Sözlük Kullanarak Türkçe için Kavram Madenciliği Metotları Geliştirme

Cem Rıfkı Aydın[1], Ali Erkan[1], Tunga Güngör[1], Hidayet Takçı[2]

[1] Boğaziçi Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul

[2] Cumhuriyet Üniversitesi, Bilgisayar Mühendisliği Bölümü, Sivas

cemrifkiaydin@gmail.com, alierkan@gmail.com,
gungort@boun.edu.tr, htakci@gmail.com

Özet: Kavram madenciliği yazınsal, görsel veya işitsel metinlerden anlamlı kavramlar çıkarma işlemine denir. Başta İngilizce olmak üzere yaygın olarak konuşulan Batı dillerinde bu alanda oldukça fazla sayıda çalışma yürütülmüş olsa da Türkçe'de şu ana kadar kavram madenciliği üzerine geliştirilmiş çok iyi başarı oranı veren bir çalışma yoktur. Bu çalışmada metinlerden kavram çıkarmak için yapay zeka algoritmalarının kullanımı yanında en çok yararlanılan, kelimelerin birbiriyle olan ilişkilerini hiyerarşik bir düzen içinde içeren veritabanı olan WordNet yerine şu ana kadar denenmemiş bir yol olarak TDK sözlüğü kullanılmıştır. Bir kelimenin sözlükteki kelime tanımı içerisinde yer alan kelimeler, o kelimenin kavramıyla alakalı sözcükler olabildiğinden anlamlı sonuçlar çıkarılmıştır. Başarı oranları, daha önce Türkçe'de kavram madenciliği üzerine geliştirilmiş çalışmaların verdiği sonuçlardan daha iyidir.

Anahtar Sözcükler: Kavram Madenciliği, Bağlamsal Analiz, Sözlüksek İlişkiler, Türkçe Derlem

Developing Concept Mining Methods in Turkish Using Dictionary

Abstract: Concept Mining is a process, through which expressive concepts are extracted from textual, visual, or audio artifacts. Although there have been developed many methodologies in this domain, mainly for English amongst many Western languages, there has been no work developed in Turkish so far in this domain, that has high success rates. In this work, instead of using WordNet, a lexical database which has synset relations defining the hierarchical relationships between words besides artificial intelligence methods, TDK dictionary is made use of, a novel approach. Since the words in this dictionary's word definition may be relevant to the concept of that word, expressive results could be achieved. Success rates for this work are seen to be higher than that which have been developed for concept mining domain in Turkish so far.

Keywords: Concept Mining, Context Analysis, Lexical Relationships, Turkish Corpora

1. Giriş

Kavram, bir kelimenin daha soyutsal ve genel anlamını ifade eden kelimeye denir. Kavramlar soyut veya somut kelimeler olabilir. İnsanlar algıladığı nesne veya olgunun kavramını kolayca çıkarabilmektedir; ama bilgisayarlar insan beyninin karmaşık ve üstün nöral algılama sistemine sahip olmadığı için kavram çıkarma bu makineler için daha zor olmaktadır. Bunun için makine öğrenme ve yapay zeka uygulamalarından faydalanılmaktadır.

Kavram madenciliği her ne kadar genel olarak yazınsal metinlerden anlamlı kavramlar çıkarma olarak tanımlansa da görsel ve işitsel metinlerden de kavram çıkarma işlemleri vardır. Bu çalışmada ise sadece yazınsal metinlerden kavram çıkarılmıştır. Kavram madenciliği zor; ama bir o kadar kullanımı ve yararlılığı fazla olan bir alandır. Örneğin tıp alanında hastaların ve hastalıkların sınıflandırılmasında yardımcı bir rol üstlenebilirken [1] [2], hukuksal alanda ise davaları sınıflandırmakta kullanılabilmektedir [3]. Diğer kullanım alanlarına örnek olarak açık uçlu anketlerin değerlendirilmesi, dokümanların sınıflandırılması ve müşteri profillerinin değerlendirilmesi verilebilir.

Şu ana kadar yabancı dillerde kavram madenciliği üzerine çoğunlukla WordNet sözlüğünden faydalanılmıştır. WordNet kelimelerine birbiriyle olan ilişkisini ifade eden *synset* adlı kümeleri barındıran bir veritabanıdır. Bu küme içinde eşanlamlılık (synonymy), zıt anlamlılık (antonymy), genel anlamlılık (hypernymy) gibi bir sürü ilişki barınmaktadır, kavram bir kelimenin daha soyutsal ve genel anlamını ifade ettiğinden, bu veritabanındaki *hypernymy* özelliğinden faydalanılmaktadır. Örneğin *kedi* kelimesinin WordNet veritabanında *hypernym*'i karşılığına bakarak bu kelimenin kavramı hayvan olarak belirlenebilir. Her ne kadar İngilizce'de WordNet veritabını oldukça gelişmiş olsa da, Türkçe için geliştirilmiş olan WordNet veritabanı oldukça eksik ve emekleme aşamasındadır. Bu yüzden bu çalışmada WordNet yerine TDK (Türk Dil Kurumu) sözlüğü kullanılmıştır.

TDK sözlüğü sözcüklerin anlam cümlelerini içermektedir ve bu anlam cümlesindeki kelimelerle sözcük arasında birçok ilişki bulunmaktadır. Bu ilişkiler arasında, aynı WordNet'te olduğu gibi eşanlamlılık, genel anlamlılık, nedensellik gibi özellikler bulunmaktadır. Bu ilişkiler sözcüğün kavramıyla alakalı olabileceğinden, sözlüğün kullanımı başarılı sonuçlar vermiştir. TDK sözlüğüyle yürütülen bu çalışmada kavram çıkarma algoritması, daha önce Türkçe'de bu alan üzerine yapılan çalışmalardan daha başarılı sonuçlar vermiştir.

Çalışmanın ikinci bölümünde literatür araştırması üzerine değinilmiş, kavram madenciliği üzerine yapılan

çalışmalardan kısaca bahsedilmiştir. Üçüncü bölümde geliştirdiğimiz algoritma(lar) anlatılmıştır. Dördüncü bölümde değerlendirme sonuçlarına yer verilirken, son bölümde ise sonuç ve öneriler verilmiştir.

2. Literatür Araştırması

Şu ana kadar kavram madenciliği üzerine yapılan çalışmaların çoğunda yapay zeka algoritmaları ve WordNet kullanılmaktadır. Türkçe üzerine yapılan çalışmalar çok kısıtlı olmakla beraber yapay zeka algoritmaları kullanılmış; fakat Türkçe'de kavram madenciliği üzerine WordNet kullanımına pek başvurulmamıştır.

Bir çalışmada web sitelerinden kavram çıkartılmaya çalışılmıştır. Önce stop-word olan sözcükler elenmiş, ardından frekansı belli bir eşik değeri aşan kelimeler kavram setine atanmıştır. <*html*>, <*body*> ve <*title*> gibi etiketlere (tag) de belli katsayılar atanmış ve bu etiketlerin temsil ettiği kelimelerin kavram olup olmayacağının belirlenmesinde bu katsayı skorları da etkili olmuştur. Örneğin <*b*> ve <*title*> etiketlerine daha yüksek değerli katsayılar atanmıştır; çünkü bunlar daha yüksek öneme sahiptir. Bu çalışma *bag-of-words* özelliğine sahiptir. [4]

Diğer bir çalışmada WordNet içinde bulunan *synset* ilişkileri kullanılarak kümeleme (clustering) algoritması izlenmiştir. Burada bütün *synset* ilişkilerinin hesaba katılmasının kümeleme üzerinde başarısız sonuçlara neden olduğu gözlenmiştir. [5]

Bir çalışmada ise kavramlar dokümandaki frekans, çap gibi özelliklerine bakılarak çıkarılmış, ardından bunun üzerine metin sınıflandırılması yapılmaya çalışılmıştır. Ancak bu kavram çıkarma işleminde kavramlar dokümanlardaki kelimelerden birisi olabilmekte, dokümanda geçmeyen bir kelime kavram olarak belirlenememektedir, bu da pek başarılı sonuçlar vermemektedir. [6]

Diğer bir çalışmada bir yapay zeka uygulaması olan *Latent Dirichlet Allocation* kullanılmıştır. Bu algoritma her ne kadar dokümanlardan konu (topic) çıkarmak için geliştirilmiş bir metot olsa da, bu çalışmada kavram çıkarmada elde edilen başarı yüksektir. [7]

Türkçe üzerine yürütülen bir çalışmada ise kümeleme (clustering) algoritması izlenmiştir. [8] Kümeleme algoritması izlenirken, ilk önce manüel olarak kümelere kelimeler atanmış, sonra dokümanlara da bu kümeler atanarak kavram çıkarılmaya çalışılmıştır. Geliştirilen bu algoritmanın başarı oranı %51'dir. Bizim geliştirdiğimiz algoritmaya göre ise ne yapay zeka algoritmaları, ne de WordNet kullanılmış, onun yerine sözlük kullanılarak istatistiksel bir metot izlenmiştir.

3. Sözlük Kullanımı

Şu ana kadar kavram madenciliği üzerine geliştirilmiş istatistiksel metotlar arasında en çok başvurulan yöntem WordNet kullanımıdır. Kavramlar bir kelimenin genel anlamını ifade ettiği ve WordNet'te bulunan hipernimi özelliği de bir kelimenin genel anlamını belirttiği için bu veritabanının kullanımı oldukça faydalı sonuçlar vermiştir. Fakat yalnızca hipernimi özelliğinin hesaba katılıp, diğer ilişkilerin gözardı edilmesi başarı oranlarını düşürebilmektedir. Örneğin bir dokümanda *talebe* kelimesi çok sık geçiyorsa, bu kelimenin yaygın kullanılan eşanlamlı kelimesi olan *öğrenci* sözcüğü bu dokümanın kavramı olarak belirlenebilmelidir. Türkçe sözlüğündeki anlam cümlelerinde bir sürü anlam ilişkisinin varlığını ve Türkçe WordNet veritabanının pek gelişmiş olmadığını hesaba kattığımızda, TDK sözlüğünü kullanmamız mantıklı gelmekte, sonuçlar da başarılı çıkmaktadır.

Bu çalışmada dokümanlardan anlamlı kavramlar çıkarmak için TDK sözlüğünün elektronik XML formatından yararlanılmıştır. Sözlükteki bir kelimenin XML formatındaki özellikleri aşağıdaki gibidir.

- <name>: Kelimenin ismini,
- <affix>: Kelimenin son eki olup olmadığını,
- <lex_class>: Kelimenin grubunu (isim, sıfat, vb.),
- <stress>: Kelimenin hangi hecesinin vurgulandığını,
- <pronunciation>: Kelimenin telaffuzunu,
- <origin>: Kelimenin geldiği dili,
- <meaning>: Kelimenin anlamını,
- <quotation>: Kelimenin kullanıldığı bir alıntı cümleyi,
- <atasozu_deyim_bilesik>: Kelimenin hangi atasözü, deyim veya bir bileşik isimde kullanıldığını belirtir.

Bu çalışmada, sözlükten faydalanılırken göz önünde bulundurulan özellikler <*name*>, <*lex_class*> ve <*meaning_text*> tag'leridir. Bu kelimeler arasından sadece isim olanlar hesaba katılmış diğerleri elenmiştir, çünkü kavramlar çoğunlukla isim olarak düşünülmektedir. Diğer etiketlerin (tag) pek bir önemi yoktur, örneğin bir kelimenin hangi dilden geldiğinin (<origin> etiketi) bu kelimenin kavramıyla hiçbir ilişkisi yoktur. Anlam cümlelerindeki kelimeler de (çekim) eklerinden ayrılıp işlenmelidir. Dikkat edilmesi gereken bir nokta da bir kelimenin (*dernek* kelimesi gibi) birden çok anlama sahip olabilmesidir, bunun için bağlamsal analiz gerçekleştirilerek, dokümandaki kelimenin hangi anlamının kullanıldığı tespit edilmektedir. Dokümandaki sözcüğün ve bu sözcüğün sözlükteki anlam cümlesinde geçen kelimeler arasında birçok anlamsal ilişki vardır, bunlardan bazıları aşağıda açıklanmıştır.

Kavram çıkarmada en çok kullanılan anlamsal ilişki daha önce bahsedildiği gibi hipernimi özelliğidir. Şu ana kadar özellikle İngilizce'de kavram madenciliği alanında yürütülen çalışmalarda hipernimi dışında bir anlamsal ilişkiden pek faydalanılmamıştır, bunun nedeni ise kavramın bir sözcüğün genelde soyut anlamını ifade etmesi ve hipernimi özelliğinin

bu anlamı içermesidir. WordNet'in hipernimi anlamsal ilişkisi (*synset*) bir genel anlam sözcüğü döndürebilirken, sözlük tanımlarında birden fazla hipernim kelimesi döndürülebilmektedir. Örneğin *aslan* kelimesinin sözlük tanımında *kedigiller* ve *hayvan* gibi iki hipernim kelime olabilir, bu da WordNet'teki *aslan* kelimesinin tek hipernim kelimesine sahip olmasına kıyasla daha başarılı sonuçlar verebilmektedir. Hipernimi özelliği yanında sinonimi (eş anlamlılık) özelliği de kavram belirlemede bir rol oynayabilir, örneğin bir dokümanda *hekim* kelimesi çok geçiyorsa bu dokümanın *doktor*lar ile ilgili bir konu işlediği kanısına varabiliriz, dolayısıyla bu dokümanın kavramı *doktor* diyebiliriz. Kavram çıkarmada kullanılabilecek olan başlıca iki anlamsal ilişki hipernimi ve sinonimi olsa da diğer anlamsal ilişkilerden de (zıt anlam hariç) bu süreçte faydalanılabilir. Diğer anlam ilişkilerinden bazıları meronimi (bileşen anlam ilişkisi), hiponimi (daha dar kavram anlam ilişkisi) ve zıt anlamdır (antonimi), bunlardan meronimi özelliği kavram çıkarmada kullanılabilir (bir dokümanda *parmak* kelimesi sık geçiyorsa, parmak kelimesinin bileşeni olduğu el sözcüğü kavram olarak belirlenebilir). Ayrıyeten sözlük anlam cümlelerinde kullanabilirlik (sabun-yıkama), yer (mutfak-ev), etki (kaza yapmak-yaralanmak), altolay (uyumak-horlamak), önkoşul (işe gitmek-uyanmak) gibi WordNet'te *synset* olarak bulunmayan anlamsal ilişkilerin bulunması, sözlük kullanımı ile daha başarılı sonuçlara ulaşılabilmesini sağlar.

Bu anlam benzerlikleri (zıt anlam hariç) kelimeleri birbiriyle ilişkilendirerek, birçok alanda kullanılabilir. Kümeleme yöntemi uygulanacak olursa benzer anlamlara sahip kelimeler aynı kümeye atanabilir. (Örneğin karanfil kelimesi ile gül kelimesinin TDK anlam cümlelerinde ortak kelimeler olduğu için –bitki gibi- bunlar aynı kümeye atanabilir.) Bu çalışmanın ana algoritmasından ayrı olarak bu yöntem uygulanmış; fakat çok başarılı sonuçlar elde edilmemiştir (kümeleme yöntemi ile kesinlik başarı oranı %40.17 olarak tespit edilmiştir), bunun nedeni ise bütün derlem doküman kelimeleri ile oluşturulan büyük veri seti matrislerinin (satırların doküman kelimelerini, sütunların ise doküman kelimelerinin sözlük tanım cümlelerindeki sözcükleri temsil ettiği) oldukça fazla sayıda 0 değeri içermesidir. PCA (Principal Component Analysis) uygulanarak boyut azaltılmaya çalışılmış ve 3 ile 4 seviyeli hiyerarşik metotlarla matrisler oluşturulmuştur. Fakat daha sonra kümeleme yerine daha basit bir istatistiksel metot izlenmiş, sonuçların daha başarılı olduğu gözlenmiştir.

3.1. Kelime Anlamının Tespiti

ConceptNet [9], WordNet [10] gibi sözlükler yapısal (structured) bir özelliğe sahip olması nedeniyle bir ön-işleme sürecine tabîi tutulmak zorunda değildir; fakat TDK Sözlüğü için durum farklıdır. Anlam cümlelerindeki kelimeler genelde çekim ekleriyle birlikte bulunmaktadır ve kavramlar bu çalışmada isim olarak düşünüldüğü için Boğaziçi Üniversitesi'nde geliştirilen BoMorP ve BoDis araçları kullanılmıştır. [11] [12] Bu araçlarla kelimeler çekim eklerinden ayrılarak kelimelerin kökleri elde edilmektedir ve kelime grupları (isim, sıfat vb.) belirlenebilmektedir.

Bir dokümanda geçen kelimenin TDK sözlüğünde birden çok anlamı bulunabilmektedir, bu durumda hangi anlamının kullanıldığı sözlüğe bakılarak belirlenebilir. Bunun için birden çok sözcük anlamı bulunan kelimelerin 30 kelimelik bağlamlarına bakılmıştır. Buna göre kelimenin dokümanda sağında geçen 15 ve solunda geçen 15 kelimeye bakılmıştır. Sözlük anlam cümlelerinden hangisinde bağlamlarda geçen ortak kelime frekansı normalize edilmiş (sözcük anlam cümlesi uzunluğuyla bölünerek) haliyle en fazlaysa, o anlam cümlesi kullanılmakta olan anlam olarak belirlenmektedir. Bu muğlaklık gidermenin formülü aşağıda verilmiştir. Bu formülde m sözlük anlam cümlesini, c_w ise w kelimesinin derlemdeki bağlamını (context) belirtmektedir.

$$\text{argmax}_m \, Benzerlik(m, c_w) = \frac{OrtakKelimeSayisi(m,c_w)}{Uzunluk(m)} \quad (1)$$

3.2. Kavramların Çıkarılması

Bu çalışmada iki farklı algoritma geliştirilmiş, ikinci algoritmanın daha başarılı olduğu gözlenmiştir. Geliştirdiğimiz algoritmalara göre, bir dokümanın olası kavramları çıkarılırken, o dokümanda en sık geçen kelimelerin daha fazla bir ağırlığı olmalıdır. Örneğin bir dokümanda çok sayıda *voleybol* kelimesi geçiyorsa, o dokümanın olası kavramlarından birisini *spor* olarak atayabiliriz. Bizim geliştirdiğimiz algoritmada aşağıdaki formül kavram atamada kullanılmıştır:

$$\text{Kavram(k)} = \text{Sıklık(k)} \times \text{Konum(k)} \times \text{Kapsam(k)} \quad (2)$$

Yukarıdaki formülde *Sıklık* bir kelimenin frekansını, *Konum* o kelimenin dokümanda bulunduğu ilk konumu (dokümanda baştan kaçıncı kelime olduğunu), *Kapsam* ise kelimenin bir dokümanda ilk geçtiği yerle son geçtiği yer arasının kapsadığı kelime sayısının toplam doküman kelime sayısına bölünmesiyle elde edilen değeri ifade eder. Örneğin *hekim* kelimesi dokümanın ilk kelimesi ve son kelimesiyse, bu dokümandaki kapsamı en fazla olan kelime budur. Eğer bir kelime dokümanın başlarında geçiyorsa, bu kelimenin öneminin daha fazla olduğu anlaşılabilir, örneğin başlık kelimeleri dokümanın genel kavramını ifade edebileceği için bunların ağırlığı sonlara doğru geçen kelimelere oranla daha fazla olmalıdır. Aynı zamanda bir kelimenin ilk ve son geçtiği yerler dokümanın oransal olarak çoğunu kapsıyorsa, bu kelime dokümanın olası kavramlardan biri olabilmektedir, o yüzden kapsamı geniş olan kelimelerin de ağırlığı daha fazla olmalıdır. Yukarıdaki formüle göre ham frekanslar hesaba katılırken, diğer faktörlerin logaritmik değerleri kullanılmıştır; çünkü bir kelimenin, o kelimenin geçtiği dokümandaki frekansı diğer faktörlere göre daha önemlidir, dolayısıyla katsayısı da daha fazla olmalıdır.

Bu formülün ürettiği değerler göz önünde bulundurularak bir matris oluşturulur ve matris hücreleri bu değerlerle doldurulur. Bu matriste satırı temsil eden kelimeler, o dokümanda bulunan kelimeler, sütunları temsil eden kelimeler ise doküman kelimelerinin sözlük anlam cümlesi kelimeleridir. Her satır ve sütun kelimesi matriste en fazla bir defa bulunmaktadır.

Örneğin dokümanda geçen kelimelerden birisi *voleybol* ise matrisin bir satırı *voleybol* kelimesini etmekte, bu kelimenin sözlükte geçen anlam cümlesi kelimelerinden *oyun*, *spor* ve *takım* kelimeleri ise üç sütunu temsil etmektedir. Dokümandaki diğer kelimeler de hesaba katılıp matris değerleri formül 1'e göre oluşturulan değerlerle doldurulur ve sütun değerleri toplanır. Hangi sütun değeri en fazlaysa o sütunu temsil eden kelime dokümanın kavramı olarak belirlenebilir. Matristeki bazı hücrelerin değerinin 0 olması, o satırı temsil eden kelimenin sözlük anlam cümlesinde o sütunu temsil eden kelimenin bulunmadığı anlamına gelmektedir. Tablo 1'e göre bir dokümanda iki tane isim olduğu (*kaplan* ve *maymun*) farzedilirse ve bu kelimelerin frekansları sırasıyla 2 ve 3 ise, bu dokümanın kavramının *hayvan* olduğu görülebilmektedir. (*Hayvan* kelimesi her iki doküman sözcüğünün de sözlük anlam cümlesinde geçmektedir ve bu kelimeye denk düşen sütunun değeri diğer sütun toplamlarından daha fazladır.)

	Kedi-giller	Post	Hayvan	Kuyruk	Kap-lan	May-mun
Kaplan	2	2	2	0	2	0
Maymun	0	0	3	3	0	3
Toplam	2	2	**5**	3	2	3

Tablo 1. Genel sözlük algoritmasına göre kelimeleri *kaplan* ve *maymun* olan dokümanın kavramı *hayvan* olarak belirlenmektedir

Yukarıda anlatılan algoritmaya göre iki seviyeli bir metot izlenmiştir. İki seviyeli algoritmaya göre dokümanlardaki kelimeler birinci seviyede, o kelimenin anlam cümlesindeki kelimeler ise ikinci seviyede yer almaktadır. Ayrıyeten üç seviyeli yapılar da geliştirilerek kavram atama işlemi yapılmıştır, buna göre birinci seviyede dokümandaki kelime, ikinci seviyede o kelimenin anlam cümlesindeki kelimeler, üçüncü seviyede anlam cümlesindeki kelimelerin anlam cümlesindeki kelimeleri geçmektedir. Bu hiyerarşik yapıya bir örnek Şekil 1'de verilmiştir.

- ***Jaguar [1]***
  - ***Hayvan [0.44]***
    - ***Organizma [0.26]***
    - ***Yaratık [0.26]***
    - ***Duygu [0.26]***
    - ***İçgüdü [0.26]***
  - ***Kedigiller [0.44]***
    - ***Kedi [0.26]***
    - ***Soy [0.26]***
    - ***Sınıf [0.26]***
    - ***Hayvan [0.26]***
  - ***Post [0.44]***
    - ***Tüy [0.26]***
    - ***Hayvan (Elenir)***
    - ***Deri [0.26]***

Şekil 1. *Jaguar* kelimesinin 3-seviyeli hiyerarşik yapısı

Burada kelimelerinin yanındaki sayısal değerler, matriste bu kelimelere denk düşen hücrelerin çarpılacak katsayılarıdır. Örneğin en üst seviyedeki kelimenin katsayısı 1 iken, ikinci seviyenin katsayısı 0.44, üçüncü seviyenin katsayısı ise 0.26 olabilmektedir. Bu katsayı atama mantığının nedeni hiyerarşik yapıda üstten aşağıdaki seviyelere inildikçe anlamsal ilişkinin zayıflamasıdır, katsayılar ise bir seviye aşağı inince kelimelerin genelde geometrik olarak artışından dolayı geometrik olarak azalır. Eğer bir kelime hiyerarşik yapıda birden fazla defa görülüyorsa en üst seviyedeki korunur, diğerleri elenir. (Şekil 1'de *hayvan* kelimesi iki defa göründüğü için alt seviyedeki elenir.) En başarılı sonuçların iki seviye için çıktığı, üç seviyeli yapının yalnızca bir derlemde daha başarılı olduğu, dört seviyeli yapının başarısız sonuçlar verdiği gözlenmiştir. Sözlükte en çok geçen %1'lik kelime *stop-words* olarak belirlenmiş, bunlar elenmiştir. Bu algoritmaya göre bir dokümanın kavramı belirlenirken, o kavram kelimesi dokümanda geçmek zorunda değildir. Bir dokümandan çıkarılan kavramın o dokümanda bulunmak zorunda olmayışı doğru ve faydalı bir yaklaşımdır; çünkü sadece dokümanda bulunan önemli kelimeleri çıkarma işlemi olan anahtar kelime çıkarımı ile kavram çıkarımı arasındaki en büyük fark budur. Bu algoritmanın psödokodu Şekil 2'de verilmiştir.

Yukarıda anlatılan genel sözlük algoritmasının (DictAlg) büyük derlemlerin ve uzun dokümanların kavramlarını çıkarmada başarılı olduğu gözlense de kısa doküman veya metinlerden kavram çıkarmada başarısız olduğu gözlenmiştir. Bunun nedeni ise bir kelimenin kavramının onun sözlük tanımındaki bütün kelimelerle alakalı olmamasıdır.

Girdi

F1: Derlem dokümanları

Output

F2: Doküman kavramları

Başla

1: *L* <- *F1*'i *liste*ye ata
2: her *L* dokümanı **için**
3: *Matris* <- Ø
4: doküman *i*'deki her *j* kelimesi **için**
5: Dokümanda *j* kelimesinin kullanıldığı bağlamdaki anlam cümlesi kelimelerini *Anlam*'a ekle
6: *j* kelimesini *Anlam*'a ekle
7: *Anlam*'daki her *k* kelimesi **için**
8: *Matris*(*j, k*) = *Sıklık*(*j*) * *Konum*(*j*) * *Kapsam*(*j*)
9: **döngü bitimi**
10: **döngü bitimi**
11: *Matris*in boş hücrelerini sıfır değeriyle doldur
12: *Matris* <- Aynı olan satır ve sütunları sil
13: *Liste* <- topla(*Matris* sütunları)
14: *Liste* <- sırala(*Liste*)
15: En yüksek değere sahip Liste elemanlarını F2'ye ekle
16: **döngü bitimi**

Bitim

Şekil 2. Genel sözlük algoritması

Geliştirilen ikinci algoritmaya göre dokümandaki kelimelerin bütün derlemde bulunan 30 kelime boyutlu bağlamlarına (kayan pencere), yani 15 sağındaki, 15

solundaki kelimeye ve kelimenin kendisine bakılmıştır. Hangi kelimeler hem bu bağlamlarda, hem de sözcüğün sözlük anlam cümlelerinde geçiyorsa onlar hesaba katılmış, diğerleri gözardı edilmiştir. Bu derlem-bazlı yaklaşım mantıklıdır; çünkü büyük derlemlerde kelimeler, bağlamlarında genelde bu sözcüklerin genel anlamını ifade eden, yani kavramlarıyla birlikte geçer. Bu algoritma için de iki alt-algoritma geliştirilmiştir, onlar da aşağıdaki gibidir:

- Yoğun kayan pencere algoritması: Buna göre 30-kelimelik bağlamlarda (kayan pencere) ve sözlük anlam cümlelerinde geçen ortak kelimelerin hepsi hesaba katılıp, matris değerleri ona göre doldurulmaktadır.
- Seyrek kayan pencere algoritması: Bu algoritmaya göre 30-kelimelik bağlamlarla sözlük tanım cümlelerinde geçen ortak kelimelerden hangisinin bütün derlem bağlamlarında frekansı en yüksekse sadece o hesaba katılmıştır, yani önceki alt-algoritmaya göre kısıtlı olarak bir kelimenin bağlamlarından en fazla bir kelime (sözlük anlam cümlesinde de geçiyorsa) göz önünde bulundurulmaktadır. Bu algoritmaya göre dokümandaki her kelime için iki kelime hesaba katılmıştır: Dokümandaki kelimenin kendisi ve bu kelimenin bağlamlarında, sözlük anlam cümlesinde de geçme şartıyla, en sık geçen kelime. Matris değerleri de buna göre doldurulmuştur.

Yukarıda anlatılan iki alt-bağlam algoritmasını genel sözlük algoritmasına göre daha başarılı sonuçlar verdiği gözlenmiştir; çünkü bir kelimenin sözlük tanımındaki her sözcük o kelimenin kavramıyla, yani genel anlamıyla ilişkilendirilemez.

4. Deneyler ve Sonuçlar

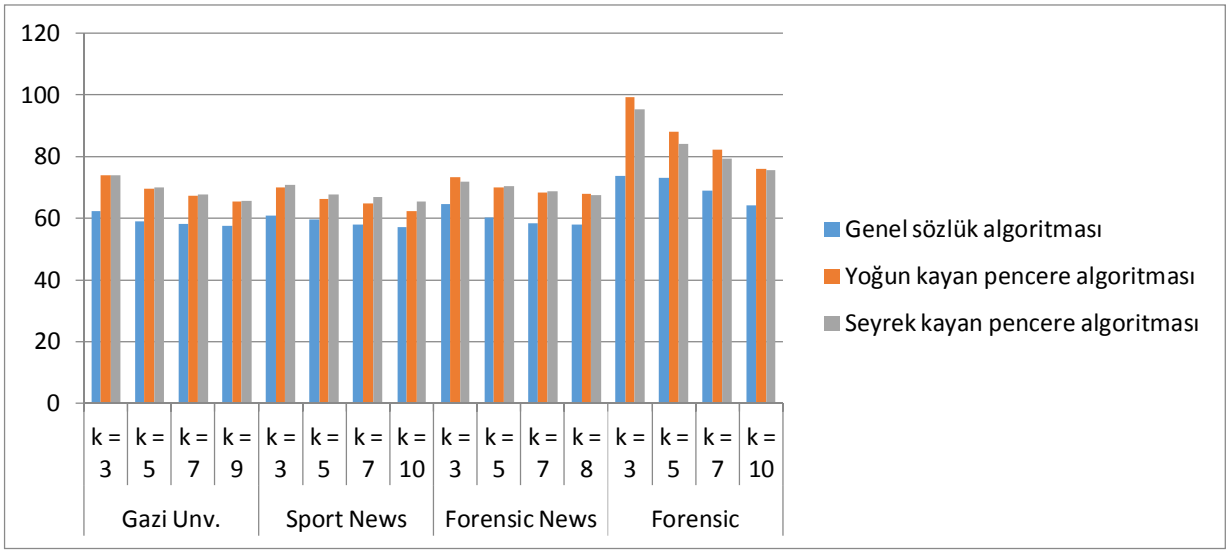
Bu çalışma için dört derlemdeki toplam 368 dokümandan geliştirilen üç algoritmaya göre kavramlar çıkarılmıştır. Bu derlemler *Gazi Üniversitesi derlemi (Gazi Unv..), spor haberleri derlemi (Sport News), yargıtay kararları derlemi (Forensic)* ve *yargıtay haberleri derlemi*dir *(Forensic News).*

Başarıyı ölçmek için dokümanlardan manüel olarak kavramlar çıkarılmış, bunlar algoritmanın çıkardığı kavramlarla kıyaslanmıştır. Kavramlar hem algoritmik, hem de manüel olarak belirlenirken önem sırasına göre çıkarılmıştır, ilk 3, 5, 7, 8, 9, 10 ve 15 kavramlar kıyaslanmıştır. Örneğin Tablo 2'ye göre birinci doküman için manüel olarak çıkarılan (önem sırasına göre) ilk üç kavramdan ikisi (*spor* ve *karşılaşma*), algoritmik olarak çıkarılan ilk üç kavramda yer aldığı için başarı oranı 2 / 3, yani 0.66'dır. Doküman 2'de ise baş 3 kavram arasında sadece bir ortak kelime olduğu için başarı oranı 0.33 olarak bulunabilmektedir.

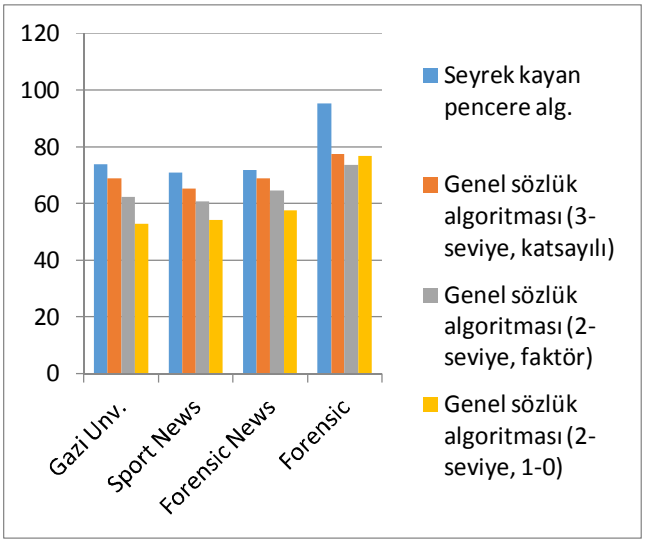
Dokümanlar	Algoritma	Manüel
Doküman 1	Spor, Oyun, Karşılaşma	Spor, Karşılaşma, Politika
Doküman 2	Mahkeme, Avukat, Hakim	Avukat, Sanık, Karşılaşma

Tablo 2. İki dokümanın baş üç kavramı

İlk kavramları kıyaslama yanında izlenen diğer bir başarı ölçme metriği ise algoritmik olarak çıkarılan ilk kavramları, o dokümandan manüel olarak çıkarılan bütün kavramlarla kıyaslama yoludur. Bu yöntem çok daha yüksek başarı oranları vermektedir.



Tablo 3. Dört derlem için elde edilen başarı yüzdeleri



Tablo 4. Birinci algoritma alt-metotları ile ikinci alt-algoritma (seyrek kayan pencere) sonuçlarının kıyaslanması

Başarı sonuçları Tablo 3'te verilmektedir. Bu sonuçlara göre genel sözlük algoritmasında (birinci algoritma) en başarılı sonuçlara 2-seviyeli hiyerarşik yapıda, frekans ve diğer faktörler göz önünde bulundurularak doldurulan matris ile ulaşılmaktadır. 3-seviyeli hiyerarşik yapılı ve sadece 0-1 değerleriyle (frekans ve diğer faktörler gözardı edilerek) doldurulan matris ile kavram çıkarma işlemi göreceli daha başarısız sonuçlar vermektedir. En başarılı sonuçlara ikinci algoritma ile (bağlam tabanlı) ulaşılmıştır. Bazı derlemlerde sonuçların daha başarısız çıkmasının sebebi bu derlemlerin bir sürü farklı konuda dokümanları içermesidir, dolayısıyla bir kelimenin bağlamlarından onun genel anlamını ifade eden kavramını çıkarmada zorluk çıkmaktadır.

Örnek amacıyla Tablo 4'te birinci algoritmanın bütün alt-metotlarıyla ikinci algoritmanın bütün derlemler için k değerinin üç olarak hesaba katılmasıyla elde edilen başarı sonuçlarının kıyaslaması verilmiştir. Genel sözlük algoritmasına (İki-seviye, 1-0) göre frekans ve diğer faktörler gözardı edilerek matris hücreleri sadece kolon hücreleri satır hücrelerinin sözlük tanım cümlelerinde geçip geçmemesine göre 0 ve 1 değerleriyle doldurulmuştur. Genel sözlük algoritmasına göre (İki-seviye, faktör) ise öncekinden farklı olarak frekans, ilk konum ve kapsam faktörleri, genel sözlük algoritmasına (üç-seviye, katsayı) göre ise üç-seviyeli hiyerarşik yapı hesaba katılmıştır. İkinci algoritma ise seyrek kayan pencere algoritmasıdır.

5. Sonuç ve Öneriler

Bu çalışmada daha önce Türkçe'de kavram madenciliği üzerine denenmemiş bir yol olarak sözlük tabanlı bir algoritma izlenmiş ve başarılı sonuçlar elde edilmiştir. WordNet'in sadece bir özelliği (hipernimi) hesaba katılarak kavram çıkarma yöntemi yetersiz kalabilmektedir. Bu sözlük tabanlı kavram çıkarma algoritması ile dokümanlarda geçmeyen kelimeler de kavram olarak belirlenebilmektedir. Geliştirilen alt-metotlardan en başarılısı ikinci algoritmadır; çünkü bir kelimenin kavramı onun sözlük tanımı kelimelerinin hepsiyle alakalı değildir, dolayısıyla bazı sözlük anlam cümlesi kelimelerinin elenmesi mantıklı sonuçlar doğurmuştur.

Gelecek çalışmalarda, kavram madenciliği üzerine bu makalede anlatılan algoritmanın çıkardığı kavramlarla metin sınıflandırması (*text categorization*) üzerine çalışılacaktır. Ayrıca arama motoru sonuçlarında dokümanlardan çıkarılan kavramlara göre sayfa sıralamasının (*page ranking*) düzenlenmesinin başarı oranını artırıp artırmayacağı incelenecektir.

6. Teşekkür

Bu çalışma 5187 onay numarasıyla Boğaziçi Üniversitesi Araştırma Fonu ve 110E162 onay numarasıyla TÜBİTAK tarafından desteklenmiştir. Cem Rıfkı Aydın TÜBİTAK BİDEB 2210 bursuyla Yüksek Lisans öğrenimi süresince desteklenmiştir. Haşim Sak'a Biçimbirimsel Analiz ve Muğlaklık giderici araçlarını bize sağladığı için teşekkür ederiz.

7. Kaynaklar

[1] Faber, V., Hochberg, J.G., Kelly, P.M., Thomas, T.R. &White, J.M.,"Concept Extraction – a datamining technique", **Los Alamos Science**, (1994).

[2] Bennett, N.A., He, Q. Chang, C.T.K. & Schatz, B.R., "Concept Extraction in the Interspace Prototype", **Technical Report, Dept. of Computer Science**, University of Illinois at Urbana-Champaign, Champaign, IL, (1999).

[3] Moens, M. & Angheluta, R., "Concept Extraction from Legal Cases: The Use of a Statistic of Coincidence", **International Conference on Artificial Intelligence and Law, ICAIL, ACM**, (2003).

[4] Ramirez, P. M. & Mattmann, C. A., "ACE: Improving Search Engines via Automatic Concept Extraction", **Information Reuse and Integration**, (2004).

[5] Pennock, D., Dave, K. & Lawrence S., "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews", **Twelfth International World Wide Web Conference (WWW'2003), ACM**, (2003).

[6] Chengzhi, Z. ve Dan, W.,"Concept Extraction and Clustering for Topic Digital Library Construction", **International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM**, (2008).

[7] AlSumait, L., Barbar´a, D. ve Domeniconi, C., "OnLine LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking", **Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM**, (2008).

[8] Uzun, M., "Developing a concept extraction system for Turkish", **International Conference on Artificial Intelligence, ICAI**, (2011).

[9] ConceptNet, http://conceptnet5.media.mit.edu/, 8 Ocak 2014.

[10] WordNet, http://wordnet.princeton.edu/, 8 Ocak 2014.

[11] Sak, H., Güngör, T. ve Saraçlar, M, "Morphological disambiguation of Turkish text with perceptron algorithm", **CICLing 2007**, vol. LNCS 4394, pp. 107-118, (2007).

[12] Sak, H., Güngör, T. ve Saraçlar, M., "Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus", **GoTAL 2008**, vol. LNCS 5221, pp. 417-427, Springer, (2008).