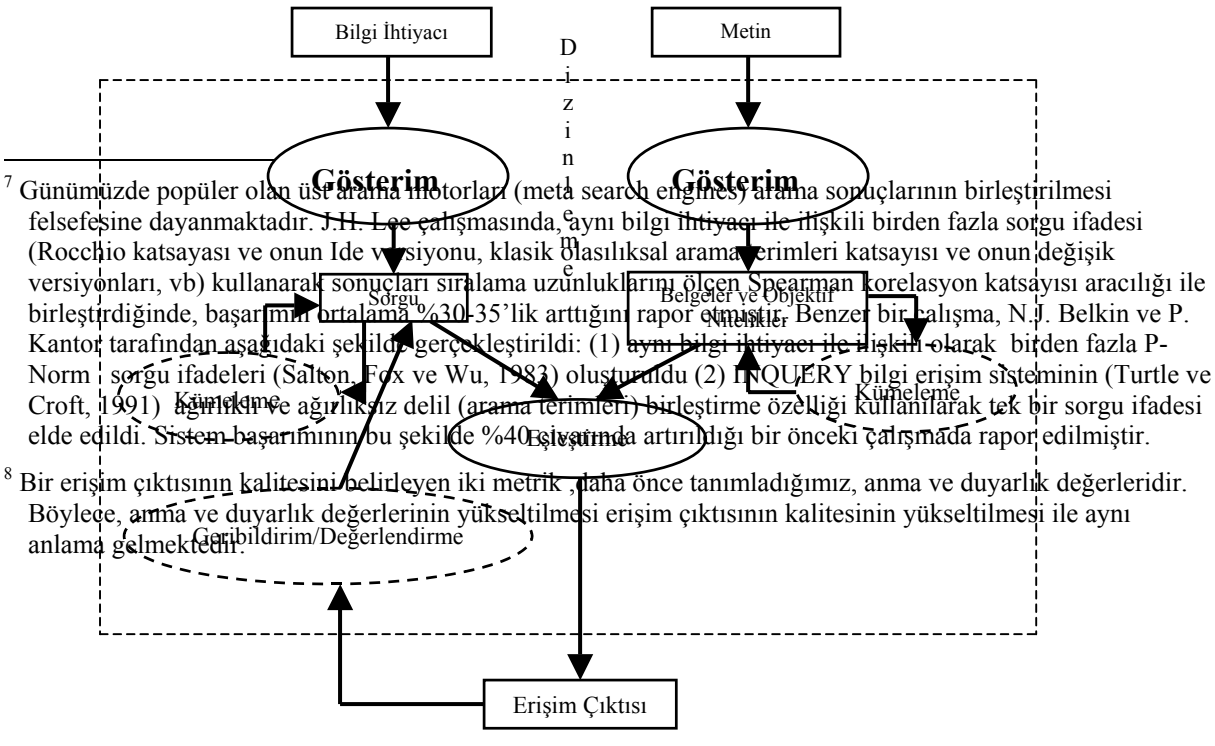
[4] "İçerik belirteçleri", "dizin terimleri" ve "anahtar sözcükler" makale boyunca eş anlamlı olarak kullanılmaktadır.

[5] İleride de belirtileceği üzere erişim çıktısındaki belgelerin kümelendirilmesi arama motorlarında kullanıcı arayüzü tasarımının bir parçası olarak önem kazanmıştır (Leuski, 2001). Belgeler tek tek ilgililik derecesine göre kullanıcıya sunulmaz, bunun yerine genellikle iki veya daha fazla belgeden oluşan öbekler halinde kullanıcıya sunulur. Google arama motoru (www.google.com) olaya benzer bir perspektiften bakarak içerik olarak aynı olan fakat farklı site adreslerine sahip belgeleri eleme amacıyla erişim çıktısını arka planda kümeleme tekniği uygulamaktadır.

[6] Sorguları kümelendirmede kullanılan ve ikili tercih ilişkisi tabanlı basamak inme algoritması (steepest descent algorithm) bilgi süzgeçleme alanına da başarıyla uygulanmıştır (Mettrop ve Nieuwenhuysen, 2001).

1995).[7] İçerik belirteçlerinin kümelendirilmesinde LSA (Latent Semantic Analysis) tekniği (Deerwester, Dumais, Furnas, Landauer ve Harshman, 1990; Foltz, 1996), belirteçlerin ayrım gücünü temel alan sıradüzensel (hiyerarşik) (Van Rijsbergen, 1979) veya düz kümeleme teknikleri kullanılabilir (Salton, 1989; Salton, Wong ve Yu, 1976; Sezer, 1999). Oysaki, sorguların kümelenmesinde sorgularla ilgili belgelerin kesişim derecesi temel alınılır.

Şekil 1'de görüldüğü üzere, tipik bir bilgi erişim sistemi geribildirim özelliğine sahiptir. Sistem tarafından döndürülen belge çıktısının kullanıcının bilgi ihtiyacını karşılamaktan uzak olduğu durumlarda, kullanıcı geribildirim sürecini başlatarak daha kaliteli bir belge çıktısı[8] elde etmek isteyebilir. İleride değinileceği üzere, tipik bir geribildirim sürecinde, hata (herhangi bir belgenin eldeki bilgi ihtiyacı ile ilgili olması bağlamında, sistem kararının kullanıcı görüşü ile örtüşmemesi) oranının tekrarlı ve etkileşimli bir süreç boyunca kullanıcının tatmin olabileceği bir düzeye indirgenmesi hedeflenir (Salton ve Buckley, 1990).



[7] Günümüzde popüler olan üst arama motorları (meta search engines) arama sonuçlarının birleştirilmesi felsefesine dayanmaktadır. J.H. Lee çalışmasında, aynı bilgi ihtiyacı ile ilişkili birden fazla sorgu ifadesi (Rocchio katsayası ve onun Ide versiyonu, klasik olasılıksal arama terimleri katsayısı ve onun değişik versiyonları, vb) kullanarak sonuçları sıralama uzunluklarını ölçen Spearman korelasyon katsayısı aracılığı ile birleştirdiğinde, başarımın ortalama %30-35'lik arttığını rapor etmiştir. Benzer bir çalışma, N.J. Belkin ve P. Kantor tarafından aşağıdaki şekilde gerçekleştirildi: (1) aynı bilgi ihtiyacı ile ilişkili olarak birden fazla P-Norm sorgu ifadeleri (Salton, Fox ve Wu, 1983) oluşturuldu (2) INQUERY bilgi erişim sisteminin (Turtle ve Croft, 1991) ağırlıklı ve ağırlıksız delil (arama terimleri) birleştirme özelliği kullanılarak tek bir sorgu ifadesi elde edildi. Sistem başarımının bu şekilde %40 civarında artırıldığı bir önceki çalışmada rapor edilmiştir.

[8] Bir erişim çıktısının kalitesini belirleyen iki metrik, daha önce tanımladığımız, anma ve duyarlık değerleridir. Böylece, anma ve duyarlık değerlerinin yükseltilmesi erişim çıktısının kalitesinin yükseltilmesi ile aynı anlama gelmektedir.

Şekil 1. Bir bilgi erişim sisteminin işlevsel mimarisi

Bu bölumde arka yüz kavramlarını teşkil eden belgeler, içerik belirteçleri ve sorgular üç alt başlık halinde incelenmekte, eşleştirme süreci (ya da daha bilinen adıyla erişim fonksiyonları) ve etkinlik ölçümleri tartışılmaktadır.

2.1 İçerik Belirteçleri

İçerik belirteci bir belgenin veya bilgi ihtiyacının gösterimi (temsil edilmesi) için kullanılır. Rasgele işlenen metinler üzerinden aynı alan (domain) içinde olsalar bile ortak yapı kalıpları elde edebilmek çoğunlukla mümkün değildir. Zaten işlenen veri (örneğin, metin) üzerinde bir yapı empoze edilebiliyorsa, bu süreç, doğasına göre veri tabanı veya uzman modeller aracılığı ile daha etkin olarak herhangi bir bilgi erişim modeline gerek kalmaksızın modellenebilir. Ele alınan bir metin, (bütünlük arz eden) bir bilgi taşıdığı için, buradaki kritik soru bu metnin veya belgenin içerik açısından nasıl temsil edileceğidir. Çünkü gerek duyulduğunda bu belgeye erişilebilmelidir. Başka bir deyişle, belge işleme sürecine yakından bakıldığında, belgeleri çoğu zaman sisteme sunuldukları halleri ile değil, belgelerin içeriğini yansıtan belirteç kümesi (surrogate record) halinde kullanma zorunluluğu görülür. Bu içerik belirteçlerine anahtar sözcük, üst veri (metadata), dizin terimi, tanımlayıcı, veya kısaca terim gibi adlar verilir.

1950'lerin sonunda bir metnin konusunu belirten sözcükleri (metindeki geçiş sıklıklarına dayanarak) belirlemeye yarayan bir program geliştiren Hans Peter Luhn, anahtar sözcüklerle dizinleme ve arama yapmanın modern "kaşif"i olarak bilinmektedir. Luhn ilk kez bir makale adında geçen sözcükleri, her bir sözcüğün basılı dizinlerde "giriş" (entry) olarak yer almasını algoritmik olarak sağlayan bilgisayarla dizinlemeyi geliştirmiştir. KWIC (Key-Word-in-Context) olarak bilinen bu dizinleme türü bibliyografik dizinlerin hazırlanmasında halen kullanılmaktadır (Svenonius, 2000, s. 28, 44, 190).

Her bir dizin terimi belgelerin içeriğini çoğu zaman bütünüyle değil, ancak bir yönüyle ifade eder ve bir belge için bir çok dizin terimi seçilir. Verilen bir belge için dizin terimlerinin seçilmesi sürecine *dizinleme* adı verilir. Dizinleme süreci kontrollu veya kontrol edilmeyen bir terim sözlüğü (vocabulary) üzerinden elle (manual) ya da otomatik olarak gerçekleştirilebilir. Kontrollü dizinlemede bir belgeyi temsil edecek terimlerin seçimi belli bir konu sözlüğü temel alınarak konu uzmanlarınca yapılır. Bu tarz ile yüksek bir biçimdeşlik (uniformity) ve kalite elde etmek mümkündür; fakat dizinlemenin yavaş ve maliyetli olması ve en önemlisi kullanıcının sorgu ifade etmede kullanacağı kelime dağarcığının kontrollü sözlükle çakışması gereksinimi kontrollü dizinlemenin dezavantajlarından birisidir.[9] Kullanıcılar konu uzmanları tarafından kullanılan kontrollü sözlüklerle (örneğin, Kongre Kütüphanesi Konu Başlıkları Listesi) aşina değildirler (Tonta, 1990). Dahası, konu uzmanları tarafından aynı kontrollü sözlüğün kullanıldığı durumlarda bile dizinleme tutarlılığı (indexing consistency) son derece düşük olmaktadır (Tonta, 1991). Yapılan deneysel araştırmalar otomatik dizinlemenin kontrollü dizinleme ile elde edilen performansı yakaladığını göstermiştir (Van Rijsbergen, 1979; Salton, 1989).

Metin yazarının sözcük dağarcığı ile bir bilgi erişim sistemi kullanıcısının sözcük dağarcığı arasındaki farka *sözcük dağarcığı farkı* diyelim. Kullanımı daha pratik ve hızlı olan otomatik dizinleme sözcük dağarcığı farkının açılmasına yol açar. Sözcük dağarcıkları arasındaki fark, belge derlemindeki bir belgenin verilen bir sorgu ifadesi ile ilgili olmasına (ya da daha da kısıtlayacak olursak her ikisinin de aynı kavrama karşılık geldiğini

[9] Aslına bakılırsa, bu sorun, belgelerin tam metinlerinde geçen her sözcüğün dizinlendiği, kontrol edilmeyen bir terim sözlüğü kullanıldığı zaman da tam olarak ortadan kalkmamaktadır (Tonta, 1995). Kullanıcıların bilgi ihtiyaçlarını ifade etmek için kullandıkları terimlerle ilgili belgelerin tam metinlerinde geçen terimler arasındaki çakışma sorgu cümlelerinde geçen terim sayısı arttıkça hızla düşmektedir. Kullanıcılar, hakkında bilgi bulmak istedikleri konuları sorgu cümlelerinde iyi tanımlayamamaktadırlar. Zaten tam olarak ne aradıklarını tanımlayabilselerdi belki de ilgili bilgi erişim sistemini kullanmalarına gerek kalmayacaktı. Bilgi erişimin temel paradoksu "hakkında bilgi bulmak için bilmediğin bir şeyi tanımlama gereği"dir. Bu paradoks, bir bakıma, "sözlük" sözcüğünün anlamını bilmeyen (ve yakınında sorabileceği birisi olmayan) bir kimsenin çaresizliğine benzetilebilir (Blair ve Maron, 1985).

varsaymamıza) rağmen, söz konusu belgenin çoğunlukla elimizdeki sorgu ifadesi ile eşleşmemesine (ya da eşleştirme derecesinin düşük olmasına) yol açar.

Bilgi erişim sistemlerinde dağarcık farkını kapatmak için kullanılan araçlardan birisi de *gömü*lerdir (thesauri). Tipik bir bilgi erişim sistemi için gömü, terimlerin belli bir ilişkiye göre düzenlenmesidir (Srinivassan, 1992). Gömü, dizinleme ve erişim hizmetlerinde terimlerin kullanımına rehberlik eder. Bilgi erişim sisteminin sorgu işleme sürecinde yardımcı yapı olarak, satırda belgeleri ve sütunda (dizin) terimleri tuttuğunu varsayalım (ki bu uygulama oldukça sık kullanılan ve "ters dizin" kütüğü olarak adlandırılan bir yapıdır). O zaman, eş anlamlılar ilişkisinin karşılıklı dışlayan kümeler olduğunu varsayarak, gömünun dizinlemede kullanımı sütunların belirli bir ad altında birleştirilmesinden ibarettir. Erişimde ise, eş anlamlılar ilişkisinin kesişen kümeler olduğunu varsayarak, gömü sorgu genişletmeye karşılık gelir. Eş anlamlı olan her bir terim için verilen bir sorgu belge uzayına karşı eşleştirilir.

Gömüler, elle ve otomatik olmak üzere iki türlü üretilirler. Elle gömü üretimi insan emeği ile terimler arasında önceden ilişki (eş/zıt anlamlı ilişkiler, dar/geniş terimler, vd.) kurulmasına ve bu ilişkilerin gömü oluşturmak için kullanılmasına dayanır. Gömü sıradüzensel (hiyerarşik) ilişkiler şeklinde de oluşturulabilir. Örneğin, '*A*' ve '*B*' herhangi iki eş anlamlı küme olsun. '*A*' terimi '*B*' teriminden daha dar (narrower) anlamlıdır, ya da '*B*' terimi '*A*' teriminden daha geniş (broader) anlamlıdır demek, matematiksel olarak '*A*'nın '*B*'nin alt kümesi (ya da tersi) olduğunu belirtmektir. Bu tür tek yönlü ilişkiler birbirleriyle aşağıdan (dar anlamlıdan) yukarı (geniş anlamlılar) doğru bağlandıklarında sıradüzensel ilişki oluşturulur. Sıradüzensel ilişkiler gömü işlevinin yanı sıra, sorgu sonuçlarını süzmede de kullanılırlar. Bu tür ilişkilerle oluşturulan bilgi erişim sistemleri kavram tabanlı sistemler olarak da adlandırılmaktadır (McCune, Tong, Dean ve Shapiro, 1985).[10]

Otomatik gömü üretimi teknikleri, terimlerin herhangi bir belgede birlikte geçme olasılıklarını temel alır. Herhangi iki terimin aynı gömü alt kümesi içerisinde yer alması onların anlamsal olarak eş anlamlı olduğu anlamına gelmez; ilgili iki terimin aynı küme içerisinde yer alması demek, yalnızca ve yalnızca sistemin verilen derlem bazında bu iki terimi istatistiksel olarak biribirinden ayırt edememesi demektir. Gömü yapısının bir bilgi erişim sisteminin etkinliğine (effectiveness) olan katkısı üzerinde yapılan çalışmalarda gömünün üretildiği derleme benzer derlemlerde kullanılması şartıyla anma değerinde

[10] Ali Alsaffar ve ötekilerinin çalışmaları kavram tabanlı sistemlerin bir sürekli (persistent) sıradüzensel ilişki tutmadan Boole (Alsaffar, Deogun, Raghavan ve Sever, 1999) veya vektör (Alsaffar, Deogun, Raghavan ve Sever, 2000) tabanlı sistemlerin üstüne etkin olarak nasıl kurulabileceği açısından ilginçtir.

%20'lere yaklaşan artışlar elde edilebildiği görülmüştür (Salton, 1989; Crouch ve Yang, 1992; Chen ve Lynch, 1992). Türkçede ise, alanı çok farklı konuları içeren küçük bir derlemde, çeşitli parametrelere göre üretilen gömüler için yapılan performans araştırmasında, gömü kullanımının erişilen ilgili belge sayısını artırmazken, ilgili belgeleri erişim çıktısında üst sıralara yerleştirdiği görülmüştür (Sezer, 1999).

2.2 Belgeler

Tipik bir bilgi erişim sisteminde belgeler terimler ile gösterilir. Verilen bir derlem bağlamında terim sözlüğü geleneksel olarak aşağıdaki gibi gerçekleştirilebilir: (1) harf olmayan karakterler boşluklarla yer değiştirilir; (2) tek harfli sözcükler silinir; (3) bütün karakterler küçük harfli yapılır; (4) durma listesinde adı geçen sözcükler silinir; (5) sözcükler gövdelenir (stemming); (6) tek karakterli gövdeler atılır. Son adım olarak, istenirse, (6). adımın sonunda elde edilen listedeki yüksek sıklıklı[11] sözcükler terim sözlüğünden çıkarılarak derleme duyarlı ikinci bir durma listesi oluşturulur. Ya da, yüksek sıklıklı sözcükler, otomatik eş anlamlı sözlük oluşturmanın bir parçası olarak, orta sıklıklı sözcüklerle birleştirilerek tamlama (phrase) oluştururlar.[12] Türkçe gibi sondan eklemeli (agglunative) dillerde gövdelemenin (bir sözcükten çekim eklerinin atılıp, yapım eklerinin korunması) bilgi erişim sistemi içindeki önemi yadsınamaz. Nitekim GÖVDEBUL algoritması (Duran, 1999) kullanılarak yapılan deneylerde anma ve duyarlık değerlerinde gövdeleme yapmaksızın yapılan sorgulara göre sırasıyla ortalama %20 ve %25 artış gözlenmiştir (Sezer, 1999). Bu deneylerde Türkçeye yerelleştirilen SMART sistemi kullanılmıştır (http://ata.cs.hun.edu.tr/~km/arsiv.html).

Otomatik olarak elde edilen gövdelenmiş sözcüklere "terim" denir. Daha önce de belirtildiği gibi terimler hem belgeleri göstermede hem de sorguları ifade etmede kullanılırlar. Bu ikisi arasında bir ayrım yapmak istediğimizde öncekine belge terimleri ve diğerine de sorgu terimleri adını vereceğiz. Bir belge teriminin ağırlığı terim belge içinde yer alıyorsa bir, aksi takdirde sıfırdır (ikili ağırlık). Bu yaklaşıma Boole modeli adı verilir. Diğer bir popüler yaklaşım ise vektör tabanlı[13] modellerde kullanılan *tf*idf* değerleridir. Burada, (**tf***)* terimin

[11] Bir sözcüğün sıklığı, ilgili sözcüğü taşıyan belgelerin derlem içindeki sayılarına eşittir.

[12] Tamlamaya katılan orta sıklıklı sözcük kendi başına terim sözlüğünde de yer alır.

[13] Vektör uzayı modelinde sorgular ve belgeler terim vektörleri biçiminde ele alınır. '*t*' tane ayrık terimin olduğu bir derlemde, *i.* belge,

$D_i = (a_{i1}, a_{i2}, \ldots a_{it})$,

ilgili belgede geçme sıklığı, yani *terim sıklığı*dır (term frequency). Terimin derlemde geçtiği belge sayısına ise *belge sıklığı* (document frequency) **(df)** denir. Terim sıklığı yüksek olan bir terim aynı zamanda derlem içindeki diğer belgelerde de sık geçiyorsa, ilgili terimin ayırt edici özelliği veya belge içindeki diğer terimlere göre göreceli değeri düşük olmalıdır. Bir terimin terim sıklığı (yani ilgili bir belgede geçme sıklığı) yüksek ve derlemdeki diğer belgelerde geçme sıklığı düşükse, o terimin göreceli ağırlığı yüksek olmalıdır. Bu kıstası sağlamak için *devrik belge sıklığı (inverse document frequency)* **(idf)** kullanılmıştır. "*idf*" parametresi terimin belge sıklığı arttıkça azalan özelliktedir. Tipik bir *idf* parametresi *log(N/df$_j$)*'dir. Burada *N*, derlemdeki toplam belge sayısı; *df$_j$*, *j*. terimin belge sıklığıdır. t_j teriminin D_i belgesi için ağırlığı w_{ij} ile gösterilirse, w_{ij}

$$wij=tf_{ij}*log(N/df_j) \qquad (1)$$

formülü ile hesaplanır. Yukarıda df_j , t_j teriminin belge sıklığı; tf_{ij}, t_j *teriminin* D_i belgesinde geçme sıklığı (terim sıklığı) ve *N* derlemdeki toplam belge sayısıdır.[14]

Terimler birbirleri ile belirli bir ilişki altında kümelendiği gibi belgeler de kümelere (clusters) bölünebilirler. Buradaki ideal amaç ise, belge arama uzayını, anma değerini sabit tutarak, küçültmektir. Belgeleri kümeleme süreci, belgeler birbiri ile karşılaştırılıp benzer bulunanların kümelenmesi ile en alt düzeyde başlar. Daha sonra kümeler birbiri ile karşılaştırılarak bir üst seviyede kümelenir. Bu işlem, tek bir küme kalana dek sürer. Oluşan yapıda sorgu en üst düzeyden başlayarak kümelerle karşılaştırılmaya başlanır ve en ilgili bulunan küme yönünde ilerlenir. Literatürde bu işleme *sıradüzensel kümeleme* (hierarchical clustering) denir (Van Rijsbergen, 1979). Bu yaklaşım arama motorlarının büyük bir çoğunluğunca 'directory search' (rehber arama) adı altında sağlanmaktadır. Kavram tabanlı bir arama motoru olan Excite'da (http://www.excite.com) ise, rehber aramaya ek olarak, geniş (broad) arama sonuçları düz (flat) olarak kümelendirilerek kullanıcıya sunulmaktadır. Böylece

---

ve *j*. sorgu ,

$Q_j=(q_{j1},q_{j2},...,q_{jt})$

biçiminde gösterilir. Burada a_{ik} ve q_{jk} sırasıyla, *k* teriminin D_i belgesi ve Q_j sorgusu içindeki göreceli ağırlıklarıdır.

[14] *"tf*idf"* metodunda terimlerin göreceli ağırlıkları önem taşır. *tf*idf* metodu ile birlikte diğer terim ağırlıklarını tartışan ve bu terimlerin karşılaştırmalı etkinliğini gösteren çalışmalara da rastlanmaktadır (Salton ve Buckley, 1988).

kullanıcı sorgusunu daraltmada ya da 'rafine' etmede hazır bloklardan biri veya birkaçıyla işleme devam ederek bilgi ihtiyacını istenilen düzeyde tatmin edebilmektedir.[15]

2.3 Sorgular

Bir sorgu, kullanıcının bilgi ihtiyacının resmi (formal) olarak belirtilmesidir. Kullanıcı çok değişik biçimlerde bir sorguyu ifade edebilir.

Arama terimleri (ya da sözcükleri) Boole işleçleri ile bağlanır (Salton, 1989; Van Rijsbergen, 1979). Boole işleçleri *ve (and), ya da (or)* ve *değil (and not)'*dir. 'Ve' işleci ile bağlanan terimlerin hepsini içeren belgeler, 'ya da' işleci ile bağlanan terimlerden en az birini içeren belgeler, 'değil' işleci ile bağlanan terimi içermeyen belgeler erişim çıktısında yer alabilirler.

Kullanıcı doğal dil ile sorgu ihtiyacını belirleyebilir. İlgili sorgu metni, Bölüm 2.2'de adımları verilen tipik bir dizinleme sürecinde olduğu gibi, arama terimleri sorgu vektörüne çevrilir. Sorgu vektörü ağırlıklandırılmış arama terimlerini (örneğin, *tf*idf* kullanılarak) içerebileceği gibi, ağırlıkların değişimini basit bir şekilde ikili değerler kümesi ile sınırlandırabilir (bir arama terimi ilgili sorgu vektöründe ya vardır ya da yoktur, fakat her ikisi olamaz). Doğal dilde girilen sorgularda ise terimlerin tamamının aynı belgede bulunma şartı yoktur. Belgenin, kullanıcının bilgi ihtiyacı ile ilgili olma derecesi, sorgu terimlerinin ne kadarını içerdiği ile doğru orantılıdır. Dolayısıyla sorguda geçen terimlerin tamamını içeren bir belge bu açıdan en iyi belgedir. Ancak bir belgenin erişim çıktısında yer alması için sorgu cümlesinde geçen tüm terimleri içermesi gerekmeyebilir. Kullanıcı tarafından verilen bir eşik değerini (threshold) aşan belgeler de erişim çıktısında yer alabilir. Başka bir deyişle, örneğin, kullanıcı bilgi ihtiyacına %80 veya daha fazla benzerlik gösteren belgeleri görmek isteyebilir.

Olasılık modeli arama terimlerini, geribildirim aracılığı ile ilgili belgelerde bulunabilme olasılıklarını temel alarak ağırlıklandırır; belge terimleri ise ikili ağırlığa sahiptirler (Robertson ve Jones, 1976; Crestani, Lalmas, Van Rijsbergen ve Campbell, 1998). Bu modelde, sorgu başlangıçta arama sözcüklerinin bir listesi olarak ya da doğal dilde ifade edilir. Sistem tarafından döndürülen belge çıktısının kullanıcının bilgi ihtiyacını

[15] Bu tür kullanıcı arayüzleri ile ilgilenen okuyucuya 'Light House' (http://www.lighthouse.org) aracını salık verebiliriz (Leuski, 2001). Bu araç, bir arama motoru tarafından döndürülen belgeleri iki boyutlu kümelendirerek (başka bir deyişle sınıflandırarak veya gruplandırarak) kullanıcıya grup etiketleri ile birlikte sunmaktadır.

karşılamaktan uzak olduğu durumlarda, kullanıcı geribildirim sürecini başlatarak daha kaliteli bir belge çıktısı elde etmek isteyebilir. Bu sürece *geribildirim* süreci denir (Salton ve Buckley, 1990). Geribildirim sürecinde, kullanıcı erişim çıktısındaki belgeleri çeşitli ilgililik düzeylerine göre sınıflandırır. Bu sınıflandırma temel alınarak, yapılan sınıflandırma hatası düzeltilmeye (daha doğrusu azaltılmaya) çalışılır. En basit ve en çok kullanılan sınıflandırma düzeyi, ilgili ve ilgisiz olmak üzere ikilidir (çok düzeyli geribildirim için bkz. Wong, Ziarko, Raghavan ve Wong (1989); Bollmann-Sdorra, Raghavan ve Sever (1999)). Hangi teknik uygulanırsa uygulansın, sınıflandırıcılar (classifiers), pozitif ve negatif örnekleri içeren belirli bir sıralı belge kümesi (erişim çıktısı) üzerinden eğitilirler (tümevarım süreci). Anma ve duyarlık değerleri açısından daha kaliteli olacağı varsayılan yeni bir erişim çıktısı ise arama sözcüklerinin yeniden ağırlıklandırılmasıyla elde edilir (tümdengelim süreci) (Wong ve Yao, 1990). [16] Eğitim aşamasında kullanıcı tarafından sisteme sunulan bilgiler kullanılarak, sorgu ifadesi içinde yer alan bir arama terimi eldeki belgede yer alıyorsa, belgenin ilgili olabilme olasılığı Bayes modeli (Duda ve Hart, 1973) üzerinde birtakim varsayımlar[17] yapılarak hesaplanır. Bu olasılık değeri arama teriminin yeni ağırlığını oluşturur.

Kavram tabanlı modeller ise kullanıcının bilgi ihtiyacını kurallar biçiminde ifade eder (Alsaffar et al., 2000, 1999; McCune et al., 1985). Ana kavramın alt kavramları bir üst kavramı oluştururken birbirleri ile 've' işleci ile bağlanabileceği gibi 'veya' işleci ile de bağlanabilir (örneğin, eğer belge (<kavram_1> ve <kavram_2>) veya <kavram_3>) içeriyorsa o zaman <ana kavram> belgede geçiyor demektir). Bir alt kavram, diğer bir üst kavramı belirli bir inanç derecesiyle belirleyebilir (Alsaffar et al., 2000). Bu yönüyle arama terimleri, yani belgede yazılı (literal) olarak yer alması istenen somut kavramlar) kullanıcı tarafından ağırlıklandırılabilir. Kavram, vektör, ve Boole tabanlı modeller arasındaki köprü P-Norm cümlecikleri ile kurulabilir (Alsaffar et al., 2000; Salton et al., 1983; Akal, 2000). Ayrıca vektör modeli içinde Boole modeli sorgu dilinin kullanılması konusundaki ilginç bir yaklaşım için okuyucu (Wong et al., 1989) no'lu analitik çalışmayı gözden geçirebilir.

2.4 Erişim Fonksiyonları

[16] Göz önünden kaçırılmaması gereken husus, geri bildirim sürecinin erişim modelinden bağımsız olup herhangi birine takılabılır (plug-in) olmasıdır.

[17] İkili bağımsız modeli içinde tanımlı bu varsayımlar aşağıdaki gibidir: (1) terimlerin ilgili belgelerdeki ve ilgisiz belgelerdeki dağılımı birbirinden bağımsızdır (2) belge terimleri ikili değere sahiptirler (Salton, 1989; Van Rijsbergen, 1979; Crestani et al., 1998).

Sorgu cümlesindeki terimlerle dizin terimleri arasında eşleşme olup olmadığı çeşitli erişim fonksiyonları kullanılarak belirlenebilir. Blair (1990) 12 değişik erişim fonksiyonunu ayrıntılı olarak incelemektedir.[18] Bu fonksiyonlar kabaca üç grup olarak sınıflandırılabilir:

1) Sorgu ve dizin terimlerinin *n*-boyutlu bir uzaydaki vektörler olarak işlem gördüğü ve ağırlıklandırıldığı vektör uzayı erişim fonksiyonu;
2) Sorgu ve dizin terimleri arasında kesin eşleşme (exact match) gerektiren erişim fonksiyonları/Boole erişim fonksiyonları; ve
3) Sorgu ve dizin terimlerinin olasılık kuramına göre ağırlıklandırılmasına dayalı erişim fonksiyonları.

Aşağıda söz konusu üç gruptaki erişim fonksiyonlarının resmi tanımları verilmektedir.

Daha önce bir bilgi erişim sisteminde üç ana nesne kümesi olduğunu söylemiştik. Bunlar sırasıyla, içerik belirteçleri (veya kısaca terimler), belgeler ve sorgulardır. Terimler hem sorguları hem de belgeleri göstermede kullanıldığı için, vektör uzayı modelinde pratik olarak sorgular ve belgeler terim uzayında bir nokta olarak görülebilir (ve bu varsayım sıkça yapılır).[19] Bu yaklaşımda her iki noktadan geçen ayrık (distinct) iki vektör (belge vektörü ve sorgu vektörü) düşünülür. Bu iki vektörün vektörel çarpımı -ki iki vektör arasındaki açının kosinüsüne eşit olduğundan kosinüs katsayısı olarak da bilinir- ya da skalar çarpımı -iç çarpım katsayısı olarak da bilinir- sorgu-belge noktaları arasındaki benzerliğin derecesini verebilir. Bu katsayılar aşağıda verilmiştir:

$$\text{İç Çarpımı } (D_r, Q_s) = \sum^t a_{ri} * q_{si} \qquad (2)$$

$$\text{Vektör Çarpımı } (D_r, Q_s) = (\sum^t a_{ri} * q_{si}) / (\sum^t (a_{ri})^2 * \sum^t (q_{si})^2)^{1/2} \qquad (3)$$

Formüllerde D_r belge vektörünü, Q_s sorgu vektörünü, a_{ri} ve q_{si} ise i. öğenin, sırasıyla, belge vektörü D_r ve sorgu vektörü Q_s'teki ağırlıklarını temsil etmektedir.

Boole modelinde bir belge veya sorgu, terimler kümesinin bir alt kümesi olarak düşünebilir. Bu durumda, iki küme (sorgu-belge) arasındaki eşleştirmelerin derecesi erişim fonksiyonunun değerini oluşturur. Örneğin, Jaccard katsayısı eldeki iki küme ($D_r=\{d_{r1}, d_{r2}, \ldots, d_{rt}\}$ ve $Q_s=\{q_{s1}, q_{s2}, \ldots, q_{st}\}$) arasındaki kesişimin oranını verir. Diğer yandan Dice katsayısı ise D_r ve Q_s kümeleri arasındaki kesişimi onların ortalama büyüklükleriyle ilişkilendirir. Aşağıda her iki katsayının resmi tanımları verilmiştir:

---

[18] Blair'in kapsamlı olarak incelediği erişim fonksiyonlarının kısa bir özeti için bkz. (Tonta, 1995).

[19] Terim uzayı kullanılarak yapılan modellemede [belgelerin ve sorguların gösterimi, karşılıklı (sorgu-belge) ve kendi içlerindeki (belge-belge, sorgu-sorgu) ilişkiler] olası paradoks durumlar Bollmann-Sdorra ve Raghavan'ın (1993) ilginç analitik çalışmasında daha ayrıntılı olarak incelenmektedir.

Jaccard Katsayısı $(D_r, Q_s) = |(D_r \times Q_s)| / |(D_r + Q_s)|$ *(4)*

Dice Katsayısı $(D_r, Q_s) = 2*|(D_r \times Q_s)| / (|D_r| + |Q_s|)$ *(5)*

Olasılık modelinde ise, daha önce de belirtildiği üzere, sorgu terimleri, geribildirim aracılığı ile ilgili belgelerde bulunabilme olasılıkları temel alınarak ağırlıklandırılır; belge terimleri ise genellikle ikili ağırlıklandırılır. Terimlerin ilgili belgelerde ve ilgisiz belgelerde dağılımının birbirinden bağımsız olduğunu varsayalım.[20] Daha ileri giderek, herhangi bir t_i belge terim değişkeni için aşağıdaki koşullu öncel (a priori) olasılıkları göz önünde bulunduralım:

$p_{ri}=(a_{ri}=1:$ *ilgili*$(Q_s))$ ve

$q_{ri}=(a_{ri}=0:$ *ilgisiz*$(Q_s))$.

Burada *ilgili*(Q_s) ve *ilgisiz*(Q_s) verilen bir Q_s sorgu ifadesi için sırasıyla ilgili ve ilgisiz belgeleri döndüren fonksiyonlar olsun. O zaman, kolayca görüleceği gibi, p_i eldeki belgenin ilgili olması halinde t_i'nin 1 olma olasılığını ve q_i eldeki belgenin ilgisiz olması durumunda t_i'nin 0 olma olasılığını verir. Aşağıdaki olasılık erişim fonksiyonu (eldeki Q_s sorgusuna göre derlem içindeki D_s belgesinin erişim değeri) kullanıldığında, sistemin hata yapma olasılığının en aza indirgendiği ve bu anlamda optimal olduğu ispatlanmıştır (Robertson ve Jones, 1976; Crestani et al., 1998):

Olasılık Erişim Fonksiyonu $(D_r:Q_s)$: $\sum t_i \log((p_i*(1-q_i))/(q_i*(1-p_i)))$. *(6)*

Yukarıdaki p_i ve q_i değerleri Q_s sorgusu için döndürülen erişim çıktısı üzerindeki kullanıcı değerlendirmeleri kullanılarak tahmin edilir. Ancak geribildirim üzerinden öncel olasılık değerlerini (p_i ve q_i) tahmin etmek pratik değildir.[21]

---

[20] İkil bağımsız erişim modelinde (IBEM) (Robertson ve Jones, 1976) göz önünde bulundurulan terimlerin (ilgili ve ilgisiz) belgeler içindeki dağılımının birbirlerinden bağımsız olduğu varsayımı, gerçeği yansıtmayan bir varsayım olduğu gerekçesi ile devamlı şekilde eleştirilmiştir. Bununla birlikte, Cooper (1995) yukarda verilen varsayıma aslında IBEM'de ihtiyaç duyulmadığını ve onun daha güçsüz versiyonu olan 'sıralı bağımlılık' varsayımının yeterli olacağına işaret etmiştir. Sıralı bağımlılık (linked dependence) kısaca aşağıdaki gibi açıklanabilir: bir belgenin ilgili ve ilgisiz sınıfllarda olma olasılıklarının oranı onu oluşturan terimlerin ilgili ve ilgisiz sınıflarda olma olasılık oranlarının tek tek çarpımına eşittir.

[21] Tahmin için kullanılan diğer yöntemler hakkında Yu ve Lee'nin (1986) çalışmasına; belge terimlerinin ikil değerler taşıması yerine kesikli değerler taşıması durumunda olasılık erişim fonksiyonu oluşturmadaki yaklaşım için sırasıyla Yu ve Lee'nin (1986) ve Bollmann-Sdorra ve diğerlerinin (1999) çalışmalarına bakılabilir.

Son olarak, erişim fonksiyonlarının her bir döndürülen belgeyi kesikli değerlerle ilişkilendirmesinin avantajlarını da sıralamakta yarar görüyoruz:

- Çıktıda döndürülen belgeler en benzer belge en üstte olacak şekilde sıralanabilir;
- En benzer belgeler ilk dönen belgeler olduğu için kullanıcıya en iyi '*n*' belge döndürülerek duyarlılık değeri artırılabilir;
- Erişimde en iyi dönen belge kullanıcıya danışılmaksızın direkt geribildirim olarak kullanılabilir.

2.5 Etkinlik

Bilgi erişim sistemlerinin etkinliği tipik olarak *anma*, *duyarlık* ve *posa* (ya da yanlış alarm) ölçütleri ile ölçülür. Bu ölçütlerin hesaplanmasında Tablo 1'de gösterilen ikili sınıflama tablosu kullanılır. Bu tablo her bir sorgu için oluşturulur. İlgili tablonun başlığında 'ikili sınıflama' tamlamasının olmasının nedeni, sistemin bilgi erişim sürecindeki tipik davranışının bir ikili sınıflama örneği göstermesidir (eldeki sorgu ile eşleştirilen belge ya ilgilidir ya da ilgisizdir). İkili sınıflama tablosunda her bir hücre ilgili satır ve sütunun kesişimini gösterir. Örneğin, '*a*' sistem tarafından erişilen ve kullanıcının ilgili (relevant) bulduğu belge sayısını, '*b*' sistem tarafından erişilen ancak kullanıcının ilgisiz bulduğu ("false drops") belge sayısını, '*a*+*b*' ilgili ya da ilgisiz erişilen toplam belge sayısını, '*a*+*c*' ise bir sorguya karşılık erişilen ya da erişilemeyen derlemdeki toplam ilgili belge sayısını verir. Çeşitli ölçütlere veya hedeflere göre farklı etkinlik ölçütleri bu tabloya dayanılarak çıkarılabilir. Burada çok iyi bilinen anma, duyarlık ve posa değerlerine yer verilecektir. *Anma*, kimi zaman *hedefi vurma oranı* olarak da adlandırılır, sistem tarafından erişilen ilgili belgelerin (*a*) derlemdeki toplam ilgili belgelere (*a*+*c*) oranını verir.[22] *Duyarlık*, sistem tarafından erişilen ilgili belgelerin (*a*) erişim çıktısında yer alan (ilgili ve ilgisiz) toplam belgelere (*a*+*b*) oranını verir.[23] Anma ve duyarlık değerleri 0 ile 1 arasında değişmektedir. Anma ve duyarlık değerleri ne kadar yüksek olursa bir bilgi erişim sisteminin etkinliğinin de o kadar yüksek olduğu kabul edilmektedir (Salton, 1989). Posa ise, sistem tarafından ilgili olduğu varsayılıp erişilen (*b*) fakat gerçekte ilgisiz olan belgelerin toplam ilgisiz belgelere (*b*+*d*) oranını verir.[24] Bu oran "bir sistemin ilgisiz belgeleri ne derece sağlıklı olarak reddettiğini ölçer" (Blair, 1990, s. 116).

---

[22] Döndürülen/erişilen belgenin ilgili olduğu verildiğinde erişim çıktısına dahil edilmesinin olasılığı, Pr(P→R), anma değeri ile tahmin edilir.

[23] Erişilen belgenin erişim çıktısına dahil edildiği bilgisi verildiğinde, belgenin ilgili olma olasılığı, Pr(R→P), duyarlık değeri ile tahmin edilir.

[24] Erişilen belgenin ilgisiz olduğu bilgisi verildiğinde, belgenin erişim çıktısına dahil edilmesi olasılığı, Pr(¬P→R), posa değeri ile tahmin edilir. Arama motorlarında (ya da genelde derlemdeki belge sayısının

Tablo 1. İkili Sınıflama tablosu

	İlgili (P)	İlgisiz (¬P)	
Erişilen (R)	a	b	a + b
Erişilemeyen (¬R)	c	d	c + d
	a + c	b + d	a + b + c+ d

Bir sistemin etkinliği çoğunlukla anma ve duyarlık değerleri ile ifade edilir.[25] Tabi bu değerler her bir sorgu bazında kesin değerler olabileceği gibi, belirli sayıdaki sorgular üzerinden mikro ya da makro ortalamalar alınarak da hesaplanabilir. Mikro ortalamada sayıların, makro ortalamada ise oranların aritmetik ortalaması alınır. Örneğin, bir arama motoruna iki soru yönelttiğimizi varsayalım. İlkinde, erişilen beş belgeden ikisi ilgili bulunsun, ikincisinde ise erişilen 10 belgeden birisi ilgili bulunsun. Bu iki soru için mikro ortalama yöntemi kullanılırsa ortalama duyarlık değeri %20 ((2+1)/(5+10)=3/15=0,2), makro ortalama yöntemi kullanılırsa %25 ((2/5)+(1/10)/2)=(0,4+0,1)/2=0,5/2=0,25) olarak bulunur. Mikro ortalama yöntemi belgelere, makro ortalama yöntemi sorgulara ağırlık verir. Bir başka deyişle, makro ortalama, sistemin tipik bir kullanıcı için tahmini değerini temsil ederken, mikro ortalama derlemde çok sayıda ilgili belge bulunan sorgulara gereğinden fazla ağırlık verir (Rocchio, 1971).

Blair'in (1990, s. 73-74) de vurguladığı gibi, bilgi erişim temelde bir deneme-yanılma süreci olduğundan, bilgi erişim sistemlerindeki belgelere erişmek için yapılan hemen hemen her aramada ilgili belgelerin yanı sıra değişen oranlarda ilgisiz belgelere de erişilmektedir. Ancak ideal bir bilgi erişim sistemi ilgili belgelerin tümüne ve salt ilgili belgelere erişim sağlar. Yukarıda açıklandığı üzere, duyarlık hesaplamasında, erişim çıktısında yer alan ilgili ve ilgisiz belge sayıları kullanılır; fakat kimi zaman sistemin aynı duyarlık değerine sahip erişim çıktıları arasından ilgili ve/veya önemli[26] olan belgeleri en iyi ön plana çıkaran erişim çıktısını seçmesi istenebilir (Kobayashi ve Takeda, 2000). Bu durumu aşağıdaki örnek (Tablo 2) ile açıklayalım.

---

yüksek olduğu bilgi erişim sistemlerinde) posa değerinin ölçüldüğü araştırmalara rastlanmamıştır. Çünkü yüz milyonlarca belge üzerinde arama yapılan Web ortamında posa değeri hemen hemen hep sıfır çıkacaktır.

[25] Anma, duyarlık ve yanlış alarm değerleri arasındaki ilişkiler için bkz. (Van Rijsbergen, 1979).

[26] Popüler olan belgelere bağlantı veren 'hub' sayfalara veya kendileri popüler olan sayfalara (authoritative) kısaca önemli sayfalar adını vermekteyiz.

Tablo 2. Normalize sıralama

Sıralama	1	2	3	4	5	6	7	8	9
EÇ1	+	+	+	+	+	-	-	-	-
EÇ2	-	-	-	-	+	+	+	+	+
EÇ3	+	+	+	-	-	-	+	-	+

Yukardaki tabloda '+' ve '-' sırasıyla ilgili ve ilgisiz belgeleri; EÇ1, EÇ2 ve EÇ3 aynı bilgi ihtiyacı için ifade edilen üç ayrı sorgu ifadesi ile ilişkili döndürülen erişim çıktıları olsunlar. Duyarlığı 'DK' ile gösterelim. O zaman, $DK_{EÇ1}=DK_{EÇ2}=DK_{EÇ3}=5/9$'dur; fakat sıralamalara göz attığımızda her üçünün farklı çıktılar olduğunu farkederiz (her üç erişim çıktısı erişim çıktı boyutunun sabitlendiği durumlarda tipik olarak ortaya çıkabilir).

Yukarıdaki tartışmanın önemli görüş noktalarından birisini, duyarlık değerleri aynı olmasına karşın kullanıcıların, ilgili belgelerin erişim çıktısında olabildiğince üst sıralarda yer aldığı arama sonuçlarını tercih etmeleri oluşturmaktadır. Çünkü kullanıcılar daha az çaba sarfederek ilgili belgelere eriştikleri arama sonuçlarının daha değerli olduğunu düşünmektedirler. Öte yandan, bir erişim çıktısında ilgisiz belgelerin en üst sıralarda yer aldığı, buna karşılık ilgili belgelerin çıktıda ya hiç yer almadığı ya da çıktının en sonunda listelendiği arama sonuçları kullanıcıların sabrını zorlayıp onları arama yapmaktan vazgeçirebilir. Bu metrik gözetilerek oluşturulan ölçüte "normalize sıralama" adı verilmektedir. Sıralama elde edilen erişim çıktısında en ilgili olduğu varsayılan belgenin ilk sırada, ilgililik derecelerine göre diğer belgelerin de izleyen sıralarda yer alması demektir. Normalize sıralama (S_{norm}) elde edilen erişim çıktılarındaki sıralamaya bağlı olarak bir bilgi erişim sisteminin etkinliğini ölçmektedir (Yao, 1995). Normalize sıralama değerinin hesaplanması için kullanılan formül aşağıda verilmektedir.

S_{norm}: $$S_{norm}(\Delta) = \frac{1}{2}\left(1 + \frac{S^+ - S^-}{S^+_{\max}}\right) \qquad (7)$$

Bu formülde:

Δ : erişim çıktısı sıralaması;

S^+ : erişim çıktısında ilgili belgelerin ilgisiz belgelerin önünde yer aldığı belge çiftleri sayısı;

S^- : erişim çıktısında ilgisiz belgelerin ilgili belgelerin önünde yer aldığı belge çiftleri sayısı; ve

S^+_{max} : mümkün olan en fazla S^+ sayısıdır.

Yukarıdaki örneğimize (S^+_{max} değerini 20 kabul ederek) devam edecek olursak:

S_{norm}(EÇ1)=1/2(1+(20-0)/20) = 1;

S_{norm}(EÇ2)=1/2(1+(0-20)/20) = 0; ve

S_{norm}(EÇ3)=1/2(1+(13-9)/20) = 0.6

değerlerini elde ederiz. . Bir başka deyişle, kullanıcının, duyarlık değerleri aynı olmasına karşın, normalize sıralama değerlerine bakarak bu üç arama sonucundan ilkini diğerlerine tercih edeceği kolayca söylenebilir.

Elde edilen değerlere dikkatle bakıldığında, normalize sıralama değerinin ilgisiz belgeleri başarılı bir şekilde reddetmeyen (yani "yanlış alarm" veren) bilgi erişim sistemlerini cezalandırdığı görülecektir. Normalize sıralama değerinin, bir bakıma, tüm ilgili belgelerin ve salt ilgili belgelerin erişim çıktısında yer aldığı "ideal erişim etkinliği" ile derlemdeki tüm ilgisiz belgelerin çıktının başında, ilgili belgelerin de çıktının en sonunda yer aldığı "en kötü erişim etkinliği"[27] arasındaki değerlere belirli bir anlam yüklemeye yaradığı söylenebilir.

Birkaç ilgili belgeye hızla erişim sağlamak isteyen kullanıcılar açısından normalize sıralama değeri önemli olabilir. Öte yandan, kapsamlı arama yapan kullanıcılar (örneğin, belli bir konuda yayımlanmış tüm belgelere erişmek isteyen kullanıcılar) ya da belli bir konuda daha önce herhangi bir belge yayımlanmadığını bilgi erişim sistemi aracılığıyla doğrulatmak isteyen kullanıcılar (örneğin, patent aramaları) normalize sıralama değerlerine itibar etmeyebilirler. Normalize sıralama değeri bir bilgi erişim sisteminin etkinliğini ölçmede tek başına bir ölçüt olarak sıklıkla kullanılmasa da, ilgili belgelere sürekli ilk sıralarda erişen bilgi erişim sistemlerinin diğerlerine göre performans yönünden daha etkin sistemler olduğunu kabul etmek gerekmektedir.

Bilgi erişim sistemlerinin etkinliğini ölçmede kullanılan "kapsama" ve "yenilik" oranlarından da kısaca söz etmekte yarar vardır. Kapsama oranı (coverage ratio), erişilen ve kullanıcının daha önceden ilgili olduğunu bildiği belge sayısının, ilgili olduğu bilinen toplam belge sayısına oranıdır. Yenilik oranı (novelty ratio) erişilen ve kullanıcının daha önce görmediği ilgili belgelerin erişilen ilgili belgelere oranıdır (Korfhage, 1997, s. 198). Kapsama ve yenilik oranlarını hesaplamak için aşağıdaki formüller kullanılır:

[27] Aslına bakılırsa, bilgi erişim sistemleri bir sorgu karşılığında derlemdeki tüm belgelere erişim sağlanmasına genellikle izin vermez.

$$\textit{Kapsama oranı} = |R_k| / U \qquad (8)$$

$$\textit{Yenilik oranı} = |R_u| / |R_u| + |R_k| \qquad (9)$$

Formüllerde *U,* kullanıcının daha önceden bildiği ilgili belgelerin setini, $|R_k|$, erişilen ve kullanıcının ilgili olduğunu önceden bildiği belge sayısını, $|R_u|$ ise erişilen ve kullanıcının daha önceden görmediği ilgili belgelerin sayısını ifade etmektedir.[28] Örneğin, kullanıcının, aradığı konuda toplam 15 ilgili belge (*U)* olduğunu bildiğini varsayalım. Sistem, kullanıcının sorusuna karşılık toplam 10 ilgili belgeye erişir ve bunlardan 4'ü ($|R_k|$), kullanıcının daha önceden bildiği belgeler olursa kapsama oranı 4/15 olur ($|R_k| / U$). Aynı örneği kullanacak olursak, erişilen ilgili belgeler arasında kullanıcının daha önceden görmediği 6 belge bulunmaktadır ($|R_u|$). Dolayısıyla yenilik oranı 6/10 olur (Korfhage, 1997, s. 198). Yüksek kapsama oranı sistemin, kullanıcının görmek istediği belgelerin çoğuna eriştiği, yüksek yenilik oranı ise sistemin, kullanıcının daha önceden bilmediği yeni belgelere eriştiği anlamına gelmektedir.

Kuşkusuz kullanıcı, gerçekte daha önceden bildiği belgelerle ilgili değildir. Kullanıcı açısından yüksek yenilik oranı tercih edilir (Korfhage, 1997, s. 198).

[28] Formül için bkz. http://home.himolde.no/~molka~/in350/c4y00.html.