Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd
https://doi.org/10.53070/bbd.990438

Journal of Computer Science
ISSN: 2548-1304
Volume: IDAP-2021, Issue: Special , pp: 175-183
@Anatolian Science

Konu benzerliğine dayalı makale tavsiye sistemi

Paper recommendation system based on topic similarity

Esra Gündoğan*[1], Mehmet Kaya[2]

[1,2]Bilgisayar Mühendisliği Bölümü, Fırat Üniversitesi, Elazığ, Türkiye

(egundogan@firat.edu.tr, kaya@firat.edu.tr)

Received: Sep.3, 2021 Accepted: Sep.26, 2021 Published: Oct.20, 2021

Özetçe— Akademik ilerleme ile beraber araştırmacılar tarafından yayınlanan makale sayısı her geçen gün artmaktadır. Yayın sayısındaki artış ilgilenilen konu ile ilgili çalışmalara ulaşmayı zorlaştırmaktadır. Tavsiye sistemleri bu noktada araştırmacılar için önemli bir araçtır. Kullanıcıların profiline ya da yayınların konu benzerliğine dayalı makale tavsiye sistemleri istenilen bilgiye ulaşmada kullanıcılara oldukça yardımcı olmaktadır. Bu çalışmada girilen makalenin konusuna benzer makaleleri tavsiye etmek için bir yaklaşım önerilmiştir. Oluşturulan sistem doküman benzerliği, kümeleme ve anahtar kelime çıkarımı konularının birleştirilmesiyle hem anahtar kelime hem de içerik benzerliklerini dikkate alarak makale tavsiye etmektedir. Derin öğrenme tabanlı yöntemlerin çalışmanın her adımında kullanılması tavsiye sisteminin performansını artırmıştır. Çalışma bilgisayar biliminde makine öğrenmesi, yapay zeka, insan bilgisayar etkileşimi gibi farklı kategorilerden makaleleri içeren bir veri seti üzerinde uygulanmıştır. Kullanıcılara sorguları ile yüksek benzerliğe sahip makaleler önerilmiştir. Böylece istenilen konuya yönelik çalışmalara erişim daha hızlı ve daha kolay bir hale getirilmiştir.

Anahtar Kelimeler : *Anahtar kelime çıkarımı, doküman benzerliği, kümeleme, makale tavsiye sistemi*

Abstract— Along with academic progress, the number of papers published by researchers is increasing day by day. The increase in the number of publications makes it difficult to reach studies on the subject of interest. Recommendation systems are an important tool for researchers at this point. Paper recommendation systems based on the profile of the users or the topic similarity of the publications are very helpful to the users in reaching the desired information. In this study, an approach is proposed to recommend papers similar to the subject of the paper searched. The created system recommends papers considering both keyword and content similarities by combining document similarity, clustering, and keyword extraction. The use of deep learning-based methods in every step of the study has increased the performance of the recommendation system. The study has been applied to a dataset containing papers from different categories such as machine learning in computer science, artificial intelligence, human-computer interaction. Users are offered papers with high similarity to their queries. Thus, access to studies on the subject of interest has been made faster and easier.

Keywords: *Keyword extraction, document similarity, clustering, paper recommendation system*

1. Giriş

Günümüzde, akademik ilerlemeye bağlı olarak, bilimin her alanından araştırmacılar tarafından birçok çalışma yürütülmektedir. Araştırmacılar bu çalışmalarını ve sonuçlarını diğer araştırmacılar ile paylaşarak akademik gelişime katkıda bulunmaktadırlar. Her yıl konferanslarda ve dergilerde binlerce

makale yayınlanmaktadır. Bunun sonucu olarak, bilimsel bilgi hızlı bir şekilde artmıştır [1]. Bu bilgi fazlalığı içerisinde ihtiyaç duyulan bilgiye ulaşmak da giderek zorlaşmıştır.

Akademisyenler veya araştırmacılar konularına yönelik önceden yapılmış çalışmalara ulaşmak, çalışılan konuda son durum hakkında bilgi sahibi olmak ister ve elde ettikleri sonuçlar doğrultusunda çalışmalarına yön verirler. Bu kadar fazla yayın içerisinde istenilen çalışmalara ulaşmak kolay değildir. Bu araştırma alanına yönelik arama işlemini kolaylaştırmak için tavsiye sistemleri geliştirilmiştir. Makale tavsiye sistemleri, en alakalı makaleleri otomatik olarak araştırmacılara sunmak ve araştırmacılara zaman kazandırmak için çok önemlidir.

Çoğu makale tavsiye sistemi kelimelerin bağlamını ve makaleler arasındaki anlamsal benzerliği dikkate almaz [2]. Ancak bu çalışmada derin öğrenme tabanlı yöntemler ile makaleler işlenerek kelimelerin ve dokümanların anlamsal benzerliğine dayalı bir tavsiye sistemi geliştirilmiştir. Bu çalışma temel olarak iki aşamadan oluşmaktadır. Birinci aşamada makalelerin konularına göre kümelenmesi, ikinci aşamada ise kullanıcı tarafından girilen makaleye benzer makalelerin ilgili kümede aranarak kullanıcıya önerilmesi gerçekleştirilmektedir. İlk olarak bilgisayar bilimleri alanında insan-bilgisayar etkileşimi, makine öğrenmesi, yapay zeka, bilgisayar görmesi gibi farklı kategorilerden makalelerin başlık ve özet bilgilerini içeren bir makale veri seti oluşturulmuştur. Bu veri seti hiyerarşik kümeleme ile kategorilerine göre kümelere ayrılmıştır. Bu kümelerin hangi kategorilere ait olduğu belirlemek için Word2Vec yöntemi ile kümedeki makalelerin anahtar kelimeleri çıkarılmış ve kümeyi temsil eden anahtar kelime vektörü bulunmuştur. Bu şekilde ortak kategorideki makaleler aynı kümelerde yer almış ve her küme için o kümede yer alan makalelerin konusuna yönelik anahtar kelimeler elde edilmiştir. İkinci aşamada ise, kullanıcı tarafından girilen bir makalenin ilk olarak başlık ve özetinden bir anahtar kelime vektörü oluşturulmuştur. Bu vektör küme vektörleri ile karşılaştırılarak hangi kümedeki makalelerin sorgudaki makaleye benzer olduğu belirlenmiştir. Küme belirlendikten sonra kümedeki tüm makalelerin ve kullanıcı tarafından girilen makalenin Doc2vec vektörleri bulunarak benzerlikleri hesaplanmıştır. Makaleler benzerlik skorlarına göre sıralanarak girilen makaleye en benzer ilk on makale kullanıcıya önerilmiştir. Bu şekilde, kümeleme, anahtar kelime çıkarımı ve doküman benzerliği konuları bir araya getirilerek bir tavsiye sistemi oluşturulmuş ve araştırmacılara makale arama sürecinde büyük kolaylık sağlanmıştır.

Makalenin geri kalan kısmı şu şekilde organize edilmiştir. Bölüm 2 makale tavsiye sistemleri konusunda daha önce yapılan çalışmaları sunmaktadır. Bölüm 3’ de oluşturulan tavsiye modelinin aşamaları ve kullanılan yöntemler detaylı bir şekilde açıklanmıştır. Bölüm 4 çalışmanın oluşturulan veri seti üzerinde uygulanması ve sonuçların tartışılmasını içermektedir. Bölüm 5’ te ise sonuçlar yer almaktadır.

2. İlgili Çalışmalar

Makale tavsiye sistemleri araştırma makalelerinin sayısındaki artış ile beraber makalelere daha hızlı erişimi sağlamak için önemli bir araştırma konusu haline gelmiştir. Son zamanlarda bu konuda birçok farklı yaklaşım önerilmiştir. Makale tavsiyesi ya kullanıcı profiline ya da araştırma alanına göre yapılmaktadır. Kullanıcı profilini dikkate alan çalışmalarda tavsiye kişilerin önceki tercihleri ya da okudukları makalelerden ilgi alanlarının tespitine dayalı olarak yapılmaktadır. Araştırma konusuna yönelik yapılan tavsiye de ise girilen makale ismi ya da anahtar kelimeler doğrultusunda kullanıcıya benzer konudaki makalelerin sunulması amaçlanmaktadır.

Lee ve ark. [3] sadece içerik tabanlı ya da graf tabanlı yaklaşım kullanıldığında karşılaşılabilecek sorunları ortadan kaldırmak için, iki yaklaşımı birleştiren hibrit bir yaklaşım önermiştir. Bu yaklaşımda, birbirinden ayrı olarak içerik tabanlı ve graf tabanlı tavsiye gerçekleştirilmiş ve daha sonra iki yaklaşımdan elde edilen sonuçlar birleştirilmiştir. Yapılan testler sonucunda bu yaklaşımın içerik ve graf tabanlı yaklaşımların sınırlamalarını ortadan kaldırabileceği görülmüştür. Pan ve ark. [4] hem atıf bilgisi hem de makalelerin içerik bilgisinden elde edilen özellikleri kullanarak bir heterojen graf modellemişlerdir. Heterojen grafa dayalı bir benzerlik öğrenme algoritması ile makale tavsiye sistemi oluşturmuşlardır. Bu çalışmada makaleler arasındaki benzerliklerin dikkate alınması tavsiye sisteminin performansını artırmıştır.

West ve ark. [5] makale tavsiyesi için atıf tabanlı bir yöntem önermişlerdir. Düğümlerin atıfları temsil ettiği bir graf oluşturulmuş ve PageRank algoritması ile sıralanmıştır. Ağ hiyerarşik olarak kümelenmiş ve bu hiyerarşik ağaca göre makale tavsiyesi yapılmıştır. Bu çalışma ile diğer çalışmalara göre daha kapsamlı tavsiyeler sunulduğu görülmüştür. Son ve ark. [6] atıf analizi ile ağ analizini birleştiren bir tavsiye sistemi önermişlerdir. Sadece makaleler arasındaki doğrudan bağlantıları kullanmak yerine ilgili makale ile dolaylı yoldan bağlantılı tüm makaleleri karşılaştıran bir atıf ağı oluşturulmuştur. Bu ağ üzerinden makalelerin ilişkileri incelenmiş ve ilgili makaleye benzer makaleler tavsiye edilmiştir.

Xia ve ark. [7] tarafından önerilen yöntemde tavsiye için ortak yazar ilişkileri ve kullanıcıların önceki tercihleri kullanılmıştır. Makaleler arasındaki ortak yazar ilişkileri ile ilgili iki özellik belirlenmiştir. Bir tavsiye listesi oluşturmak için, bu iki özellik ortak yazar ilişkileri ile birleştirilmiştir. Bulut ve ark. [8] kullanıcının aramasına uygun makale tavsiye etmek için araştırmacının önceki makalelerini ve araştırma alanını dikkate alarak kullanıcı profiline özgü bir tavsiye sistemi oluşturmuşlardır.

Kullanıcı profiline özgü tavsiye sistemler kullanıcının önceki çalışmaları, atıf ilişkileri, daha önceden okuduğu makaleler gibi ölçütler göz önüne alınarak gerçekleştirilir. Ancak yeni kullanıcılar ve yeni makaleler için bu sistemleri kullanmak verimli değildir. O yüzden bu çalışmada kullanıcı profili hakkında bilgi sahibi olmadan en verimli şekilde makale tavsiyesine odaklanılmıştır. Ayrıca araştırma konusuna yönelik tavsiye sistemlerinde sorgular veya girilen anahtar kelimeler doğrultusunda ilgili olabilecek makaleler bir liste halinde sunulmaktadır. Ancak yapılan bu listelemede genellikle anahtar kelimelerin benzerliği dikkate alınmakta makalenin içeriği göz ardı edilmektedir. Oysa makale içeriği çalışmaların benzerliği konusunda büyük önem taşımaktadır. Sadece anahtar kelime benzerliğine dayalı bir tavsiye kullanıcı için verimli olmayacaktır. Bu çalışmada, kişinin arattığı makaleye en benzer makalelerin tavsiyesi için hem anahtar kelimeler hem de girilen makalenin içeriği dikkate alınmıştır. Bu şekilde kullanıcının özellikleri bilinmese bile arattığı makalenin içeriğinden kişiye en uygun makalenin önerilmesi sağlanmıştır. Hedef makale ile aday makalelerin anlamsal benzerliklerinin dikkate alınması tavsiye sisteminin performansını artırmıştır.

3. Makale Tavsiye Sistemi

Önerilen makale tavsiye sistemi 4 adımda gerçekleştirilmiştir.

3.1. Doküman Vektörlerinin Oluşturulması

Benzer dokümanları kümelemek için ilk olarak metin dokümanların sayısal forma dönüştürülmesi gerekir. Bu şekilde dokümanlar vektörler ile ifade edilir. Doküman vektörlerini oluşturmak için farklı yöntemler vardır. En bilineni bag-of-words yöntemidir [9]. Ancak bu yöntemde kelimelerin sırası dikkate alınmaz ve kelimelerin anlamına bakılmaz. Bu dezavantajı ortadan kaldırmak için hem kelime sırasını hem de kelimelerin anlamını dikkate alan Doc2vec yöntemi önerilmiştir [10]. Doc2vec cümle, paragraf veya dokümanların vektör temsilini oluşturan derin öğrenme tabanlı bir yöntemdir ve metin işlemede son zamanlarda sıklıkla kullanılmaktadır [11].

Dokümanları vektörlere dönüştürme işleminde her bir makalenin başlık ve özet kısımları çıkarılarak bir doküman oluşturulmuştur. Vektörler oluşturulmadan önce her dokümana ön işleme uygulanmıştır. Ön işleme gürültülü veriyi azaltmak ve performansı artırmak açısından önemlidir. Ön işleme adımında noktalama işaretleri kaldırılmış, küçük büyük harf dönüşümü yapılmış ve durak kelimeleri kaldırılmıştır. Durak kelimeleri metinde sık geçen ve anlamsal olarak önemi olmayan kelimelerdir. Bu kelimeler kaldırılarak eğitim süresi kısaltılmıştır. Önişleme adımından sonra her dokümanda yer alan kelimeler belirlenmiş ve Doc2Vec modeli ile eğitilerek doküman vektörleri oluşturulmuştur. Oluşturulan doküman vektörleri her makaleyi anlamsal olarak temsil eden vektörlerdir. Vektörlerin vektör uzayında yakın konumlanması anlamsal olarak birbirine yakın olduklarını gösterir. Dolayısıyla vektör uzayında birbirine yakın olan vektörlerin ait oldukları dokümanlar birbirine anlamsal olarak benzemektedir. Oluşturulan doküman vektörlerinin birbirlerine olan uzaklıklarına göre aynı konudaki makaleleri kümeleme gerçekleştirilir.

3.2. Doküman Kümeleme

Kümeleme verileri anlamlı gruplara ayıran bir denetimsiz öğrenme yöntemidir. Giriş verisi yorumlanarak kümeler oluşturulur. Doküman kümeleme ise dokümanların benzerliklerine göre

gruplandırılmasıdır. Doküman kümeleme sayesinde çok fazla doküman içerisinde arama işlemi kolaylaştırılır. Veri seti arttıkça çalışma zamanı ve kümeleme kalitesi açısından daha iyi performans gösteren algoritmalara ihtiyaç duyulur [12]. Birch algoritması büyük veri setlerinde oldukça verimli çalışan bir birleştirici hiyerarşik kümeleme algoritmasıdır. Veriyi tek bir tarama ile oldukça iyi kümeleyebilir. Bu çalışmada dokümanları kümelemek için Birch algoritması kullanılmıştır.

Birch algoritması kümeleme için iki terim kullanır [13,14]. Kümeleme özniteliği ve kümeleme öznitelik ağacı. Kümeleme özniteliği, kümeler hakkında bilgiyi özetleyen üç boyutlu bir vektördür. Kümeler için gerekli bilgiyi elde etmek için bir ağaç oluşturulur. Birch algoritması 4 aşamada gerçekleştirilir. İlk aşamada, veri seti taranır ve veri bir ağaç oluşturmak için hafızaya alınır. Hafıza yeterli değilse yaprak düğümden ağaç tekrar oluşturulur. İkinci aşamada, aykırı veriler kaldırılır. Daha küçük bir ağaç oluşturmak için veri seti tekrar boyutlandırılır. Üçüncü aşamada, var olan bir kümeleme algoritması ağacın yapraklarına uygulanır. Son aşamada, isteğe bağlı olarak kümeler iyileştirilir.

3.3. Tavsiye Kümesinin Seçimi

Kullanıcı sorgusunda yer alan makaleye benzer makaleleri tavsiye etmek için ilk olarak hangi kümedeki makalelerin kullanılacağının belirlenmesi gerekir. Bu şekilde veri seti içindeki tüm makaleleri sorgu ile karşılaştırmak yerine sadece belirli bir küme içerisindeki makalelere bakılacaktır. Tavsiye kümesini belirlemek için hangi kümedeki makalelerin hangi konuya ait olduğunu bilmek gerekir. Bu yüzden her küme için o kümede yer alan makalelerin konusunu yansıtan anahtar kelimeler belirlenmiştir.

Binlerce doküman içerisinde ihtiyaç duyulan bilgiyi içeren dokümanlara ulaşmakta anahtar kelime çıkarımı kolaylaştırıcı olmaktadır. Anahtar kelime çıkarımı bir dokümanın konusunu en iyi tanımlayan terimlerin otomatik olarak belirlenmesidir. Birkaç anahtar kelime seçilerek kullanıcının makalenin tamamını okumasına gerek kalmadan konusunu anlaması ve istediği bir makale olup olmadığını belirlemesi sağlanabilir. Bu yüzden anahtar kelime çıkarımı bilgi çıkarımında önemli bir rol oynar [15-17]. Bu çalışmada kelime graf modeline dayalı TextRank algoritması [18,19] kullanılmıştır. TextRank algoritması PageRank algoritmasından türetilmiştir. Pagerank algoritması web sayfalarının önemini ölçmek için kullanılan bir algoritmadır. TextRank algoritması bu fikirden esinlenilerek metindeki önemli kelimeleri çıkarmak için geliştirilen bir algoritmadır. Metindeki kelimelerden ağırlıklı veya ağırlıksız graf oluşturularak kelimelerin skoru belirlenir. Bu skorlar sıralanıp belirlenen sayıda kelime seçilerek anahtar kelime olarak verilir.

TextRank algoritmasında graf oluşturulurken her kelime bir düğüm olarak belirlenir. Bu çalışmada her kelimenin Word2vec temsili düğüm olarak belirlenmiştir. TextRank algoritması için ağırlıklı bir graf oluşturulmuştur. Grafta kenarlara atanan ağırlıklar kelime vektörlerinin kosinüs benzerlikleri ile belirlenmiştir.

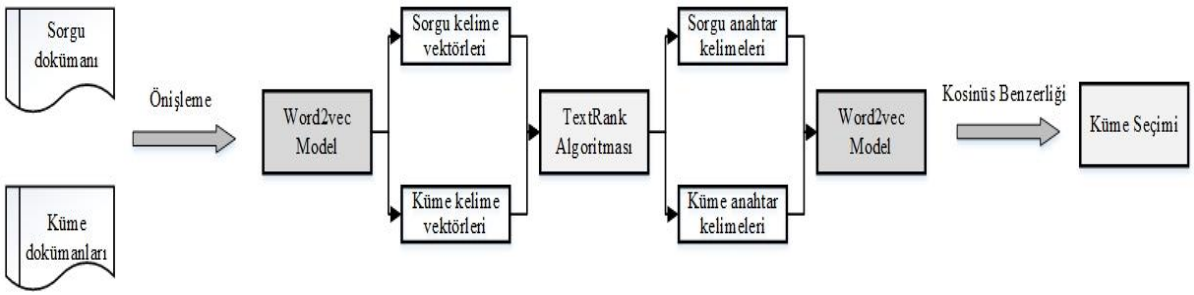
V_i ve V_k vektörleri arasındaki ağırlık W_{ik} aşağıdaki gibi hesaplanır:

$$W_{ik} = \frac{V_i \bullet V_k}{\|V_i\|\|V_k\|} = \frac{\sum_{j=1}^{n} V_{j,i} V_{j,k}}{\sqrt{\sum_{j=1}^{n} V_{j,i}^2}\sqrt{\sum_{j=1}^{n} V_{j,k}^2}} \quad (1)$$

Algoritmanın ikinci aşaması her kelimenin skorunu bulmaktır. V_i düğümü için skor aşağıdaki gibi belirlenir:

$$S(V_i) = (1-d) + d * \sum_{k \in In(V_i)} \frac{W_{ik}}{\sum_{V_j \in Out(V_k)} W_{kj}} S(V_k) \quad (2)$$

d, 0 ile 1 arasında ayarlanan bir parametredir. $Out(V_k)$, V_i düğümünden çıkan bağlantıların kümesi ve $In(V_i)$ ise V_i düğümüne gelen bağlantıların kümesidir. Bu formüle göre bir kelimenin önemi kaç kelime ile bağlantılı olduğuna ve bağlantılı olduğu kelimeler ile arasındaki kosinüs benzerliğine bağlıdır. Her kelimenin skoru bulunduktan sonra en yüksek skora sahip kelimeler anahtar kelime olarak belirlenir.



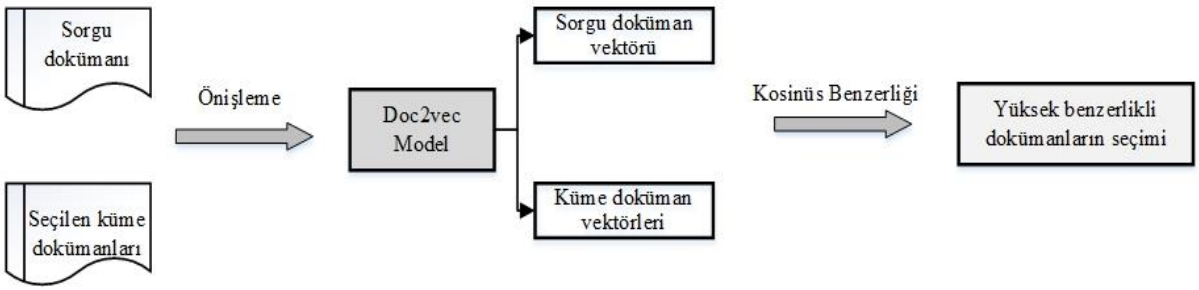
Şekil 1. Tavsiye kümesi seçimi

Küme seçimi işleminin aşamaları Şekil 1' de verilmiştir. Küme seçimi için kümelere ait anahtar kelime vektörü ve sorgu makalesine ait anahtar kelime vektörü kullanılır. İlk olarak bir kümede yer alan tüm dokümanların başlık ve özetinden anahtar kelimeleri çıkarılır. Word2vec modeli ile anahtar kelimeler vektörize edilir. Her dokümana ait anahtar kelime vektörlerinin ortalaması alınarak her kümeye ait anahtar kelime vektörü hesaplanır. Daha sonra sorguda yer alan makalenin başlık ve özeti kullanılarak anahtar kelimeleri çıkarılır. Bu anahtar kelimeler Word2vec modeli ile vektörize edilerek sorguya ait anahtar kelime vektörü elde edilir. Küme anahtar kelime vektörleri ile sorgu anahtar kelime vektörü arasındaki kosinüs benzerlikleri hesaplanır. En yüksek kosinüs benzerliğine sahip küme tavsiye kümesi olarak seçilir.

3.4. Makale Tavsiyesi

Kullanıcı tarafından girilen makaleye en benzer makalelerin bulunduğu küme belirlendikten sonra bu kümeden kullanıcıya tavsiye edilecek makalelerin belirlenmesi gerekir. Sadece anahtar kelime benzerliğine bakarak tavsiye pek verimli değildir. Tavsiye başarısını artırmak için bu aşamada dokümanların benzerliğine bakılmıştır. Doküman benzerliği için Doc2vec yöntemi kullanılmıştır. Bu şekilde dokümanların anlamsal özellikleri de dikkate alınarak benzerlikleri bulunmuştur.

Tavsiye işleminin aşamaları Şekil 2' de verilmiştir. İlk olarak kullanıcı tarafından girilen makalenin başlık ve özeti alınarak ön işleme uygulanmıştır. Sorgu makalesine ait doküman vektörü Doc2vec modeli ile elde edilmiştir. Daha sonra kümedeki her dokümana aynı işlem uygulanarak doküman vektörleri elde edilmiştir. Kullanıcı doküman vektörü ile kümedeki doküman vektörleri arasındaki kosinüs benzerlikleri hesaplanmıştır. Bulunan benzerlik skorları sıralanmış ve ilk 10 makale kullanıcıya tavsiye edilmiştir.



Şekil 2. Sorgu ile benzer tavsiye makalelerinin seçimi

4. Deneysel Sonuçlar

Bu bölümde önerilen makale tavsiye sisteminin performansını değerlendirmek için bazı testler yapılmıştır. Kullanılan yöntemlerin ve önerilen sistemin başarısı tartışılmıştır.

4.1. Veri Seti ve Model Oluşturma

Önerilen sistem bilgisayar bilimi, ekonomi, fizik gibi farklı alanlardan ya da belirli bir alanda farklı kategorilerden makaleleri tavsiye için kullanılabilir. Bu çalışmada örnek uygulama alanı olarak bilgisayar bilimi seçilmiştir. Farklı alanlardan makaleleri kümeleyip kullanıcıya istenilen konuda makale tavsiye etmek nispeten daha kolay olacaktır. Ancak belirli bir alandan makaleleri kategorilerine göre ayırıp kullanıcıya konusuna uygun makaleyi önermek daha zordur. Bu yüzden belirli bir alandaki makaleleri kullanarak önerilen sistemin performansı gösterilmek istenmiştir.

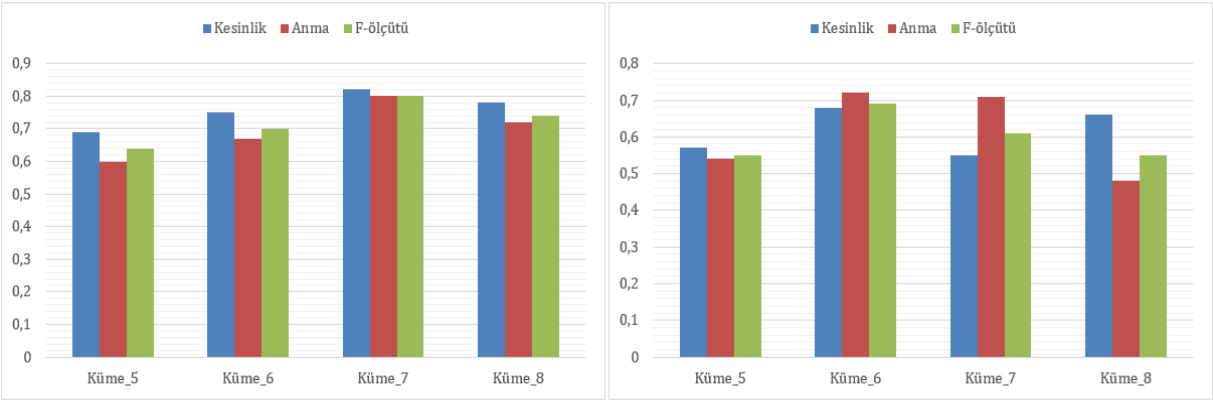
Veri setini oluşturmak için gerekli makaleler Arxiv platformundan alınmıştır. Arxiv; ekonomi, fizik, matematik, bilgisayar bilimi vb. birçok alandan makaleleri içeren açık-erişim bir depodur. Bilgisayar bilimi alanında birçok kategoriden makaleleri kapsamaktadır. Bu çalışmada yapay zeka, makine öğrenmesi, kriptoloji, yazılım mühendisliği, çoklu etmen sistemler, robotik, insan-bilgisayar etkileşimi, ağ, bilgisayar görmesi, bilgi çıkarımı ve sosyal bilgi kategorilerinden makaleler elde edilerek bir veri seti oluşturulmuştur. Bu veri seti eğitim ve test verisi olarak ayrılmıştır. Eğitim için toplam 92,325 makale, test için toplam 4,000 makale kullanılmıştır. Veri seti oluşturulurken makaleler kategorilerine göre elde edilmiştir. Ancak her makale tek bir kategoriye ait değildir. Çünkü bir makale birden fazla konu sentezlenerek oluşturulabilir. Örneğin, yapay zeka konusunda yazılmış bir makale aynı zamanda makine öğrenmesi ile ilgili olabilmektedir. Kategori konularının etkileşimi bir makalenin birden fazla kategoriye dahil edilmesine neden olur. Veri seti için ise ana kategori dikkate alınarak makaleler gruplandırılmıştır.

Önceki bölümde açıklandığı gibi çalışma için iki farklı model oluşturulmuştur: Kümeleme ve doküman benzerliği için Doc2vec modeli, anahtar kelime çıkarımı için Word2vec modeli. İlk olarak eğitim verisi kullanılarak Doc2vec modeli eğitilmiştir. Test verisi olarak belirlenen 4,000 makale kullanılarak her makaleye ait doküman vektörleri elde edilmiştir. Bu vektörler hem kümeleme hem de dokümanların benzerliğini bulmak için kullanılmıştır. Daha sonra eğitim verisi kullanılarak Word2vec modeli eğitilmiştir. Test verisindeki makalelerin anahtar kelimelerinin TextRank algoritması ile çıkarımında kullanılmıştır.

4.2. Kümeleme Performansı

Eğitilen Doc2vec modeli ile test verisindeki her dokümanın vektörü elde edilmiştir. Doküman kümeleme için Birch algoritması seçilmiştir. Bu algoritma K-means algoritması ile performans bakımından karşılaştırılmıştır. Test verisi üzerinde iki algoritma küme sayısı 5, 6, 7 ve 8 olarak belirlenip çalıştırılarak 4 farklı kümeleme elde edilmiştir. Kümeleme sonucunda aynı kategoride veya benzer kategoride olan makalelerin aynı küme içerisinde bulunması beklenmiştir. Kümeleme başarısını değerlendirmek için kesinlik, anma ve F-ölçütü kullanılmıştır. Kesinlik doğru tahmin edilen pozitif örnek sayısının toplam tahmin edilen pozitif örnek sayısına oranıdır. Anma doğru tahmin edilen pozitif örnek sayısının toplam pozitif örnek sayısına oranıdır. F-ölçütü ise bu iki değerin harmonik ortalamasıdır.

Şekil 3’ de iki algoritmanın farklı küme sayılarına göre performans karşılaştırması yer almaktadır. K-means algoritması ile elde edilen sonuçlara bakıldığında en yüksek başarı küme sayısı 6 iken elde edilmiştir. Birch algoritması ile en yüksek başarıya küme sayısı 7 iken ulaşılmıştır. En başarılı kümeleme için F-ölçütü değerleri karşılaştırıldığında Birch algoritması K-means algoritmasından daha iyi performans göstermiştir. Genel olarak kümeleme sonuçlarına bakıldığında Birch algoritması daha yüksek doğrulukta dokümanları kümelemiştir. Birch algoritmasında vektörlerin birbirlerine yakınlıklarına bakılarak kümeleme gerçekleştirilirken, K-means algoritmasında vektörlerin ilk küme merkezlerine olan yakınlıklarına bakılarak kümeleme gerçekleştirilir. Bu merkezlerin rastgele seçimi kümeleme sonucunun kalitesini büyük oranda etkilemektedir. Farklı merkezlerin seçimi farklı kümeleme sonuçlarına neden olmaktadır. Bu yüzden Birch algoritması K-means algoritmasına oranla daha yüksek performansa sahiptir. Sonuçlara bakıldığında en yüksek F-ölçütü değeri küme sayısı 7 iken elde edilmiştir. Bu yüzden dokümanlar 7 farklı kümeye ayrılarak çalışmanın diğer aşamaları gerçekleştirilmiştir.



(a) **(b)**

Şekil 3. Birch (a) ve K-means (b) algoritmalarının performans karşılaştırması

4.3. Anahtar Kelime Çıkarımı Performansı

Dokümanların anahtar kelimelerini bulmak için Word2vec ve ağırlıklı TextRank algoritması kullanılmıştır. Test veri setinden makaleler seçilmiş ve bu makalelerin anahtar kelimeleri çıkarılmıştır. Çıkarılan anahtar kelimeler ile makalenin anahtar kelimeleri karşılaştırılarak kesinlik, anma ve F-ölçütü değerleri hesaplanmıştır. Bu ölçütler denklem 3, 4 ve 5 olarak tanımlanır:

$$\text{Kesinlik} = \frac{\text{Algoritmanın çıkardığı doğru anahtar kelime sayısı}}{\text{Algoritmanın çıkardığı anahtar kelime sayısı}} \tag{3}$$

$$\text{Anma} = \frac{\text{Algoritmanın çıkardığı doğru anahtar kelime sayısı}}{\text{Makalenin anahtar kelime sayısı}} \tag{4}$$

$$\text{F} - \text{Ölçütü} = 2 \text{ x } \frac{\text{Kesinlik x Anma}}{\text{Kesinlik} + \text{Anma}} \tag{5}$$

Tablo 1. Anahtar kelime çıkarımı performans değerlendirmesi

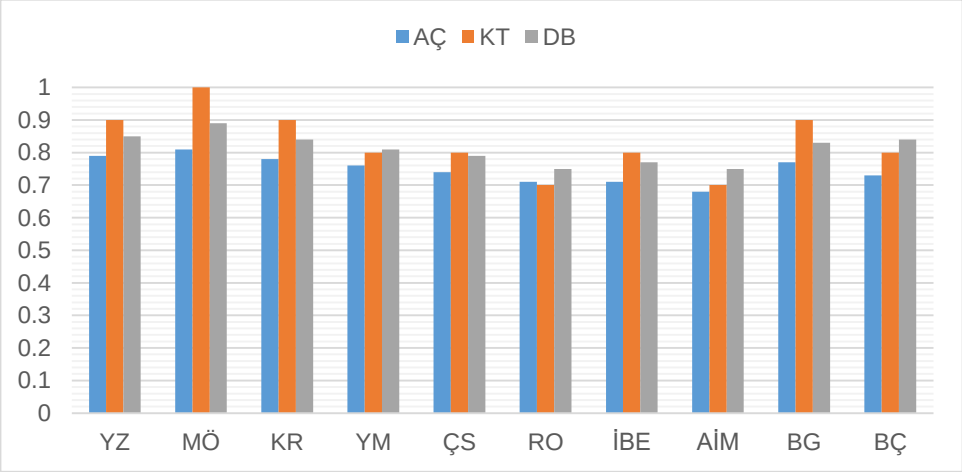
Makale sayısı	Kesinlik	Anma	F-ölçütü
100	0,68	0,78	0,72
200	0,73	0,75	0,74
300	0,75	0,70	0,72
400	0,76	0,68	0,71
500	0,76	0,69	0,72

Tablo 1'de farklı sayıda makalenin anahtar kelimelerinin çıkarımından elde edilen sonuçlar görülmektedir. Kesinlik, anma ve F-ölçütü değerleri makale gruplarının ortalama değerleridir. Anahtar kelime çıkarımında makalelerin başlık ve özet bilgileri kullanılmıştır. Oysa anahtar kelimeler tüm makalenin konusunu yansıtacak şekilde oluşturulur ve makalenin içeriği oldukça önemli bir kısımdır. Sadece başlık ve özetten anahtar kelime çıkarımı sonuçları etkilemiştir. Ancak buna rağmen sonuçlara bakıldığında kullanılan yöntem makale için bilgilendirici kelimelerin çıkarılmasında önemli bir başarıya sahiptir.

4.4. Makale Tavsiye Sisteminin Performansı

Önerilen sistemde kullanıcı tarafından girilen bir makale ismi alınarak bu makaleye benzer makalelerin tavsiyesi amaçlanmıştır. Araştırma makalesi tavsiye sistemini test etmek için 10 farklı

kategoriden makale içeren 10 kullanıcı sorgusu oluşturulmuştur. Kullanıcı sorgusunda yer alan makaleye en benzer 10 makale tavsiye edilmiştir. Bu kategoriler; yapay zekâ(YZ), makine öğrenmesi(MÖ), kriptoloji ve güvenlik(KR), yazılım mühendisliği(YM), çoklu etmen sistemleri(ÇS), robotik(RO), insan-bilgisayar etkileşimi(İBE), ağ ve internet mimarisi (AİM), bilgisayar görmesi ve örüntü tanıma (BG) ve bilgi çıkarımıdır (BÇ). Şekil 4' de sorgu makalelerine göre verilen 10 öneri için performans değerlendirilmektedir. Her kategori için, önerilen makalelerin kategori tahmin başarısı (KT), anahtar kelime çıkarma başarısı (AÇ), doküman benzerlik başarısı (DB) ortalama değerleri sunulmuştur. Kategori tahmin başarısı, önerilen 10 makaleden kaçının sorgu makalesi ile aynı kategoride olduğunu gösterir. Anahtar kelime çıkarma başarısı, önerilen 10 makale için F-ölçütü değeri ile ifade edilir. Doküman benzerlik başarısı, önerilen makaleler ile sorgu makalesi arasındaki benzerliklerin ortalamasıdır. Tüm kategoriler dikkate alındığında, tavsiye sistemi için ortalama kategori tahmin başarısı %83, anahtar kelime çıkarma başarısı %75 ve doküman benzerlik başarısı %81' dir.



Şekil 4. Makale tavsiye sisteminin performans değerlendirmesi

5. Sonuç

Tavsiye sistemleri son zamanlarda verinin artışı ile birlikte birçok alanda önemli bir yere sahip olmuştur. Bilimsel makale tavsiye sistemleri de akademik çalışmaların artışı ile birlikte ilgi çeken bir araştırma konusu haline gelmiştir. Her geçen gün yayınlanan makale sayısı artmakta ve istenilen konuya yönelik makalelere ulaşmak zorlaşmaktadır. Bu çalışmada konuya yönelik makale tavsiyesi için bir yöntem önerilmiştir. Bu yöntem ile sadece anahtar kelime benzerliklerinden ziyade makalelerin hem anlamsal hem de sözdizimsel benzerlikleri dikkate alınarak tavsiye gerçekleştirilmiştir.

Çalışmanın ilk aşamasında makaleler konularına göre kümelenmiştir. Böylece arama yapılacak makale havuzu daraltılmıştır ve arama daha hızlı bir şekilde gerçekleştirilmiştir. Ayrıca arama yapılan makaleye benzer makalelerin ortak konuda makale içeren bir kümede aranması daha yüksek benzerliğe sahip makalelere ulaşmayı sağlamıştır. Makale benzerliklerinin bulunmasında ve anahtar kelime çıkarımında derin öğrenme tabanlı yöntemlerin kullanılması tavsiye sisteminin başarısını artırmıştır. Bilgisayar bilimi alanında seçilen makalelerden oluşan bir veri seti ile çalışma gerçekleştirilmiştir. Yapılan deneyler sonucunda kullanıcı sorgusunda yer alan makaleye benzer makaleler yüksek bir başarı oranında önerilmiştir. Geliştirilen tavsiye sistemi kullanıcı hakkında bilgi sahibi olmadan araştırılan konuya yönelik makalelere kullanıcının daha hızlı ve doğru bir şekilde erişimini sağlamıştır.

Teşekkür

Bu çalışma Fırat Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından MF.20.09 numaralı proje kapsamında desteklenmiştir.

Kaynaklar

[1] Xia, F., Wang, W., Bekele, T. M., and Liu, H. (2017). Big scholarly data: A survey. IEEE Transactions on Big Data, 3(1), 18-35.

[2] Liu, H., Kou, H., Yan, C., and Qi, L. (2020). Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph. Complexity, 2020.

[3] Y. C. Lee et al., "Recommendation of research papers in DBpia: A Hybrid approach exploiting content and collaborative data," in 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings, 2017, pp. 2966–2971.

[4] Pan, L., Dai, X., Huang, S., and Chen, J. (2015). Academic paper recommendation based on heterogeneous graph. In Chinese computational linguistics and natural language processing based on naturally annotated big data (pp. 381-392). Springer, Cham.

[5] J. D. West, I. Wesley-Smith, and C. T. Bergstrom, (2016). "A Recommendation System Based on Hierarchical Clustering of an Article-Level Citation Network," IEEE Trans. Big Data, vol. 2, no. 2, pp. 113–123.

[6] Son, J., and Kim, S. B. (2018). Academic paper recommender system using multilevel simultaneous citation networks. Decision Support Systems, 105, 24-33.

[7] F. Xia, H. Liu, I. Lee, and L. Cao, (2016). Scientific Article Recommendation: Exploiting Common Author Relations and Historical Preferences, IEEE Trans. Big Data, vol. 2, no. 2, pp. 101–112.

[8] Bulut, B., Gündoğan, E., Kaya, B., Alhajj, R., and Kaya, M. (2020). User's research interests based paper recommendation system: A deep learning approach. In Putting Social Media and Networking Data in Practice for Education, Planning, Prediction and Recommendation (pp. 117-130). Springer, Cham.

[9] L. Steinert, and H. U. Hoppe, (2016). A comparative analysis of networkbased similarity measures for scientific paper recommendations. In 2016 Third European Network Intelligence Conference (ENIC) (pp. 17-24). IEEE.

[10] Q. Le, and T. Mikolov, (2014). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196).

[11] Gündoğan, E., and Kaya, M. (2019, September). Evaluation of Session-Suitability of Papers in Conference Programs. In 2019 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-5). IEEE.

[12] Shirkhorshidi, A. S., Aghabozorgi, S., Wah, T. Y., and Herawan, T. (2014). Big data clustering: a review. In International conference on computational science and its applications (pp. 707-720). Springer, Cham.

[13] Lorbeer, B., Kosareva, A., Deva, B., Softić, D., Ruppel, P., and Küpper, A. (2018). Variations on the clustering algorithm BIRCH. Big data research, 11, 44-53.

[14] Xia, X. (2020). Clustering Analysis of Interactive Learning Activities Based on Improved BIRCH Algorithm. arXiv preprint arXiv:2010.03821.

[15] Wang, H., Ye, J., Yu, Z., Wang, J., and Mao, C. (2020). Unsupervised keyword extraction methods based on a word graph network. International Journal of Ambient Computing and Intelligence (IJACI), 11(2), 68-79.

[16] Firoozeh, N., Nazarenko, A., Alizon, F., and Daille, B. (2020). Keyword extraction: Issues and methods. Natural Language Engineering, 26(3), 259-291.

[17] Bharti, S. K., and Babu, K. S. (2017). Automatic keyword extraction for text summarization: A survey. arXiv preprint arXiv:1704.03242.

[18] Qingyun, Z., Yuansheng, F., Zhenlei, S., and Wanli, Z. (2020). Keyword extraction method for complex nodes based on TextRank algorithm. In 2020 International Conference on Computer Engineering and Application (ICCEA) (pp. 359-363). IEEE.

[19] Pan, S., Li, Z., and Dai, J. (2019). An improved TextRank keywords extraction algorithm. In Proceedings of the ACM Turing Celebration Conference-China (pp. 1-7).