GÜFBED/*GUSTIJ* (2019) 9 (3): 536-546
DOI: 10.17714/gumusfenbil.514154 Araştırma Makalesi / *Research Article*

Eski Dilde Kullanılan Sözcükler Arasındaki Anlamsal Yakınlıkların Doğal Dil İşleme Yöntemleriyle Tespiti

Using NLP Methods for the Discovery of Semantic Similarities between Words in Old Turkish Language

Mustafa CANIM*
IBM Thomas J. Watson Research Center P.O. Box 218, Yorktown Heights, New York, U.S.A.

• Geliş tarihi / *Received*: 17.01.2019 • Düzeltilerek geliş tarihi / *Received in revised form*: 23.04.2019 • Kabul tarihi / *Accepted*: 07.05.2019

Öz
Makina öğrenme tekniklerinin doğal dil işleme alanında kullanımı son yıllarda oldukça popüler bir çalışma konusu haline gelmiştir. Doğal dil işleme yöntemlerinin yabancı dillerdeki birçok uygulamasına rastlanılmasına rağmen Türkçe ve özellikle eski dil metinlerdeki uygulamaları oldukça yetersizdir. Bu alandaki eksikliğin giderilmesine yönelik olarak yapılan bu çalışmada, Kültür Bakanlığı kaynaklarından elde edilen 32000 sayfa doküman, temizleme işleminden geçirildikten sonra, bu metinlerden elde edilen kelimeler üzerinde iki katmanlı bir sinir ağı modeli çalıştırılmıştır. Pencere boyutu, uzay boyutu, örnekleme miktarı gibi birçok farklı parametre ile geliştirilen modellere ait vektör uzayları bir sunucuya kopyalanarak bir sorgulama sistemi ve RESTful API servisleri oluşturulmuştur. Ayrıca bu sorgulama sisteminin doğrudan kullanabilmesi için bir kullanıcı portalı oluşturularak RESTful API ile beraber internet kullanımına açılmıştır. Yapılan bu çalışmanın iki farklı amaçla kullanılması hedeflenmektedir. Birinci hedef bu sistemin Türk Dil Kurumu ve Kültür Bakanlığı gibi kurumların ve diğer eski dil sözlük hizmeti sağlayan şirketlerin internet sitelerine entegre edilmesi ve aratılan sözcüklere yakın terimlerin kullanıcılara getirilmesidir. İkinci hedef ise tarih ve edebiyat gibi eski dilin kullanıldığı bilimsel çalışmalarda metinlerin günümüz Türkçe'sine çevrilmesi esnasında ortaya çıkan hataların azaltılmasıdır.

Anahtar kelimeler: Doğal dil işleme, Kelime simgeleri, NLP, Yapay sinir ağları

Abstract
Leveraging machine learning techniques in NLP domain has been a very hot research field due to the advancements in artificial intelligence area. Despite the popularity of this field, there is no known study on application of ML techniques on old Turkish language. This study aims to fill in this gap where 32000 pages of text has been downloaded from the websites of Ministry of Culture and a two-layer neural network model has been built on top of them to discover the semantic similarities between Turkish words in old Turkish language. The algorithm has been run with different parameters such as window size, dimension size, sampling size etc. and the produced vector spaces are uploaded into public servers for the purposes of enabling a RESTful API based query interface. Also a web UI has been created to provide a querying mechanism for regular users. The services that are developed can be used for two different purposes. One of them is to integrate these services into existing old Turkish language dictionary websites that are made available by third party providers as well as other institutions such as Ministry of Culture and Turkish Language Institution. Secondly, the developed services are intended to be used for mitigating the translation errors made during the translation of old Turkish texts into modern Turkish language in the areas of history and Turkish literature. Also enabling these services for public use will encourage other researchers to pursue this academic work and compare their results with the experimental results presented in this paper to make further improvements in this field.

Keywords*: Artificial neural networks, Natural language processing, NLP, Word embeddings*

* Mustafa CANIM, mustafacanim@gmail.com, Tel: (+1 914) 707 18 91, orcid.org/0000-0002-3653-267X

1. Giriş

Makina öğrenme tekniklerinin doğal insan dilinin işlemesinde kullanılması son yıllarda büyük ölçüde popülerlik kazanmıştır. İnternet sitelerinde geçen metinlerin hangi dilde yazılmış olduğunun otomatik olarak tespiti, gereksiz e-maillerin filtrelenmesi veya arama terimlerinin otomatik olarak tamamlanması bu uygulamalardan yalnızca bazılarına örnek olarak gösterilebilir (Soon vd., 2001; Kaya ve Ertugrul, 2016; Marrero ve Urbano 2018; Biemann vd., 2018). Bu tür metinlerin incelenmesinde en çok kullanılan yöntemlerden biri metin sınıflandırmasıdır. Çok büyük ölçekli metin tasniflerinde gözetimli öğrenme yöntemlerinin uygulanması sınıfların çokluğu nedeniyle oldukça zahmetlidir. Bu tür metinlerin sınıflandırılmasının büyük ölçekli yapılabilmesi için yakın zamanda Word2Vec isimli yöntem önerilmiştir (Mikolov vd., 2013; Church 2017). Önerilen bu yöntemde dilde bulunan her bir sözcük uzayda bir vektöre dönüştürülür. Bu vektörler aynı zamanda kelime simgeleri (Word Embeddings) olarak adlandırılır. Bu işlem sonrasında ortaya çıkan kelime simgeleri, kelimeler arasındaki ilişkilerin tespit edilmesinde kullanılır.

Öte yandan tarih ve Türk dili ve edebiyatı gibi bilimsel sahalarda eski dilde yazılmış olan eserlere sıkça başvurulmaktadır. Günümüz Türkçe'sinde kullanılmayan birçok sözcük bu metinlerde kullanıldığı için metinlerin okunmasında zorluklar yaşanmaktadır. Bu amaçla internette mevcut birçok Osmanlıca sözcük sitesi bulunmaktadır. Bu siteler her ne kadar aratılan sözcükleri kullanıcıya getirse de anlamsal olarak yakın sözcüklerin kullanıcıya getirilmesi gibi bir hizmet sağlamamaktadır. Bir diğer problem ise eski dilde yazılmış olan metinlerin günümüz diline çevrilmesi esnasında ortaya çıkan sorunlardır. Bu süreçte karşılaşılan en önemli problemlerden birisi, çeviri esnasında karşılaşılan bazı kelimelerin eski metinlerdeki deformasyon sebebiyle el yazması metinden okunamamasıdır. Okunmasında zorlanılan kelimenin etrafında bulunan sözcüklerin bir sisteme girilmesi durumunda bu kelimelere en yakın kelimeleri öneren bir sistemin geliştirilmesinin çeviri hatalarını azaltmada çok ciddi yardımı olacaktır. Bu çalışmada geliştirilen sistem sayesinde eski dil metinlerde geçmekte olan kelimeler arasındaki ilişkiler matematiksel olarak tespit edilebilmekte ve verilen sözcükler için yakın sözcüklerin kullanıcıya getirilmesi sağlanmaktadır.

Bu çalışmanın yapılabilmesi için Kültür Bakanlığının sitesinden "Mesneviler" kategorisinde 35, "Divanlar" kategorisinde ise 54 kitap sunuculara indirilmiştir. PDF formatında ve Latin harflerle yazılmış olan bu metinler öncelikle temizleme işlemine tabi tutularak yaklaşık 32000 sayfa metin işlenmiştir. Elde edilen bu metin çıktılarında kullanılan her bir kelime uzayda bir vektör olarak temsil edilerek iki katmanlı bir sinir ağı modelinin geliştirilmesinde kullanılmıştır. Farklı parametrelerle üretilen vektör uzayları bir sunucuya kopyalanarak bir sorgulama (query) sistemi ve de RESTful API servisleri oluşturulmuştur. Bu sorgulama servislerine kolay erişim için bir internet portalı oluşturulmuştur. Hiper parametrelerin uzay vektörleri üzerindeki etkilerinin incelenmesi için bir deney düzeneği oluşturulmuştur. Yapılan deneylerde 200 ve 400 farklı boyutlu uzaylar arasında ciddi bir fark olmadığı, Skip-gram algoritmasıyla elde edilen uzaylarda yapılan sorgularda CBOW algoritmasına kıyasla daha yüksek cosine yakınlığı tespit edilmiştir. 5 kelimeden oluşan pencere boyutunun 8 kelimeye kıyasla çok daha iyi sonuçlar ortaya koyduğu ve de sıkça kullanılan sözcüklerin sayısı düştüğünde daha iyi sonuçlar elde edildiği gözlemlenmiştir. Köklerine ayırma işleminin örnekleme boyutunda olduğu gibi uzaylardaki vektör sayısını düşürdüğü için sorgulanan sözcükler için cosine mesafesini düşürücü etkisi olduğu gözlemlenmiştir. Eski dilde kullanılan sözcükler için kelime kök ayrıştırıcı geliştirilmesi durumunda ve daha fazla terimden oluşan veri kümelerinde işlem yapıldığı takdirde daha olumlu sonuçlar elde edileceği sonucuna varılmıştır.

Bu çalışmada geliştirilen servislerin tarih ve Türk dili ve edebiyatı sahasında başvurulan bir kaynak olması hedeflenmektedir. Osmanlıca terimler sözlüklerinde girilen sözcüklere anlamca en yakın olan kelimelerin kullanıcılara önerilmesi sayesinde kullanıcıların kelimeler arasındaki ilişkileri öğrenmesi amaçlanmaktadır. Ayrıca literatürde benzeri bulunmayan bu çalışmanın bu sahada ve özellikle Türkçe dili üzerinde doğal dil işleme sahasında daha farklı çalışmaların yapılmasında öncü olacağı düşünülmektedir. Eski dil metinlerde geçen kelimeler için geliştirilecek olan kelime kökü ayrıştırıcı sistemlerin geliştirilmesi gibi çalışmaların yapılması da bu şekilde teşvik edilecektir. Bu anlamda erişime sunulan REST servisleri sayesinde bu alanda çalışma yapmayı planlayan araştırmacıların kendi sonuçlarını geliştirilen sistemle karşılaştırabilmesi hedeflenmektedir.

1.1. Daha önce yapılan çalışmalar

Eski dil metinler üzerinde yapılan çalışmaların büyük çoğunluğu, anlamsal benzerlikten çok, harflerin tanınması için gerekli olan resim işleme tekniklerine dayanmaktadır. Ataer ve Duygulu (2007), Osmanlıca metinlerin resim işleme yoluyla getirilmesi (retrieve) için yöntemler önermişlerdir. Öztürk ve Özbay (2000), farklı fontlardaki eski dil metinlerdeki karakterlerin yapay sinir ağlarıyla tanınmasına yönelik bir çalışma yapmıştır. Arifoğlu ve Duygulu (2011), eski dil metinlerdeki kelimelerin resim işleme yoluyla getirilmesi için iki farklı yöntem önermişlerdir. Bu çalışmada, Fuzuli Divanından Leyla ile Mecnun bölümünden 10 sayfa veri işlemede kullanılmıştır. Adıgüzel vd. (2012) eski dil metinlerde ikileştirme (binarization) ve çizgi segmentasyonunu (line segmentation) çalışmışlardır. Kılıç vd. (2008) bu çalışmalarında segmentasyon (segmentation) ve normalizasyon (normalization) yapabilen, kenarların tespitini sağlayan (edge identification) bir optik karakter tanımlama sistemi (OCR) önermiş bulunmaktalar.

Bu çalışmaların dışında Türkçe diline yönelik doğal dil işleme konusu altında yapılmış olan çalışmalar mevcuttur. Bilgin vd. (2017) Türkçe dili için daha etkili bir bağlılık ayrıştırması yöntemi önermiştir. Kalender vd. (2018) THINKER ismini verdikleri Türkçe diline özgü bir nesne ismi bağlama sistemi (entity linking) önermekteler. Kılınç vd. (2017) diğer dillere kıyasla Tükçe diline yönelik olarak metin sınıflandırması konusunda yeteri kadar veri kaynakları bulunmaması nedeniyle bu alanda bir çalışmakta yapmış ve bu verileri genel kullanıma sunmuşlardır. İlgen vd. (2016) Türkçe diline özgü kelime anlam belirsizliklerinin giderilmesi için kullanılan özellikler kümelerini çalışmışlardır. Yan yana bulunan ve kelime paketi gibi özelliklerin hangilerinin Türkçe için daha uygun olduğunu incelemişlerdir. Yapılan diğer bir çalışmada Türkçe dilindeki anlamsal eklerin gözetimsiz öğrenme yoluyla tespitine yönelik bir yöntem önerilmiştir (Can, 2017).

Türkçe dili dışında genel olarak kelimeler arasındaki ilişkilerin tespit edilmesi konusunda aşağıdaki çalışmalar bulunmaktadır. Zhu vd. (2017) bilgi ağlarında geçen terimler arasındaki anlamsal ilişkilerin tespiti için "wpath" metodu önermiştir. Li vd. (2015) büyük ölçekli bilgi ağlarında bulunan kelimeler arasındaki ilişkilerin ortaya çıkarılması için hızlı çalışan metotlar önermişlerdir. Han vd. (2013) kelimeler arasındaki benzerliklerin tespitinde sıkça kullanılmakta olan PMI (Pointwise mutual information) için geliştirilmiş bir yöntem önererek eş anlamlı kelimelerin tespiti ve benzer kelimelerin bulunmasında kolaylık sağlamayı amaçlamaktalar.

Yukarıda bahsedilen çalışmalara bakıldığı zaman eski dilde kullanılan sözcüklere yönelik anlamsal ilişkilerin tespiti ile ilgili bir çalışmaya rastlanmamıştır. Bu anlamda, yapılan bu çalışmanın akademik ortamda benzeri çalışmalar için öncü olması hedeflenmektedir.

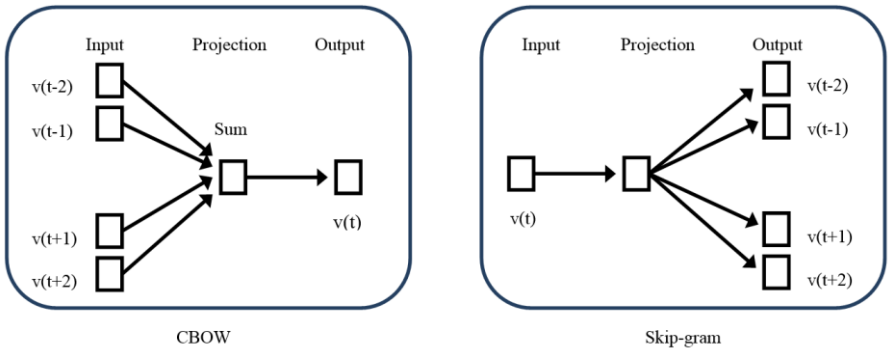
2. Materyal ve Metot

Eski dilde sıkça kullanılan sözcüklerin kelime uzayındaki vektörlere dönüştürülmesi için son yıllarda doğal dil işleme sahasında çok popüler hale gelen ve iki katmanlı bir sinir ağı modeli olarak geliştirilen Word2Vec algoritması kullanılmıştır. Gözetimsiz öğrenme mantığıyla çalışan bu algoritmanın detayları aşağıda anlatılmaktadır.

2.1. Yapay Sinir Ağlarının Doğal Dil İşlemede Kullanımı

Word2Vec doğal dilde yazılmış olan metinlerin işlenmesini sağlayan iki katmanlı bir sinir ağı modelidir. Google çalışanlarından Tomas Mikolov ve takım arkadaşları tarafından geliştirilen bu algoritma girdi olarak büyük metinleri işleyerek bir vektör uzayı oluşturur (Mikolov vd., 2013). Ortaya çıkan bu vektör uzayı yüzlerce boyuttan oluşur (genellikle 200 ile 400 boyut aralığı tercih edilir) ve girdi olarak işlenen metindeki her bir sözcük bu uzayda bir vektör ile temsil edilir. Dolayısıyla her bir kelime için ayrı özellik vektörü (feature vector) oluşturulmuş olur. Bu algoritmanın doğal dil işlemede popüler hale gelmesindeki en büyük etken yakın anlamlardaki kelimeleri temsil eden vektörlerin birbirilerine mesafe olarak yakın olmasını sağlamasıdır. Bu bir anlamda kelimeler arasındaki anlam ilişkilerinin matematiksel olarak tespit edilmesi anlamına gelir. Kelimeleri temsil eden vektörler metin içerisinde anlam itibariyle birbirine ne kadar yakın ise bu uzayda da birbirilerine o kadar yakınlık gösterirler. Örnek olarak milyonlarca sayfadan oluşan bir metin kümesinde 'İngiltere', 'Fransa', 'Elma' ve 'Portakal' kelimeleri sıkça kullanılmakta olsun. Bu sayfalarda geçen cümleler yapay sinir ağı algoritması ile işlendiği zaman ortaya bir vektör uzayı çıkacaktır. Bu uzayda 'İngiltere' ve 'Fransa' kelimelerini temsil eden vektörler arasında bir yakınlık olması yanı sıra 'Elma' ve 'Portakal' kelimelerini temsil eden

vektörler arasında da bir yakınlık tespit edilir. Bu yöntem sayesinde bunun gibi sözcüklerin metinlerde geçmekte olan yerleri arasındaki yakınlıklara bakılarak kelimeler arasındaki anlamsal yakınlıklar tahmin edilmeye çalışılır. Bu nedenle bu yöntem sadece tek bir dildeki metinlerin işlenmesi için tasarlanmış bir model değildir. Bu yöntem bir derin sinir ağı olmamakla beraber girdi olarak girilen metinleri rakamsal olarak temsil edilebilir hale getirerek derin sinir ağları tarafından işlenmesine olanak sağlar. Bu algoritmanın uygulama alanı metinlerin işlenmesi ile kısıtlı değildir. Örnek olarak genler, müzik listeleri, kodlar ve sosyal medya ağları gibi alanlarda benzerliklerin tespit edilmesinde de kullanılır.



Şekil 1. CBOW ve Skip-gram algoritmaları

Yapay sinir ağı ile üretilen vektörlere kelime simgesi ismi verilir. Kelime simgeleri üretilirken iki farklı yöntem kullanılır. Bunlar Skip-gram ve CBOW yöntemleridir. Kelime simgelerinin öğrenimi gözetimsiz öğrenme (unsupervised learning) olduğu için Skip-gram ve CBOW (Continuous bag of words) yöntemleri kullanılarak bir çeşit öğrenme etiketleri üretilir. CBOW yönteminde verilen kelimeler için bir kelime tahmini yapılmaya çalışılırken Skip gram tekniğinde verilen bir kelimeden bu kelimenin çevresindeki kelimeler tahmin edilmeye çalışılır (Bu iki yöntem Şekil 1'de gösterilmektedir). CBOW yöntemi daha çok küçük veri kümeleri için etkili olurken Skip-gram yöntemi daha çok büyük veriler için tercih edilmektedir. Skip-gram modelinde verilen sözcük için etrafında bulunan sözcüklerin bulunmasında aşağıdaki öğrenme hedefi uygulanır (Mikolov vd., 2013). Verilen bir kelime dizisi $w_1, w_2, w_3, \ldots, w_T$ için Skip-gram modelinde eşitlik 1'deki ortalama log ihtimalinin maksimize edilmesi hedeflenir.

$$\frac{1}{T} = \sum_{t=1}^{T} \sum_{-c \le j \le c, j \neq 0} log\, p(w_{t+j} / w_t) \qquad (1)$$

Eşitlik 1'de c parametresi öğrenme kontex boyutunu belirtir ve bu parametrenin büyüklüğüne göre öğrenme esnasında yapılan işlem miktarı değişir. Skip-gram'ın sade halinde $p(w_{t+j} | w_t)$ aşağıdaki softmax hesaplaması kullanılır:

$$p(w_O | w_I) = \frac{\exp({v'_{wO}}^{\top} v_{wI})}{\sum_{w=1}^{W} \exp({v'_w}^{\top} v_{wI})} \qquad (2)$$

Eşitlik 2'de v_w ve v'_w w'nin sinir ağındaki vektörel girdi ve çıktısını belirtir. W ise uzaydaki tüm kelimelerin boyutunu belirtir. Çok büyük uzaylarda W çok büyük olduğu için bu hesaplama işlemi zorlaşır ve bu nedenle tam softmax hesaplaması yerine hiyerarşik softmax fonksiyonu tercih edilir (Mikolov vd., 2013).

2.2. Gözetimsiz Öğrenme Algoritmasının Eski Dil Metinlere Uygulanması

Gözetimsiz öğrenme metotlarının eski dil kullanılan metinlere uygulanması birkaç aşamadan oluşmaktadır. İlk aşamada bu metinler bilgisayar ortamına aktarılarak dijital formata çevrilir ve bilgisayar programlarıyla işlem yapılabilir hale getirilir. İkinci aşamada elde edilen bu metinler üzerinde öğrenme metotlarının uygulanması (training) ve farklı vektör uzaylarının oluşturulması işlemi gerçekleştirilir. Üçüncü aşamada ise verilen herhangi bir sözcüğün bu uzaylarda aranması ve kendisine "cosine" mesafesi olarak en yakın sözcüklerin sorgulanması gerçekleştirilir. Aşağıda bu çalışmada uygulanan bu işlemlerin her bir aşaması detaylı bir şekilde anlatılmaktadır.

2.3. Metinlerin Elde Edilmesi ve Temizlenmesi

Bu çalışmanın yapılabilmesi için gerekli olan metinlerin öncelikli olarak bilgisayar ortamına aktarılması ve de bilgisayar programları tarafından işlenebilir hale getirilmesi gerekmektedir. Bu aşamada metnin orijinal haline bağlı olarak farklı yöntemler uygulanır. Eğer kullanılmak istenen içerik basılı kitap halinde mevcut ise bu kitapların öncelikle tarayıcılar (scanner) aracılığıyla resim olarak bilgisayar ortamına aktarılması ve daha sonra resimden metine dönüştürme (OCR) teknikleri kullanılarak işlenebilir metin dosyası (text) formatına dönüştürülmesi gerekmektedir. Kullanılan alfabe eğer Latin alfabesi değil ise bir sonraki aşamada bu kelimelerin Latin alfabesi formatına dönüştürülmesi gerekmektedir. Eğer Latin harfleri şeklinde yazılmış olan bu içerik PDF dosya formatı şeklinde bilgisayar ortamına aktarılmış ise bu durumda bu çalışmada olduğu gibi bu dosya formatındaki içeriğin öncelikle text formatına dönüştürülmesi ve bu dönüştürme işlemi esnasında ortaya çıkan kullanışsız karakter ve kelimelerin (noise) bu dosyalardan temizlenmesi gerekir.

Yapılan bu çalışmada Türkiye Cumhuriyeti Kültür Bakanlığının sitesinden (URL-1, 2016) "Mesneviler" kategorisinde 35 "Divanlar" kategorisinde ise 54 kitap sunuculara indirilmiştir. PDF formatında ve Latin harflerle yazılmış olan bu metinler öncelikle temizleme işlemine tabi tutularak yaklaşık 32000 sayfa metin işlemden geçirilmiştir. Bu sayfaların 22825 sayfası Divanlar klasöründen 9456 sayfası ise Mesneviler kategorisindeki kitaplardanlar oluşmaktadır. PDF formatından "txt" formatına dönüştürme yapılırken sayfa numaraları gibi kısımların işlenmesi esnasında anlam ifade etmeyen bazı karakterler ortaya çıkmaktadır. Python programlama dili ile yazılan bir program sayesinde 89 dosyanın her birisi işlenerek temizleme yapılmıştır. Daha sonra tüm dosyalardan elde edilen içerik birleştirilerek dört faklı formata dönüştürülmektedir. Bu formatların birincisinde tüm kelimeler, SnowBall Stemmer (URL-2, 2018) kullanılarak köklerine indirgenmektedir. İkincisinde ise kelimenin orijinal hali kullanılmaktadır. Diğer iki parametre ise bu kelimelerin küçük harflere dönüştürülmesi veya orijinal halinin korunması üzerinedir. Dolayısıyla bu formatlar özetlenecek olursa: "Stemmed, lower case", "Stemmed, upper case", "Unstemmed, lower case", "Unstemmed, upper case" şeklindedir. Word2Vec uygulamalarında köklerine ayırma ve küçük harflere dönüştürme işlemi literatürdeki çalışmalarda da test edilen yöntemlerdendir (Basu vd., 2017). Köklerine ayırma işlemi için kullanılan SnowBall Stemmer Türkçe dilini de desteklemektedir (URL-4, 2019). Bu işlem esnasında SnowBall Stemmer, eski dilde kullanılan terimlerin yüzde 64'ü üzerinde kök ayrıştırma işlemi gerçekleştirmiştir. Eski dilde sıkça kullanılan sözcükler üzerine bir kök ayrıştırıcının geliştirilmesi ve bu çalışmaya uygulanması durumunda daha etkili sonuçlar elde edilecektir.

2.4. Öğrenme Aşaması ve Vektör Dosyalarının Oluşturulması

Sayfalardan ortaya çıkarılan bu dört farklı metin dosyası üzerinde gözetimsiz öğrenme algoritmalarının uygulanabilmesi için dosyalar öncelikle sunuculara kopyalanmıştır. Bulut sistemi (cloud computing) şirketlerinden bir sanal makine bu amaçla kullanılmak için kiralanmıştır. Makine öğrenmesi aşamasında kullanılan yapay sinir ağı algoritmasının kaynak kodu Google'ın kaynak kodu arşivinden sunuculara indirilmiştir (URL-3, 2019). Derleme işlemi için kaynak kod dosyası içinde gelen komut dosyalarında değişiklikler yapılmıştır. 4 farklı girdi dosyası üzerinde 3 farklı parametre ile öğrenme algoritması tekrar tekrar koşturularak 32 farklı vektör dosyası oluşturulmuştur. Bu parametre aralıkları Word2Vec kütüphanesinin indirilebildiği sitede belirtilmiştir (URL-5, 2019). Değiştirilen parametrelerden birisi vektör boyutlarıdır (dimension) ve bu boyutlar 200 ile 400 olarak test edilmiştir. Diğer bir parametre, pencere boyutudur (window size) ve bu parametre 5 ve 8 olarak değiştirilmiştir. Üçüncü parametre ise örnekleme boyutudur ki bu parametre 1e-3 ve 1e-5 olarak değiştirilmiştir. Öğrenme işlemi sonucunda ortaya toplamda 32 (4x2x2x2) farklı vektör dosyası çıkmıştır ve bu vektör dosyaları yaklaşık olarak diskte 3 GB yer kaplamaktadır. Ortaya çıkan vektör uzayları bir sunucuya kopyalanarak bir sorgulama sistemi ve de RESTful API servisleri oluşturulmuştur.

2.5. Vektör Dosyalarının Sorgulama Amaçlı Kullanımı

Ortaya çıkan vektör dosyalarında uzaydaki her bir kelime bir vektör ile temsil edilmekte olup bu uzay 200 ve 400 boyuttan oluşmaktadır. Verilen bir kelime için benzer sözcüklerin bulunması amacıyla vektörler arasındaki cosine mesafesi hesaplanmaktadır. Word2Vec kütüphanesiyle beraber gelen bir uygulamada verilen kelimelere cosine mesafesini hesaplayıp en yakın 40 vektöre

denk gelen kelimeleri listeleyen bir program mevcuttur. C dilinde yazılmış olan bu programda değişiklikler yapılarak REST API ile uyumlu çalışır hale getirilmiştir.

2.6. RESTful API Tabanlı Servislerle Sorgulama

Kullanıcıların geliştirilen sisteme erişebilmesi için iki yöntem sağlanmaktadır. Bunlardan birincisi oluşturulan RESTful API tabanlı servisler üzerinden diğeri ise internet arayüzü üzerinden sağlanmaktadır. REST servisleri ile erişim sağlanılmasının iki amacı bulunmaktadır. Bunlardan birincisi bu alanda çalışmalar yapmayı planlayan araştırmacıların kendi sonuçlarını geliştirilen bu sistemle karşılaştırabilmeleri için bir imkân sağlamak ve bu şekilde bu sahada yapılacak olan yeni çalışmaların teşvik edilmesidir. Diğer bir amaç ise Osmanlıca sözlük hizmeti veren farklı internet sitelerinden bu servislere kolay erişim sağlanması ve bu şekilde kullanıcılara aradıkları sözcüklere yakın sözcüklerin gösterilmesidir.

Tablo 1. REST servisi URL parametreleri

Parametre numarası	**Parametre açıklaması**	**Kabul edilen birinci değer**	**Kabul edilen ikinci değer**
Param1	Kelimenin köklerine ayrılıp ayrılmamış olması	stemmed	unstemmed
Param2	CBOW veya Skip-gram algoritmaları	cbow	skip
Param3	Vektör boyutu	200	400
Param4	Pencere boyutu	5	8
Param5	Örneklendirme boyutu	3	5
Param6	Sorgulanan kelime	Sözcük	Sözcük

Sistemin sorgulanabilmesi için GET isteği yapılan URL üzerinden değişkenlerin sunucuya aktarılması gerekmektedir. Bu parametreler Tablo 1'de listelenmektedir ve sorgulama linki şu şekilde oluşturulur: "http://HOST_URL/ kelime/ listele/ param1/ param2/ param3/ param4/ param5/ param6". Parametre 1 kelimelerin köklerine ayrılmış olan vektör veri tabanında aranıp aranmayacağını belirtmektedir. Parametre 2 hangi çeşit algoritma ile üretilen vektör veri tabanını belirtmektedir. Parametre 3 Vektör boyutunu, parametre 4 pencere boyutunu, parametre 5 ise örneklendirme boyutunu ifade etmektedir. Son olarak parametre 6 ise sunucuda aratılması istenen sözcüğü belirtmektedir. HOST_URL şuan itibariyle bu IP üzerinden işlem sağlamaktadır: 35.163.160.151:8081. Bu URL'de herhangi bir değişiklik olması durumunda son güncel IP adresi http://kelime.site adresinden elde edilebilir. Örnek olarak aşağıdaki URL'e verilen parametreler gönderildiği takdirde 'mihver' kelimesi sunucu tarafından sorgulanmaktadır:

http://HOST_URL/kelime /listele/ unstemmed/skip/400/8/5/mihver

Bu örnekte sözcük, köklerine ayrıştırma yapılmayan 400 boyutlu uzayda aranmaktadır. Bu uzaydaki vektörler Skip-gram yöntemiyle oluşturulmuş, pencere boyutu 8 ve aynı zamanda örnekleme boyutu da 1e-5 olarak belirtilmiştir.

RESTful API üzerinden yapılan GET istekleri sonucunda kullanıcıya bir JSON nesnesi gönderilir. Bu JSON nesnesi altında üç alt nesne mevcuttur. Bunlar: "links", "scores" ve "scoresRanked" şeklinde sıralanmaktadır. "Links" başlığı altında sorgulanan sözcükle alakalı 40 kelime kullanıcıya gönderilir. Bununla beraber bu ilişkili kelimelerin tekrar aynı parametrelerle sorgulanmasını sağlayan linkler mevcuttur. Chrome veya Firefox gibi tarayıcılar vasıtasıyla bu JSON nesnesi görüntülendiği takdirde bu linklere tıklayarak kullanıcılar JSON nesnesinin detaylarını inceleyebilir. Fakat JSON nesnesinin düzgün bir formatta görüntülenebilmesi için JSON görüntüleyici gibi tarayıcı eklentilerinden birinin kurulmuş olması gerekmektedir. Aşağıdaki örnekte gösterildiği gibi bu JSON nesnesi içerisinde "scoresRanked" başlığı altında her bir kelime için verilen cosine mesafesi yakınlığı ve bu kelime ile tekrar sorgulama yapılmasını sağlayan link mevcuttur.

```
scoresRanked": [
      "muhit",
      "0.874222",
```

"http://35.163.160.151:8081/kelime/listele/unstemmed/skip/400/8/5/muhit"
],[
"rubâ‘î",
"0.873617",

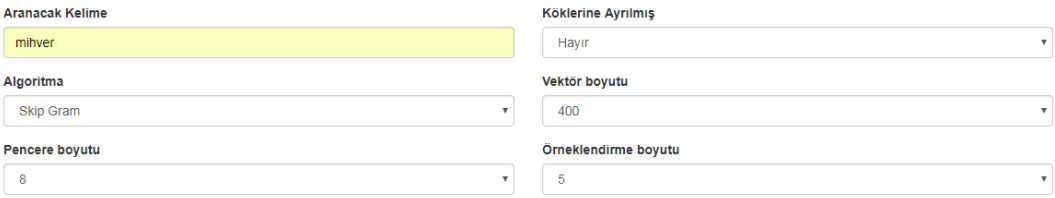
"http://35.163.160.151:8081/kelime/listele/unstemmed/skip/400/8/5/rubâ‘î"
],[
"harab",
"0.873447",

"http://35.163.160.151:8081/kelime/listele/unstemmed/skip/400/8/5/harab"
]

Aynı zamanda bu liste bir dizi şeklinde verildiği için bu liste cosine mesafesi yakınlığına göre büyükten küçüğe göre sıralanmıştır. "scores" başlığı altında bu mesafeler ayrıca listelenmiştir.

2.7. İnternet Arayüzü ile Sorgulama

RESTful API dışında bu çalışmada normal sorgulamalar için bir arayüz geliştirilmiştir. Bu arayüze ait bir ekran çıktısı Şekil 2'de verilmiştir. Bu arayüzde kullanıcılar sorgulanacak kelimeyi girdikten sonra diğer parametreleri seçebilmektedirler. "Listele" butonuna tıklandığında seçilen parametrelere göre sorgulama yapılarak aşağıda sonuçlar listelenmektedir. Daha sonra eğer kullanıcı aşağıda listelenen sözcüklerle tekrar arama yapmak isterse sonuçlar kısmında listelenen linklere tıklayabilmektedir. Her bir sözcük kendisine en yakın ilişkideki diğer 40 sözcüğe bağlanmak suretiyle bir ağ yapısı oluşturur. Bu arayüz sayesinde kullanıcılar, sözcükler arasındaki bu bilgi ağı üzerinde gezinti yapabilirler. Kullanıcılar bu siteye aşağıdaki link üzerinden erişilebilmekteler: http://kelime.site.

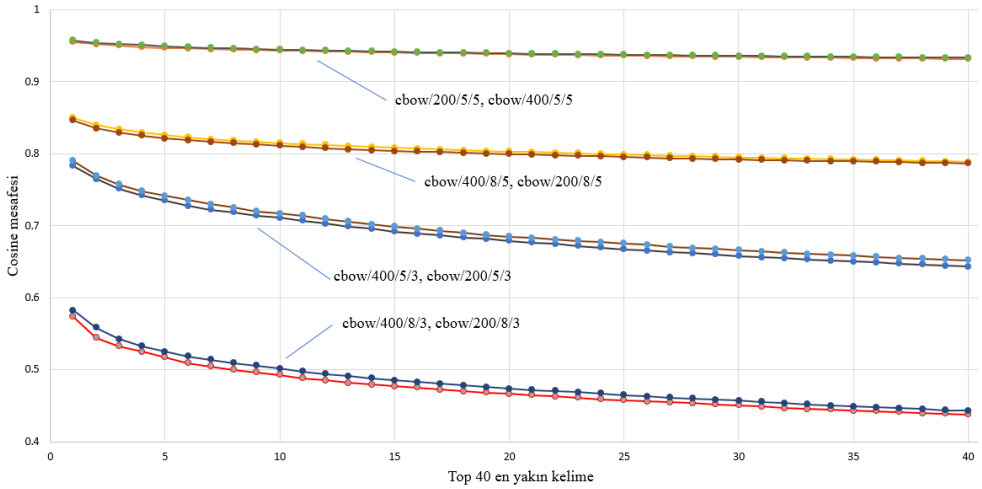


Şekil 2. İnternet tabanlı kullanıcı arayüzü

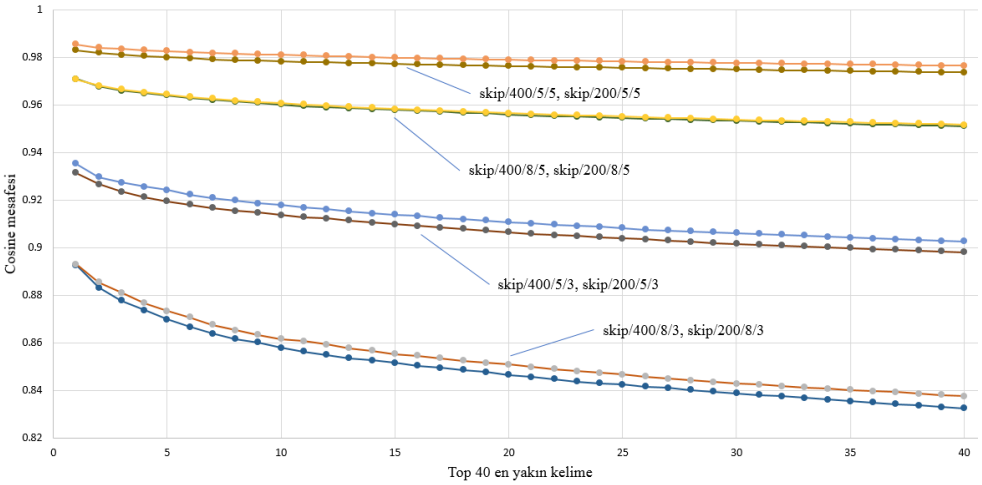
3. Bulgular

Kullanılan sinir ağı algoritması herhangi bir veri kümesi için birçok parametre ile çalıştırılabilir. Tablo 1'de bu parametreler listelenmiştir. Bu parametrelerin değişmesi durumunda ortaya farklı uzay vektörleri çıkmaktadır ve bu uzaylardaki vektörler arasındaki açılar değişiklik göstermektedir. Bu vektörel uzayların hangisinin işlenen veriyi en iyi şekilde yansıttığının tespit edilmesi detaylı bir değerlendirme gerektirir. Verilen sorgu kelimeleri sonucunda listelenen sözcüklerin sorgu kelimesine olan cosine mesafesi arttıkça kelime benzerliği de artmaktadır. Bu nedenle verilen parametreler arasındaki ilişkiler cosine mesafesi ölçeği ile karşılaştırılmaktadır. Bu işlem için öncelikle kelime havuzundan 400 farklı kelime seçildikten sonra bu kelimeler bir REST servisine gönderilerek geri gelen 40 kelimenin sorgu kelimesine olan cosine mesafesi JSON formatında kaydedilmektedir. Daha sonra Python programlama dili ile yazılan bir komut dosyası ile bu kelimelerin cosine mesafelerinin ortalaması alınmıştır. Şekil 3, 4, 5, 6'da bu deney sonuçları gösterilmektedir. Bu figürlerde x ekseni verilen kelimelere en yakın 40 kelimeye denk gelirken y ekseni bu kelimenin sorgulanan kelimeye olan cosine mesafesini göstermektedir. Şekil 3'te verilen sonuçlar kelime köklerine ayrılmış vektörlerden ve de CBOW algoritması kullanılan uzaydan elde edilmiştir. Şekil 4'de yine aynı şekilde kök ayrıştırma yapılmış fakat Skip-gram algoritması kullanılmıştır. Şekil 5 ve Şekil 6'da sırasıyla CBOW ve Skip-gram kullanılmıştır fakat farklı olarak veri kümesinde köklerine ayırma işlemi uygulanmamıştır.

Bu deneyde elde edilen bulgular aşağıda özetlenmiştir. Öncelikli olarak uzay boyutundaki değişikliğin elde edilen sonuçlarda çok ciddi bir farka yol açmadığı gözlenmiştir. 200 ve 400 şeklinde değiştirilen uzay boyutu sonuçları diğer parametreler sabit tutulduğunda ciddi bir farka yol açmamıştır. Şekillerde görülen üst üste gelmiş olan çizgiler bunu göstermektedir. Daha fazla boyut, daha fazla işlem masrafı ve depolama masrafı anlamına geldiği için bunun gibi kısıtlamaların olduğu sistemlerde 200 boyutlu uzayların oluşturulması yeterli olacaktır.



Şekil 3. Kelimeler köklerine ayrılmış ve CBOW algoritması kullanılmakta



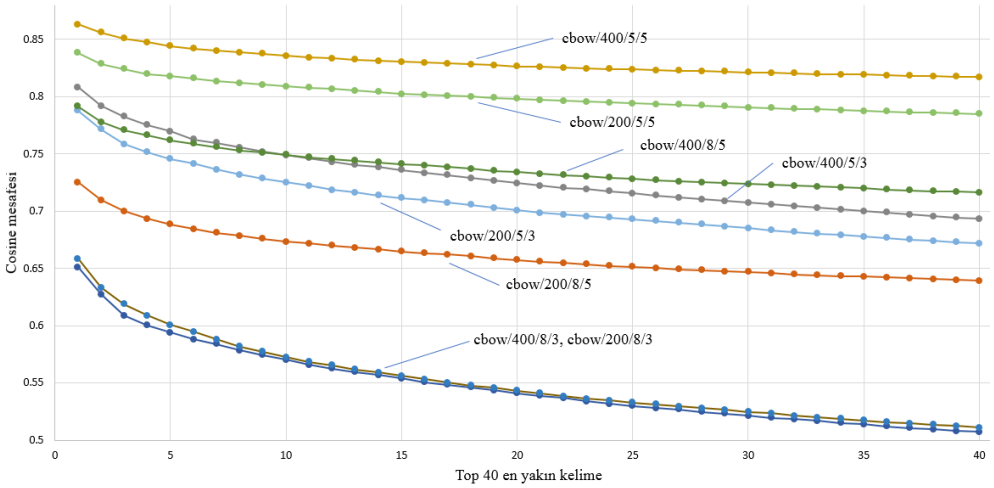
Şekil 4. Kelimeler köklerine ayrılmış ve Skip-gram algoritması kullanılmakta

Diğer bir gözlem ise genel olarak Skip-gram algoritmasında gözlenen mesafe yakınlıklarının CBOW algoritmasına göre çok daha yüksek oluşudur (Şekil 3, Şekil 4 ile karşılaştırılmalıdır, Şekil 5 ise Şekil 6 ile karşılaştırılmalıdır). Skip-gram algoritması genel olarak çok sık olmayan sözcüklerde iyi sonuçlar verir. CBOW ise performans olarak Skip-gram'a göre üstündür. Bu nedenle elde edilen sonuçlara göre performans problemi yaşanmayan veri boyutlarında Skip-gram algoritmasının kullanımı avantaj sağlayacaktır.

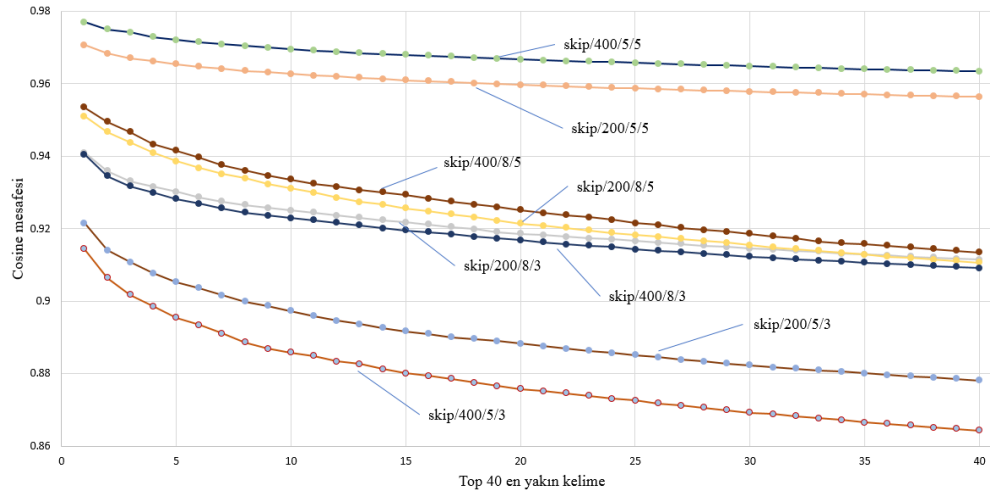
Üçüncü bir parametre olan pencere boyutu da elde edilen sonuçlar üzerinde oldukça etkilidir. Deney sonuçlarında açıkça görüldüğü gibi 5 kelimeli pencere boyutu 8 kelimeli pencere sonuçlarına kıyasla çok daha yüksek cosine mesafeli sonuçlar ortaya koymaktadır. 8 kelimeden oluşan pencerelerde kelimeler arasındaki anlamsal ilişkiler azalmaya başladığı için böyle bir sonucun gözlenmesi beklentiye uygundur.

Bir diğer parametre ise örnekleme boyutudur ki bu parametre 1e-3 ve 1e-5 olarak değiştirilmiştir. Sonuçlarda da gözlendiği üzere daha fazla örnekleme yapılan 1e-5 deneylerinde daha yüksek cosine mesafeli sonuçlar elde edilmiştir. Daha fazla örnekleme yapılan verilerde sıkça geçen terimler daha fazla temizlendiği için elde edilen uzayın daha az rastlanan sözcüklerden oluştuğu ve bu nedenle kelime vektörleri arasındaki mesafenin arttığı düşünülmektedir.

Aynı şekilde köklerine ayrılmış ve ayrılmamış kelime havuzlarından oluşturulmuş vektör uzayları karşılaştırıldığı zaman köklerine ayrılmış olan uzayda cosine mesafelerinin daha fazla olduğu görülmektedir. Bunun sebebi olarak yukarıda örnekleme boyutunda olduğu gibi köklerine ayırma işlemi yapıldığı zaman uzaydaki vektör sayısının azaldığı ve daha temiz bir veri ile uzayların oluşturulduğu düşünülmektedir.



Şekil 5. Kelimeler köklerine ayrılmamış ve CBOW algoritması kullanılmakta



Şekil 6. Kelimeler köklerine ayrılmamış ve Skip-gram algoritması kullanılmakta

4. Sonuç ve Tartışma

Yapılan bu çalışmada eski dil metinlerde geçmekte olan sözcükler arasındaki anlamsal ilişkilerin sinir ağları yöntemiyle tespit edilmesini sağlayan bir sistem geliştirilmiştir. Bu amaçla Kültür bakanlığı sitesinden elde edilen 32000 sayfa metin temizleme işleminden geçirildikten sonra üzerinde bir yapay sinir ağı algoritması koşturulmuştur. Ortaya çıkan uzay vektörleri bir sunucuya aktarılarak bir RESTful API servisi oluşturulmuştur. Kullanıcılar için http://kelime.site isimli internet sitesi üzerinden bir sorgu arayüzü oluşturularak kelimeleri sorgulama imkânı sağlanmıştır. Geliştirilen bu servislerin ve internet arayüzünün iki farklı amaçla kullanılması hedeflenmektedir. Birinci hedef bu sistemin Türk Dil Kurumu ve Kültür Bakanlığı gibi kurumların ve diğer Osmanlıca sözlük hizmeti sağlayan şirketlerin internet sitelerine entegre edilmesi ve aratılan sözcüklere yakın terimlerin kullanıcılara getirilmesidir. İkinci hedef ise tarih ve edebiyat gibi bilimsel çalışmalarda eski dilde yazılı olan metinlerin günümüz Türkçe'sine çevrilmesi esnasında ortaya çıkan hataların düşürülmesidir. Ayrıca kullanıma sunulan REST servisleri sayesinde bu alanda çalışmalar yapmayı planlayan araştırmacılara, kendi sonuçlarını geliştirilen sistemle karşılaştırabilme imkânı sunulmuş ve bu şekilde bu sahada yapılacak olan farklı akademik çalışmaların teşvik edilmesi hedeflenmiştir.

Yapılan analizlerde 200 ve 400 farklı boyutlu uzaylar arasında ciddi bir fark olmadığı tespit edilmiştir. Skip-gram algoritmasıyla elde edilen uzaylarda yapılan sorgularda daha yüksek cosine yakındalığı tespit edilmiştir. 5 kelimeden oluşan pencere boyutunun 8 kelimeye kıyasla çok daha iyi sonuçlar ortaya koyduğu ve de sıkça geçen sözcüklerin sayısı düştüğünde daha iyi sonuçlar ortaya çıktığı gözlemlenmiştir. Köklerine ayırma

işleminin örnekleme boyutunda olduğu gibi uzaylardaki vektör sayısını düşürdüğü için sorgulanan sözcükler için cosine mesafesini düşürücü etkisi olduğu gözlemlenmiştir. Eski dilde kullanılan sözcükler için kelime kök ayrıştırıcı geliştirilmesi durumunda ve daha fazla terimden oluşan veri kümeleriyle işlem yapıldığı takdirde daha olumlu sonuçlar elde edileceği sonucuna varılmıştır.

Geliştirilen bu sistemde şu ana kadar 54 kitap (32000) sayfa metin taranmıştır. Bundan sonraki çalışmalarda eski dilde yazılı olan daha fazla metin kaynağının taranması ve bu sayede sonuçların kalitesinin artırılması hedeflenmektedir. Bir diğer problem ise şu anda kullanılan sunucuların sanal makine olması nedeniyle belirli zamanlarda sorgulama sisteminde yavaşlama tespit edilmesidir. İlerleyen zamanlarda bu servislerin Amazon bulut servisleri üzerinden kiralanacak olan sunuculara aktarılarak daha güvenilir ve hızlı çalışan bir platform kurulması amaçlanmaktadır. Python ile yazılmış olan servislerin bir docker platformuna taşınarak yük dağılımı (load balancing) konularında daha yüksek performanslı, yanıt süresinin çok daha kısa olduğu bir sistemin elde edilmesi planlanmaktadır.

Kaynaklar

Adıgüzel, H., Şahin, P., Kalpaklı, M., 2012. Line segmentation of Ottoman documents. 20th Signal Processing and Communications Applications Conference, Fethiye Mugla, Turkey.

Arifoğlu, D., Duygulu, P., 2011. Word retrieval in ottoman documents. IEEE 19th Signal Processing and Communications Applications Conference, Antalya, Turkey.

Ataer, E., Duygulu P., 2007. Matching ottoman words: an image retrieval approach to historical document indexing. Proceedings of the 6th ACM international conference on Image and video retrieval, Amsterdam, Netherlands.

Basu, M., Roy, A., Ghosh, K., Bandyopadhyay, S., Ghosh, S., 2017. A Novel Word Embedding Based Stemming Approach for Microblog Retrieval During Disasters. 39th European Conference on Information Retrieval, Scotland, UK.

Can, B., 2017. Unsupervised learning of allomorphs in Turkish. Turkish Journal of Electrical Engineering & Computer Sciences 25(4), 3253-3260.

Chris, B., Faralli, S., Panchenko, A., Ponzetto, S., 2018. A framework for enriching lexical semantic resources with distributional semantics. Natural Language Engineering, Cambridge University Press, 24(1), 265-312.

Church, K. W., 2017. Word2Vec. Natural Language Engineering: Cambridge University Press, 155 p.

Deniz, K., Özçift, A., Bozyigit, F., Yıldırım, P., Yücalar F., Borandag E., 2017. TTC-3600: A new benchmark dataset for Turkish text categorization. Journal of Information Science, 43(2), 174-185.

Ganggao, Z., Iglesias, C. A., 2017. Computing Semantic Similarity of Concepts in Knowledge Graphs. IEEE Transactions on Knowledge and Data Engineering, 29(1), 72-85.

İlgen, B., Adalı, E., Tantuğ, A., 2016. Exploring feature sets for Turkish word sense disambiguation. Turkish Journal of Electrical Engineering & Computer Sciences, 24(1), 4391-4405.

Kalender, M., Korkmaz, E. E., 2018. THINKER - Entity Linking System for Turkish Language. IEEE Transactions on Knowledge and Data Engineering, 30(2), 367-380.

Kaya, Y., Ertugrul, O., 2016. A novel feature extraction approach for text-based language identification: Binary patterns. Journal of the Faculty of Engineering and Architecture of Gazi University, 31(4)

Kılıç, N., Gorgel, P., Ucan N., Kala, A., 2008. Multifont Ottoman character recognition using support vector machine. In Communications, Control and Signal Processing. St Julians, Malta.

Lushan H., Finin, T., McNamee, P., Joshi, A., Yesha, Y., 2013. Improving Word Similarity by Augmenting PMI with Estimates of Word Polysemy. IEEE Transactions on Knowledge and Data Engineering, 25(6), 1307-1322.

Marrero, M, Urbano, J., 2018. A Semi-automatic and low-cost method to learn patterns for named entity recognition. Natural Language Engineering, 24(1), 39-75.

Metin, B., Amasyalı, M., 2017. Dependency parsing with stacked conditional random fields for Turkish. Journal of the Faculty of Engineering and Architecture of Gazi University 32(2).

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space: arXiv preprint, 1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013. Distributed Representations of

Words and Phrases and Their Compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada.

Ozturk, A., Gunes, S., Ozbay, Y., 2000. Multifont Ottoman character recognition. 7th IEEE International Conference on Electronics, Circuits and Systems, Montreal, Quebec.

Peipei, L., Wang, H., Zhu, K. Q., Wang, Z., Hu, X., Wu, X., 2015. A Large Probabilistic Semantic Network Based Approach to Compute Term Similarity. IEEE Transactions on Knowledge and Data Engineering, 27(1), 2604-2617.

Soon, W. M., Ng, H. T., Lim, D., 2001. A machine learning approach to coreference resolution of noun phrases. Computational linguistics, 27(4), 521-544.

URL-1,2016. http://ekitap.kulturturizm.gov.tr/TR,78353/divanlar-ve-mesneviler.html

URL-2, 2018. http://snowball.tartarus.org/

URL-3,2019. https://code.google.com/archive/p/word2vec/source/default/source

URL-4,2019. http://snowball.tartarus.org/algorithms/turkish/stemmer.html

URL-5,2019. https://code.google.com/archive/p/word2vec/