

**T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



**DERİN ÖĞRENME YÖNTEMLERİ İLE TÜRKÇE METİNLERDE
BENZERLİK TESPİTİ**

İrfan AYGÜN

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ahmet Bedri ÖZER

ARALIK-2019

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMLERİ İLE TÜRKÇE METİNLERDE
BENZERLİK TESPİTİ

YÜKSEK LİSANS TEZİ

İrfan AYGÜN

(162129107)

Tezin Enstitüye Verildiği Tarih : 11 Kasım 2019

Tezin Savunulduğu Tarih : 13 Aralık 2019

Tez Danışmanı : Prof. Dr. Ahmet Bedri ÖZER (F.Ü)

Diğer Jüri Üyeleri : Prof. Dr. Mehmet KAYA (F.Ü)

Doç. Dr. Muhammed Fatih TALU (İ.Ü)

ARALIK-2019

ÖNSÖZ

Yüksek lisans öğrenimim boyunca desteğini esirgemeyen, yardım ve katkıları ile bu çalışmanın gerçekleşmesinde büyük emeği olan danışman hocam Prof. Dr. Ahmet Bedri ÖZER'e teşekkür ederim.

Ayrıca yaşamımın her anında beni destekleyen kıymetli aileme teşekkürlerimi sunarım.

İrfan AYGÜN
ELAZIĞ - 2019



İÇİNDEKİLER

	Sayfa No
ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	V
SUMMARY	VI
ŞEKİLLER LİSTESİ	VII
TABLolar LİSTESİ	X
KISALTMALAR LİSTESİ	XI
1. GİRİŞ.....	1
1.1. Yapısal Benzerlik Temelli Algoritmalar	1
1.1.1. Karakter Bazlı Algoritmalar.....	2
1.1.2. N-gram Temelli Algoritmalar	6
1.1.3. Token Bazlı Algoritmalar.....	7
1.1.4. Hibrit Algoritmalar.....	11
1.2. Semantik Metin Benzerliği Tespitine Yönelik Algoritmalar	14
1.2.1. Derlem Bazlı Semantik Benzerlik Metrikleri.....	15
1.2.2. Taksonomi Bazlı Semantik Benzerlik Metrikleri.....	16
1.2.3. Hibrit Metotlar.....	17
1.3. Derin Öğrenme Yöntemleri ile Anlamsal Benzerlik Çıkarımı.....	18
1.3.1. Word2Vec	19
1.3.2. Glove	23
1.3.3. FastText.....	24
1.4. Literatür Çalışması	24
1.4.1. Yapısal ve Anlamsal Benzerlik Çalışmaları.....	25

1.4.2.	Kelime Temsili Tabanlı Benzerlik Çalışmaları	29
2.	MATERYAL VE METOT İLE BULGULAR.....	34
2.1.	Bilgi Güvenliği Alanında Türkçe Derlemin Oluşturulması	34
2.2.	Veri Önışleme Adımları	35
2.2.1.	Verinin Sayısal İfadeler ve Noktalamalardan Arındırılması	36
2.2.2.	Verinin Tamamen Küçük Harflere Dönüştürülmesi	36
2.2.3.	Verinin Gereksiz Kelimelerden Arındırılması	36
2.2.4.	Verinin Yabancı Kelimelerden Arındırılması	40
2.3.	Önerilen Yöntemin Açıklanması.....	41
2.4.	Kosinüs Benzerliğinin Veriye Uygulanması	46
2.5.	Word2Vec Modelinin Oluşturulması	47
2.6.	Kosinüs Benzerliğinin Word2Vec Modeli ile Desteklenmesi	52
2.7.	Önerilen Yöntem ile Deneysel Çalışmalar	53
2.7.1.	Bilgi Güvenliği Verileri ile Deney 1	53
2.7.2.	Deney 1 Sonucunda Elde Edilen Bulgular	60
2.7.3.	Bilgi Güvenliği Verileri ile Deney 2	63
2.7.4.	Deney 2 Sonucunda Elde Edilen Bulgular	73
2.7.5.	Genel Türkçe Derlem ile Konu Bağımsız Deney 3	75
2.7.6.	Deney 3 Sonucunda Elde Edilen Bulgular	81
3.	SONUÇLAR VE TARTIŞMA	85
4.	ÖNERİLER.....	87
	KAYNAKLAR.....	88
	ÖZGEÇMİŞ	95

ÖZET

Günümüz teknolojilerinde doğal dillerin makineler tarafından algılanabilmesi birçok problemin çözümü için kaçınılmaz bir ihtiyaçtır. Metinler üzerinde yapılan çalışmalar sayesinde intihal tespiti, metin-yazar eşleştirmesi, metin konusunun tespiti, otomatik özet çıkarımı gibi birçok alanda yazılım tabanlı çözümler kullanılmaktadır. Örneklendirilen çalışmalar ve benzer birçok alanda, metinlerin bilgisayarlar tarafından yorumlanması ve işlenmesi beklenmektedir. Metinlerin yorumlanabilmesi için ise kullanılan dile ait gerek yapısal gerek ise dilbilimsel özelliklerin bilgisayarlar tarafından kavranması beklenir. Bu işlem sürecinde türlü zorluklar ve çözülmesi gereken problemler bulunmaktadır. Bu sorunlardan önemli bir tanesi, metinlerin bilgisayarlar tarafından anlamsal olarak kavranabilmesidir. Bilgisayarın metinden anlam çıkarabilir seviyeye gelmesi, bahsi geçen sorunlara büyük ölçüde çözüm sağlayacaktır. Ayrıca kullanıcı etkileşimini oldukça üst seviyeye çıkararak, çözümünde insan algısı gerektiren problemlere yazılım tabanlı alternatifler sunmak mümkün hale gelecektir. Metinlerin benzerliğini ölçmek, bu çerçevede ele alınan bir problem olarak literatürde türlü çalışmalara konu olmuştur. Geçmiş yıllarda metinleri sadece yapısal olarak kıyaslamak mümkün görülürken son yıllarda çeşitli anlamsal benzerlik tespiti yapan yöntemler geliştirilmiştir. Metinlerden anlamsal çıkarım yapmakta kullanılan güncel bir yöntem de derin öğrenme tabanlı kelime temsili yöntemidir. Bu yöntem sayesinde sözcüklerin anlamsal yakınlıklarını saptamak mümkün hale gelmiştir.

Bu çalışmada, Türkçe metinlerin hem anlamsal hem de yapısal şekilde incelenmesi sonucunda ortak bir yaklaşım ile benzerliklerinin ölçülmesi hedeflenmiştir. Yapısal benzerliğin ölçülmesinde Kosinüs Benzerliğinden, anlamsal benzerliğin tespitinde ise Word2Vec modelinden faydalanılmıştır. Tez sonucunda, bu iki farklı yaklaşımın ortak kullanımını sağlayan bir yöntem önerilmektedir. Bu kapsamda hem konu odaklı (Bilgi Güvenliği) hem de genel kullanım dili ile oluşturulan metinler üzerinde deneysel testler yapılmıştır. Elde edilen sonuçlar, önerilen yöntemin başarısını kanıtlar nitelikte bulgular ve sonuçlar bölümünde paylaşılmıştır.

Anahtar Kelimeler: Metin Benzerliği, Word2Vec, Kosinüs Benzerliği, Semantik Benzerlik, Türkçe Metinler

SUMMARY

Detection of Similarities with Deep Learning Methods in Turkish Texts

In today's technologies, the perception of natural languages by machines is an inevitable need for solving many problems. Thanks to the studies carried out on texts, software based solutions are used in many areas such as plagiarism detection, text-author matching, detection of text subject, and text summarization. It is expected that the texts will be interpreted and processed by computers in the exemplified works and many similar fields. In order to interpret the texts, it is expected that the structural and linguistic features of the language used should be comprehended by computers. There are various difficulties and problems to be solved in this process. One of these problems is the ability of computers to comprehend the texts semantically. If the computer is able to make sense from the text, it will provide a great solution to the aforementioned problems. In addition, it will be possible to offer software-based alternatives to problems requiring human perception in the solution by increasing user interaction to a very high level. Measuring the similarity of texts has been the subject of various studies in the literature as a problem addressed within this framework. In the past years, it has been possible to compare the texts only structurally, but in the recent years methods have been developed to detect various semantic similarities. A current method used to make semantic inferences from texts is deep learning based word representation method. This method makes it possible to determine the semantic affinity of words.

In this study, as a result of examining Turkish texts both semantically and structurally, it is aimed to measure similarities with a common approach. The structural similarity was measured by Cosine Similarity and the semantic similarity was used by Word2Vec model. As a result of the thesis, a method which provides the common use of these two different approaches is proposed. In this context, experimental tests were conducted on both subject-oriented (Information Security) and general-language texts. The results obtained are shared in the findings and conclusions section which prove the success of the proposed method.

Key Words: Text Similarity, Word2Vec, Cosine Similarity, Semantic Similarity, Turkish Texts

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 1.1. Skip-Gram mimarisi yapısı	20
Şekil 1.2. Skip-Gram mimarisi modeli	21
Şekil 1.3. CBOW mimarisi yapısı	22
Şekil 1.4. CBOW mimarisi modeli	22
Şekil 2.1. Bilgi güvenliği derleminde en sık geçen kelimeler.....	39
Şekil 2.2. Bilgi güvenliği derleminden stopwords'lerin silinmesi ile en sık geçen kelimeler.....	40
Şekil 2.3. Önerilen yöntemin algoritma akış şeması	45
Şekil 2.4. Kosinüs Benzerlik Algoritması vektörel gösterimi.....	47
Şekil 2.5. Kelime vektörlerinin gösterimi	48
Şekil 2.6. RSA kelimesi için Word2Vec ile bulunan yakın kelimeler	49
Şekil 2.7. Key kelimesi için Word2Vec ile bulunan yakın kelimeler	50
Şekil 2.8. Rusya kelimesi için Word2Vec ile bulunan yakın kelimeler.....	50
Şekil 2.9. IoT kelimesi için Word2Vec ile bulunan yakın kelimeler	51
Şekil 2.10. Botnet kelimesi için Word2Vec ile bulunan yakın kelimeler	52
Şekil 2.11. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 1).....	54
Şekil 2.12. Deney 1 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım1)	55
Şekil 2.13. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 2).....	56
Şekil 2.14. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 3).....	57
Şekil 2.15. Deney 1 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım3)	57
Şekil 2.16. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 4).....	58
Şekil 2.17. Deney 1 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 4)	58

Şekil 2.18. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 5).....	59
Şekil 2.19. Deney 1 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 5)	60
Şekil 2.20. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 1).....	65
Şekil 2.21. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 1)	65
Şekil 2.22. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 2).....	66
Şekil 2.23. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 2)	67
Şekil 2.24. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 3).....	68
Şekil 2.25. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 4).....	68
Şekil 2.26. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 4)	69
Şekil 2.27. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 5).....	70
Şekil 2.28. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 5)	70
Şekil 2.29. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 6).....	71
Şekil 2.30. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 7).....	72
Şekil 2.31. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 7)	72
Şekil 2.32. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 1).....	76
Şekil 2.33. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 1)	77
Şekil 2.34. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 2).....	77
Şekil 2.35. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 2)	78
Şekil 2.36. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 3).....	78

Şekil 2.37. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 3)	79
Şekil 2.38. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 4)	79
Şekil 2.39. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 4)	80
Şekil 2.40. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 5)	80
Şekil 2.41. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 5)	81



TABLÖLAR LİSTESİ

	Sayfa No
Tablo 1.1. N-gram temelli cümlelerin bölünmesi örneği	6
Tablo 2.1. Türkçe stopwords listesi	38
Tablo 2.2. Deney 1 için seçilen metinler	54
Tablo 2.3. Deney 1 sonucunda metinlerin yeni hali ve değişen kelimeler	61
Tablo 2.4. Deney 1 sonucunda benzerlik oranlarının cümle başına değişimi	62
Tablo 2.5. Deney 1 sonucunda algoritma tarafından saptanan yakın kelimeler	63
Tablo 2.6. Deney 2 için seçilen metinler	64
Tablo 2.7. Deney 2 sonucunda metinlerin ve kelimelerin değişimi	73
Tablo 2.8. Deney 2 sonucunda benzerlik oranlarının cümle bazlı değişimi	74
Tablo 2.9. Deney 2 sonucunda algoritma tarafından saptanan yakın kelimeler	75
Tablo 2.10. Deney 3 için seçilen metinler	76
Tablo 2.11. Deney 3 sonucunda metinlerin ve kelimelerin değişimi	82
Tablo 2.12. Deney 3 sonucunda benzerlik oranlarının cümle bazlı değişimi	83
Tablo 2.13. Deney 3 sonucunda algoritma tarafından saptanan yakın kelimeler	83

KISALTMALAR LİSTESİ

DDİ	: Doğal Dil İşleme
NLP	: Natural Language Processing
LCS	: Longest Common Subsequence
LSA	: Latent Semantic Analysis
SVD	: Singular Value Decomposition
LSO	: Lowest Super-Ordinate
YSA	: Yapay Sinir Ağları
CBOW	: Continuous Bag of Words
BO	: Benzerlik Oranı

1. GİRİŞ

Metin benzerliğinin hesaplanması diğer tüm benzerlik algoritmalarında olduğu gibi bilişim dünyasında oldukça sık kullanılmaktadır. Aralarındaki benzerliğin kıyaslandığı parametreler kelime, paragraf gibi küçük metin birimleri veya makale, kitap gibi büyük dokümanlar olabilmektedir. Standart araştırma ve incelemeler yapan bir internet kullanıcılarından, bireysel bilgisayar veya elektronik cihaz kullanan tekil kullanıcılara kadar hemen hemen her kullanıcı farkında olarak veya olmadan bu algoritmalarından faydalanmaktadır. Örnekleyecek olursak; işletim sistemlerinin dosya / klasör dizinlerinin oluşturulmasında ve bu dizinlere yönelik arama, açma, geri getirme gibi fonksiyonların kullanımında klasör adı, arama parametresi vb. diğer değişkenler metin benzerliği algoritmaları ile denetlenmektedir. Yine günlük hayatta sıkça kullanılan arama motorlarının başarılı sonuçlar verebilmesi için başvurdukları kaçınılmaz faktörlerinden bir tanesi de metin benzerliği algoritmaları veya anlamsal ilişki benzerliklerini çıkaran yöntemlerdir. Örnekler bilişim bilimlerine özgü incelendiğinde metin benzerliğinin en sık kullanıldığı alan olarak Doğal Dil İşleme (DDİ) çalışmaları görülmektedir. Adalı'ya [1] göre DDİ; metinlerin, semantik ve sentaks kurallarına dayanacak şekilde bilgisayarlar tarafından anlamlandırılması için yapılan çalışmalardır. Bu alan özelinde; intihal tespit sistemleri, metin – yazar eşleşmesi, konu bazlı belge sınıflandırma, içerik tarama sistemleri, içerik odaklı duygu analizleri gibi birçok çalışmanın çekirdeğinde metin benzerliği veya anlamsal benzerlik tespiti yöntemleri kullanılmaktadır. Bu çalışmalarda bazen problemin doğrudan çözümü için bazen de çözüme götüren ara işlemlerde metin bazlı benzerlik algoritmaları kullanılmaktadır. Kullanım alanının bu kadar geniş olması nedeniyle arka planında matematik ve yazılım odaklı çözümlere dayalı birçok benzerlik tespiti algoritması geliştirilmiştir. Bu algoritmalar bir sonraki bölümde yapısal benzerliğe dayalı sistemler, anlamsal metin benzerliği tespitine dayalı çalışan sistemler ve derin öğrenme tabanlı sistemler olarak detaylıca ele alınacaktır.

1.1. Yapısal Benzerlik Temelli Algoritmalar

Gali [2] ve arkadaşları yaptıkları çalışmada metin benzerliği algoritmalarını dört ana ölçüme dayanan alt başlıklar ile değerlendirmiştir. Bu başlıklar; karakter bazlı çalışmalar,

n-gram temeline dayanan çalışmalar, token bazlı çalışmalar ve hibrit çalışmalar olarak belirlenmiştir.

1.1.1. Karakter Bazlı Algoritmalar

a) Hamming Algoritması

Hamming uzaklığı, birbirine eşit iki dizgi arasındaki farkı gözlemleyebilmek için kullanılan en eski karakter bazlı eşleme algoritmalarından birisidir. Adını aldığı Richard Hamming tarafından 1950 yılında yayımlanan çalışma ile bitler arasındaki değişikliği gösterecek şekilde geliştirilmiştir. Bu metrik yapısal açıdan benzerlik veya farklılık bulma problemlerine temel seviyede çözümler bulmak için kullanılmıştır. Bitlerden oluşan dizgiler üzerinde etkili sonuçlar verdiği için çalışma yapısına uygun olarak telekomünikasyon ve bit düzeyinde denetim gerektiren ağ – kriptografi çalışmalarında sıklıkla kullanılmaktadır. Algoritmanın temel amacı; bir karakter dizesinin diğerinden ne kadar farklı olduğunu ve kaç adım ile bu dizelerin birbirine maksimum oranda benzeyeceğini ölçmektir [3]. Farklı bir bakış açısıyla, karşılaştırılan iki metnin XOR işlemine tabi tutulmasıyla elde edilecek sonuç Hamming uzaklığını verecektir. Bu algoritmanın formülü basit olarak aşağıdaki gibi gösterilebilir.

$$D_{hamming} = \sum_{i=1}^k |x_i - y_i| \quad (1.1)$$

Bu algoritmanın metin benzerliği tespitinde kullanımında bazı yetersizlikler ortaya çıkmaktadır. Bunlardan öncelikli olanı eşit uzunlukta olmayan metinlerin algoritmaya sokulabilmesi için ön işlem gerektirmesidir. Bir diğer zayıflığı birbirine yazım olarak çok yakın fakat anlamca çok uzak olan kelimelerde vereceği yüksek benzerlik sonucudur. Aşağıdaki örnekte de görüleceği üzere kolay ve kalay kelimeleri Türkçe için benzer kelimeler olmasa dahi Hamming uzaklığının 1 olmasıyla birlikte bilgisayar tarafından oldukça benzer oldukları şeklinde yorumlanmasına neden olacaktır.

$$D_{hamming}(kolay, kalay) = 1 \quad (1.2)$$

b) Levensthein Edit Distance

Vladimir Levensthein tarafından 1965 yılında geliştirilen bu algoritma ‘edit distance’ kavramına dayalı olarak dinamik programlama yöntemlerini kullanmaktadır. Edit Distance kavramı iki dizge arasındaki uzaklığı birim hamleler seviyesinde ölçmeye yaramaktadır. Bahsedilen birim hamleler ise ekleme, silme ve değiştirme işlemleridir. İki metin arasındaki mesafeyi ölçerken bir metni diğerine dönüştürmek için yapılacak ekleme, silme ve bire bir değiştirme işlemlerinin toplamı edit distance değerini vermektedir. Levensthein mesafesinin artması metinler arası benzerliğin azaldığını göstermektedir. Algoritma iki dizgenin ilk elemanlarından itibaren bire bir eşleşme mantığı ile eşit indeksli karakterleri inceler. Metinler arasındaki farklara göre karakter silme, ekleme veya mevcut karakteri değiştirme işlemleri sonrasında iki metin eşitlenmiş olur. Bu algoritmada Dinamik Programlama yöntemlerinin kullanılmasının nedeni, karmaşıklığı yüksek ve uzun işlemler barındıran çalışmaları daha küçük parçalara indirgeyebilmektir [4]. Bu yöntemle daha düşük karmaşıklık değeri ve parçalı çalışma sistemi ile uzun metinler arasında da işlemler yapabilmek kolaylaşacaktır.

İki farklı metnin (m_1 ve m_2) edit distance değerini gösteren algoritmanın çalışma prensibi ve benzerlik tespitinde kullanılan Levensthein algoritmasının formülü aşağıdaki gibidir.

$$lev_{m_1, m_2}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0 \\ \min \begin{cases} lev_{m_1, m_2}(i-1, j) + 1 \\ lev_{m_1, m_2}(i, j-1) + 1 \\ lev_{m_1, m_2}(i-1, j-1) + 1_{m_1[i] \neq m_2[j]} \end{cases} & \text{otherwise} \end{cases} \quad (1.3)$$

$$benzerlik(metin1, metin2) = 1 - \frac{lev(metin1, metin2)}{\max(|metin1|, |metin2|)} \quad (1.4)$$

Levensthein algoritması; benzerlik ve mesafe ölçümünden faydalanan yazım denetimi, intihal tespit sistemleri, konuşma tanımlama, kriptografi, DNA sarmallarının analizi gibi birçok farklı alanda sıklıkla kullanılmaktadır. Benzer şekilde arama motorlarında kullanılan, yakın içerikleri arama metinleri önerisinde veya ofis uygulamalarında kullanılan otomatik kelime düzeltme fonksiyonlarının temelinde bu

algoritmayı görmek mümkündür. Günümüz GPU teknolojilerinin en ileri seviyesi olan CUDA mimarisi de Levenstein Algoritmasını kullanmaktadır [5].

c) Longest Common Subsequence (LCS) Algoritması

Türkçeye en uzun ortak küme olarak çevrilen bu algoritma, iki dizge arasında bulunan sıralı en uzun alt kümeyi bulmayı hedefler. Anlatılan diğer algoritmalarından farkı; dizgeleri eşitlemekten öte içindeki en uzun eşleşmeye odaklanmasıdır. Bu yaklaşımıyla algoritmanın kopya ve intihal tespitinde etkili sonuçlar vermesi muhtemeldir. Fakat algoritmanın çalışma yöntemiyle birlikte yüksek karmaşıklık ve hesaplama maliyeti sunması en büyük dezavantajları olarak görülmektedir. Bu noktada Levenstein tarafından da aynı çözüme yönelik kullanılan dinamik programlama yaklaşımları ile karmaşıklığı minimize etmek mümkündür. Longest Common Subsequence (LCS) Algoritmasını bir örnek ile aşağıdaki gibi incelemek mümkündür. Örnekte m1 ve m2 şeklinde iki metin algoritmaya dizgeler halinde sunulmaktadır. Bu dizgelerin üyeleri ise harfler olacak şekilde ayrılmıştır.

$$m1 = \{K, \dot{I}, T, A, P, L, A, R\}$$

$$m2 = \{\dot{I}, T, A, L, Y, A\}$$

$$LCS_{(m1, m2)} = \{\dot{I}, T, A, L, A\}$$

Örnekte de görüleceği üzere en uzun ortak küme bulunurken sadece yan yana bulunan öğeler değil de aynı doğrultuda sıralı bulunan öğeler göz önünde bulundurulmuştur. Algoritmanın çalışmasıyla iki metinde de sıralı ve yan yana bulunan $\{\dot{I}, T, A\}$ alt kümesi sonuç olarak döndürülecek olsaydı algoritmayı manipüle etmek oldukça kolaylaşırdı. Metin benzerliği çalışmalarında algoritmanın bu özelliği sayesinde, araya girecek yeni kelime veya bağlaç ile cümlelerin benzerlik oranları kritik seviyede düşmemektedir. Bu algoritmanın benzerlik tespiti için kullanımına yönelik formül aşağıda gösterilmektedir.

$$LCS_{(m1, m2)} = \frac{LCS_{(m1, m2)}}{\max(|m1|, |m2|)} \quad (1.5)$$

d) Jaro-Winkler Algoritması

Jaro-Winkler Algoritması, Jaro'nun 1989 yılında yaptığı çalışmayı [6] bir adım daha ileri taşıyarak doğrudan metin benzerliğini bulma amacıyla geliştirilen bir algoritmadır. Jaro tarafından yapılan çalışma dizgelerdeki ortak karakterlerin sayısını ve sırasını kullanmaktadır. Öncelikle iki dizgenin uzunluğu hesaplanır ve ardından ortak karakterler bulunur. İki dizgede aynı olan karakterlerin dizge içerisindeki indisleri aynı olmasa dahi (araya karakter girmiş olabilir) yakınlıklarını hesaplayarak genel benzerliğe dahil eder veya gözden çıkarılmasına karar verir. Bu kararı vermek için aşağıdaki formülden yararlanılır.

$$[\max(|m_1|, |m_2|) / 2] - 1 \quad (1.6)$$

Formüle göre çıkan sonuç m_1 ve m_2 metnilerindeki yakınlık sınırıdır. Eşleşen karakterlerin indisleri farkı, formülden çıkan sonuçtan daha küçükse eşleşir ve benzerlik hesabına dahil edilir. Son olarak transpozisyon değeri toplam eşleşen karakter sayısının yarısı olarak bulunur.

Winkler ise yaptığı çalışmada [7] Jaro metriğini modifiye ederek dört karaktere kadar ortak bir ön kısma sahip dizgelere daha fazla ağırlık verilmesini sağlamıştır. Bu işlemi gerçekleştirebilmek için $p = 0.1$ olacak şekilde p sabitini formüle dahil etmiştir. Daha iyi açıklanabilmesi açısından formülün yeni hali aşağıda gösterilmektedir.

$$d_{jaro} = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3} \left(\frac{m}{|m_1|} + \frac{m}{|m_2|} + \frac{m-t}{m} \right) & \text{otherwise} \end{cases} \quad (1.7)$$

$$d_{winkler} = d_{jaro} + (lp + (1 - d_{jaro})) \quad (1.8)$$

Yukarıda gösterilen formüllerde m değişkeni eşleşen karakterleri, t değişkeni transpozu, p değişkeni 0.1 olarak kullanılan sabiti ve l değişkeni de ortak olan ön kısmın uzunluğunu temsil etmektedir. Winkler l uzunluğu için maksimum üç karakteri algoritma başarısı açısından optimum değer olarak saptamıştır [7].

1.1.2. N-gram Temelli Algoritmalar

Algoritma bazı kaynaklarda q-gram olarak da geçmektedir. Adını aldığı ifadelerden n; tekrarların kontrol edildiği değer ve gram ifadesi de bu tekrarın dizge içerisindeki ağırlığını hesaplamak için kullanılır [8]. Benzerliği ölçerken iki metinde de bulunan n adet bitişik karakter dizilerini tespit etmek üzere çalışan bir algoritmadır. Algoritmanın temel parametresi olan n değeri farklı versiyonlara göre değişebilmektedir. Buna bağlı olarak unigram, bigram, trigram şeklinde ifade edilen türetilmiş versiyonlar literatürde yer alır. Ayrıca n-gram yaklaşımı ile Token Bazlı Algoritmaların hibrit şekilde kullanılması sonucu Bi-Jaccard, Bi-Dice şeklinde yeniden adlandırılmış algoritmalar ortaya çıkmıştır. Algoritmanın mantığı basit şekilde aşağıdaki örnekte gösterilmektedir. Belirlenen n değeri 2 olacak şekilde sonuçlar Tablo 1.1’de gösterilmiştir.

metin = “ahmet bedri özerin kriptografi çalışmaları metni”

Tablo 1.1. N-gram temelli cümlelerin bölünmesi örneği

2-gram	Ağırlık	2-gram	Ağırlık	2-gram	Ağırlık	2-gram	Ağırlık
ri	3	iö	1	og	1	ma	1
me	2	öz	1	gr	1	la	1
et	2	ze	1	ra	1	ar	1
al	2	er	1	af	1	rı	1
ah	1	in	1	fi	1	ım	1
hm	1	nk	1	iç	1	tn	1
tb	1	kr	1	ça	1	ni	1
be	1	ip	1	lı	1		
ed	1	pt	1	ış	1		
dr	1	to	1	şm	1		

Belirtilen örnekten anlaşılacağı üzere en çok birlikte geçen ikili dizge ‘ri’ dizgesidir. Örnekte n değeri 3 seçilecek olsaydı ‘met’ dizgesinin metinde iki kez geçtiği geri kalan tüm üçlü ifadelerin ise sadece bir kez geçtiği görülecekti. Çalışılan dile göre n değerinin artması örnek uzayı büyütecektir. Türkçe için 29 harf bulunduğunu ve sadece harflerden oluşan bir metnin incelendiğini düşünürsek ihtimal uzayı 29^n olacak şekilde büyüyecektir.

N-gram ile benzerlik tespitinin yapılabilmesi için birden fazla metin gerekmekte ve her bir metne örnekte yapılan işlemler uygulanmaktadır. N değeri 3 olarak alındığında Trigram olarak isimlendirilmekte ve örnekte görülen işlemlerin üçer harf ile aynıysa gerçekleşmektedir. Trigram ve daha fazla arttırılacak n değerleri için karşılaşılabilecek temel problem kelimedeki harf sayısının yeterli olmaması haliyle kelimenin bölünemeyecek olmasıdır. Sadece N-gram kullanılan bir çalışma için bu problem önem arz etmese dahi daha sonra değinilecek hibrit çalışmalarda N-gram ile birlikte kullanılacak diğer algoritmaların çalışma prensiplerine uymaması söz konusu olacaktır.

1.1.3. Token Bazlı Algoritmalar

a) Jaccard İndeksi

Jaccard Index olarak adlandırılan indeksin özelliği iki metnin ortak özelliklerini toplam özelliklerinin sayısına oranlayarak metinler arasındaki benzerliğin tahminini kolaylaştırmasıdır [9]. Söz konusu indeksi kullanabilmek için metinler üzerinden özellik çıkarımı (feature extraction) yapılması gerekmektedir. Metinler için özellik kavramı bir harf veya harfler dizisi şeklinde olabilir. Bir örnek üzerinde inceleyecek olursak, aşağıdaki örnekte m1 ve m2 metinlerinde bi-gram kullanarak tek harf, birer kelime (farklı özellikler de seçilebilir) veya her bir metnin içindeki bitişik harf ikilileri birer özellik (feature) olarak kabul edilmiştir.

m1 = “kalem”

m2 = “emsal”

bi-gram(m1) = {ka,al,le,em}

bi-gram(m2) = {em,ms,sa,al}

(m1 ∩ m2) = {em,al}

(m1 ∪ m2) = {ka,al,le,em,ms,sa}

Yukarıdaki örnek veya benzer şekilde kurgulanmış metinler (farklı kullanım alanları da bulunmaktadır) arasındaki benzerlik oranını bulabilmek için kullanılan Jaccard formülü aşağıdaki gibidir.

$$J(m1, m2) = \frac{|m1 \cap m2|}{|m1 \cup m2|} \quad (1.9)$$

Formüle göre m1 ve m2 metinlerinin benzerlik oranı 0,333 olacak şekilde sonuç alınmıştır. Sonuçtan da yola çıkarak yapılacak değerlendirmede Jaccard Indexi 1 değerine yaklaştıkça benzerlik oranı artmaktadır. Literatürde ayrıca Jarccard Distance olarak da kullanılan tabir bulunmaktadır. Bu değer ise Jaccard Index değerinin 1’den çıkarılması ile elde edilir. Mesafenin büyümesi benzerliğin azaldığına işaret eder.

b) Dice Katsayısı

Dice benzerlik katsayısı metinler arası benzerliği bulabilmek için Jaccard Indexi benzeri bir yaklaşımla kullanılmaktadır. Bu katsayı hesaplanırken bir önceki bölümde olduğu gibi metinlerden özellik çıkarımı yapılmalıdır. Çıkarılan özellikler tek harf veya bi-gram, tri-gram şeklinde harf dizgeleri olabilmektedir. Metnin büyüklüğüne göre özellik seçiminin etkisi önemli seviyelere ulaşabilmektedir [10].

Jaccard Index için yapılan örneğin aynısını Dice katsayısı için de yapacak olursak iki benzerlik algoritması arasında kıyas imkanı doğacaktır.

m1 = “kalem”

m2 = “emsal”

bi-gram(m1) = {ka,al,le,em}

bi-gram(m2) = {em,ms,sa,al}

(m1 ∩ m2) = {em,al}

|m1| = 4

|m2| = 4

Örnekteki verilerin değerlendirilebilmesi için Dice sabitini bulmakta kullanılan formül aşağıdaki gibidir. Formüle göre; metinler arasındaki kesişimin iki katı ile metinlerin toplam özellik uzunluklarının oranı bir ‘s’ sabiti vermektedir. Bu sabit 1 değerine yaklaştıkça benzerlik oranı da yükselmektedir.

$$S_{dice} = \frac{2|m1 \cap m2|}{|m1| + |m2|} \quad (1.10)$$

Formüle göre sonuçlar değerlendirildiğinde m1 ve m2 metinleri arasındaki benzerlik oranı 0,5 olarak bulunmaktadır. Aynı metinler için Jaccard İndeksi kullanıldığında ise bu

oran 0,333 bulunmuştur. Farklı benzerlik algoritmalarının aynı metinler üzerinde değerlendirilmesiyle çıkan sonuçlar yanıltıcı olabilmektedir. Bu nedenle çıkan sonuçlar mutlaka bir değerlendirmeye tabi tutulmalı ve anlamsal ilişki ile çıkan sonuç arasında tutarlılık kontrolü yapılmalıdır. Bu işlemler neticesinde çalışılacak dil veya derleme özgü algoritmanın seçimi daha kolaylaşacaktır.

c) Kosinüs Benzerliği

Dokümanların terim vektörleri olarak gösterilmesi durumunda iki doküman arasındaki benzerlik, vektörler arasındaki kolerasyona karşılık gelmektedir. İki vektör arasındaki açının kosinüs değeri bu metrikte ana öge olarak kabul edilmektedir [11]. Kelimeleri vektörler şeklinde ifade edebilmek için birçok yöntem olmakla birlikte en sık olarak terim-frekans değerleri kullanılmaktadır. Bu yöntem bir sonraki başlıkta daha detaylı şekilde açıklanmıştır. İnceleme yapılacak dokümanlarda her bir kelime ayrı ayrı 'n' boyutlu vektörler olarak kullanılabilirler. Fakat metnin büyüklüğüne göre cümleler veya paragrafların da vektörler olarak ifade edilmesi mümkündür. Birer cümlelik iki metin için kosinüs benzerliği değeri aşağıdaki örnekte hesaplanmaktadır.

m1 = Metin benzerliği çalışmalarında farklı yöntemler kullanılabilir.

m2 = Derin öğrenme metin benzerliği ve diğer farklı problemlerde kullanılabilir.

Terimler = {metin, benzerliği, çalışmalarında, yöntemler, derin, öğrenme, farklı, ve, diğer, problemlerde, kullanılabilir}

m1 : [1,1,1,1,0,0,1,0,0,1]

m2 : [1,1,0,0,1,1,1,1,1,1]

Yukarıdaki örnekte iki cümle arasındaki benzerliği bulabilmek için terimler belirlenmiştir. Bu terimler iki cümledeki kelimelerin birleşim kümesini ifade eder. Ardından her bir cümle için vektörel gösterimi yapılırken terimler kümesinin elemanları sırasıyla cümlede geçip geçmemesine göre 1 ve 0 olarak işaretlenmiştir. Örneğin geri kalanında ise aşağıda gösterilen kosinüs benzerliği formülüne göre işlemlere devam edilmelidir.

$$S = \frac{m1.m2}{||m1||.||m2||} = \frac{\sum_{i=1}^n m1_i \times m2_i}{\sqrt{\sum_{i=1}^n (m1_i)^2 \times \sum_{i=1}^n (m2_i)^2}} \quad (1.11)$$

Formüle göre iki vektör arasındaki nokta çarpımı ile vektör boyutlarının oranı sonucu vermektedir. Vektörlerin nokta çarpımları yapıldıktan sonra Öklit Uzayından faydalanılarak vektör boyutları bulunmaktadır. Elde edilen matematiksel değerlerin oranlanması neticesinde 0 ile 1 arasındaki S değeri bulunacaktır. Bu değer 1'e yaklaştıkça metinlerin benzerlik oranı yükselecek, 0'a yaklaştığında ise iki metnin birbirine benzemediği sonucuna varılacaktır. Kosinüs benzerliğinin en önemli özelliklerinden birisi belge uzunluklarının farklı olması gibi durumlarda sorun yaşamamasıdır. Normalizasyon sayesinde bu gibi sorunlar çözülerek sonuca olumsuz etki etmesi engellenmektedir.

d) TF-IDF

Türkçesi 'terim frekansı – ters metin frekansı' olacak şekilde Term Frequency – Inverse Document Frequency kelimelerinin baş harflerinden oluşan bu yöntem, bir metin içerisindeki terimlerin çıkarılması ve ağırlıklandırılmasını ifade etmektedir. DDİ çalışmalarında sık kullanılan bir yöntem olmakla birlikte 'Bilgi Geri Getirimi' ve 'Belge Sınıflandırma' konularında sıklıkla TF-IDF'e başvurulmaktadır [12]. Metinlerle ilgili yapılan birçok analiz, istatistik ve bilişim çalışmalarında gerek tek başına gerekse diğer algoritmaların başarımını arttırmak için ilave olarak kullanılan bu yöntem aşağıdaki formüllere göre çalışmaktadır.

$$TF = \frac{\text{Terimin Dokümandaki Tekrar Sayısı}}{\text{Dokümandaki Toplam Kelime Sayısı}} \quad (1.12)$$

$$IDF = \log \frac{\text{Toplam Doküman Sayısı}}{\text{Terimi İçeren Doküman Sayısı}} \quad (1.13)$$

$$TF - IDF = TF \times IDF \quad (1.14)$$

Terim frekansının hesaplanmasında Binary (var veya yok), Raw Frequency, Log Normalization gibi farklı yöntemler kullanılmaktadır. Dokümanın uzunluğuna göre seçilecek yöntem önem kazanmaktadır. Örnek olarak Raw Frequency metodu dokümandaki tüm kelimeleri sayarak işleyeceği için uzun dokümanlarda etkin sonuç

almayı engelleyebilir. IDF hesabında ise birden fazla dokümandan oluşan bir külliyyat gözden geçirilmektedir. Bu süreçte terimin diğer dokümanlarda bulunma oranı yorumlanabilir sonuçlar doğuracaktır. Örneğin terim her dokümanda geçiyor veya çok fazla dokümanda bulunuyor ise bu terimin bağlaç veya sık kullanılan bir edat olması ihtimali yüksektir. Bu ifadeler DDİ çalışmalarında ‘stopwords’ olarak adlandırılır ve çalışmaların başarısını arttırabilmek için çoğunlukla belgeler bu ifadelerden arındırılır. TF-IDF hesaplaması benzer şekilde metinlerin ana konularını bulabilmek, yazarın sık kullandığı ifadeleri çıkararak metin-yazar eşleşmesi yapmak, yazara yönelik ilgi tespiti yapmak, metin özeti çıkarmak, arama motorları için skorlama yapmak vb. gibi birçok konuda kullanılmaktadır [13].

1.1.4. Hibrit Algoritmalar

Metin eşleştirmede kullanılan algoritmalar kendi içlerinde farklı koşullara göre avantajlı veya dezavantajlı bir kullanıma sahip olabilirler. Bu şekilde tek bir algoritmaya bağlı kalarak sonuç almaya çalışmak, somut projelerde ve gerçek hayat sorunlarında yeterli olamayabilir. Bir algoritmanın dezavantajını ortadan kaldırmak ve diğer yöntemlerin avantajlı noktalarını da sisteme entegre edebilmek için hibrit algoritmalar geliştirilmiştir [14]. Somutlaşmış bir örnekle, karakter bazlı çalışan bir algoritmanın temel problemlerinden birisi hatalı yazımları ayıramaması veya benzer yazıma sahip farklı anlamlı kelimeleri benzer bulmasıdır. Token bazlı çalışmalarda ise token büyüklüğüne göre küçük farklılıklar benzerlik oranına ciddi etkide bulunabilmektedir. İki farklı yönelimin de açığını kapatacak şekilde, token bazlı çalışmaların içerisine karakter bazlı yaklaşımlar dahil edildiğinde belirtilen sorunların etkisi minimum seviyeye inecektir.

Hibrit sistemler, karşılaşılan problemin çözümüne göre benzerlik algoritmaları ile sıralama algoritmalarının birleşimini, en kısa yol algoritmalarının birleşimini ve daha birçok farklı konuya yönelik çalışmayı da kapsayacak şekilde oluşturulabilirler. Tezin konusu gereği sadece metin benzerliğine yönelik en sık kullanılan Soft TF-IDF ve Monge-Elkan Algoritmaları bu başlık altında incelenmiştir.

a) Soft TF-IDF

Adından da anlaşılacağı üzere TF-IDF yaklaşımının yumuşatılmış versiyonu olarak bilinen algoritma, yöntem olarak Genelleştirilmiş Jaccard ölçüsüne benzemektedir.

Algoritma adında geçen yumuşatmak değimi ile kastedilen ise hata toleransının artırılmasıdır. DDİ çalışmalarının büyük çoğunluğunda geniş veri kümeleri, milyonlarca veriden oluşan derlem veya külliyatlar kullanılmaktadır. Çalışılan verinin hacmi arttıkça hatalı veri (noisy data) barındırma ihtimali de bir o kadar yükselmektedir. Bu problemin çözümü olarak veri normalizasyonu gibi özel tekniklerin kullanılmasına rağmen bir data setini tam olarak hatasız hale getirebilmek ciddi zaman maliyeti gerektirir. Proje ölçeğine göre bu datalardan kaynaklı hataları görmezden gelmek bir seçenek olarak görülse de hataları minimuma indirebilmek için algoritmaların optimizasyonuna yönelik çalışmalar (veri ön işleme) kaçınılmaz şekilde devam etmektedir. Soft TF-IDF algoritması da yanlış yazımdan kaynaklı hatalı verileri tespit etmekte sıklıkla kullanılan bir yöntem olmuştur. Ayrıca makine tarafından anlaşılması mümkün olmayan kısaltmalar veya farklı gösterimler normal TF-IDF algoritmasında farklı bir kelime olarak algılanırken Soft TF-IDF ile daha başarılı şekilde frekans ölçümü yapılabilmektedir.

Doan ve arkadaşları [14] Soft TF-IDF yöntemini üç örnek data üzerinden aşağıdaki gibi açıklamıştır.

M1 = Apple Corporation, CA

M2 = IBM Corporation, CA

M3 = Aple Corp.

Yukarıdaki üç metni gözlemlediğimizde m1 ve m3 birbirinden farklı datalar olarak algılanacaktır. Fakat m3'te yapılan yazım yanlışlığı ile Apple olarak girilmesi gereken data Aple olarak girilmiş ve Corporation yerine kısaltma kullanılarak Corp. ifadesi kullanılmıştır. Bu iki metin standart TF-IDF ile incelendiğinde herhangi bir ortak küme içerisine dahil edilmeyecektir. Fakat Genelleştirilmiş Jaccard Ölçüsündeki gibi bir yaklaşımla 'Aple' ve 'Apple' dataları aynı data olarak algılanırsa daha doğru sonuç ile daha yüksek başarı yakalamak mümkündür. Bu eşleşmenin gerçekleştirilmesine 'yumuşatma' ifadesi uygun görülerek yumuşatılmış TF-IDF algoritması olarak adlandırılmıştır. Bu algoritmanın çalışma adımları ise aşağıdaki gibi işlemektedir.

$$s(x, y') = \sum_{t \in \text{close}(x, y, k)} v_x(t) \cdot v_y(u_*). s'(t, u_*) \quad (1.15)$$

$$\begin{array}{lcl}
x = abcd & B_x = \{a, b, c, d\} & \\
& \begin{array}{c} \diagup \quad \diagdown \quad \diagup \quad \diagdown \\ 1 \quad 0.7 \quad 0.8 \quad 1 \quad 0.6 \end{array} & \\
y = aa'b'cd' & B_y = \{a, a', b', c, d'\} & \\
\text{close}(x, y, 0,75) = \{a,b,c\} & & \\
(a) & (b) &
\end{array}$$

$$s(x,y) = v_x(a).v_y(a).1 + v_x(b).v_y(b').0.8 + v_x(c).v_y(c).1 \quad (1.16)$$

(c)

Formüle göre x ve y şeklinde okunan iki veri, B_x ve B_y şeklinde tekrar oluşturulmaktadır. Sabit bir k değeri ise formülde yakınlık ölçütü olarak kullanılmıştır. Bu değerin 0,75 seçilmesi durumunda formüldeki gibi x ve y metinlerinin hata içerisindeki her bir karakterin 0,75 toleransla sapması göz önünde bulundurulmuş ve tüm benzer ifadeler ortak bir uzay üzerinde değerlendirilmiştir. Bahsi geçen yaklaşımla birlikte beş harften oluşan bir kelimenin eksik bir harf veya fazla bir harf ile yanlış yazımından dolayı oluşan hatalar telafi edilebilecektir.

b) Monge-Elkan Mesafesi

Monge-Elkan mesafesinin ölçümünde karşılaştırılacak iki metin alt tokenlere bölünmektedir. Bu tokenleştirme işleminden sonra her bir alt parçanın arasındaki benzerlik oranlarının ortalaması Monge-Elkan mesafesini vermektedir [15]. Algoritmada alt tokenler arasındaki benzerlik oranı hesaplanırken, Jaro-Winkler veya Levensthein gibi algoritmalar kullanılır. Ardından yapılan hesaplamayla birlikte çıkan ortalama ise bu iki Algoritmadan daha gerçekçi sonuçlar vermektedir. Bu nedenle hibrit yöntem metin benzerliğinde sık tercih edilen yöntemlerden birisi olmuştur. İki metin (m_1 ve m_2) arasındaki benzerliği hesaplayan algoritmanın formülü ise aşağıda gösterildiği gibidir.

$$Mango - Elkan(m_1, m_2) = \frac{1}{|m_1|} \sum_{i=1}^{|m_1|} \max\{sim'(m_{1_i}, m_{2_j})\}_{j=1}^{|m_2|} \quad (1.17)$$

1.2. Semantik Metin Benzerliği Tespitine Yönelik Algoritmalar

Metinler bire bir şekilde aynı kelimelerden oluşmasa dahi aynı anlamda veya çok yakın anlamlarda kullanılabilirler. Bu durumda doğrudan metin eşleşmesine yönelik algoritmaların bulacağı değerlerin semantik açıdan yanıltıcı olması muhtemeldir. Aşağıda verilen iki örnek cümle ile semantik benzerlik yöntemlerine duyulan ihtiyaç anlatılmıştır.

metin1 = “Veri analizi çalışmalarında data optimizasyonu katmanı birçok yöntem için elzemdir.”

metin2 = “Veri analizi yapılan araştırmalarda veri iyileştirme katmanı birçok algoritma için gereklidir.”

Örnekte verilen metin1 ve metin2 anlamsal olarak birbirine çok yakın cümlelerdir. Cümleler birbirine yakın veya eşanlamlı kelimelerden kurulu ve benzer bir sonuca varılacak anlatıma sahiptirler. Örnekteki gibi yoğun ilişkili anlatıma sahip metinler doğrudan eşleşme algoritmaları ile incelendiğinde alınan sonuçlar yanıltıcı olacaktır. Örnekteki metinlerin Kosinüs Benzerliği ile ölçümü yapıldığında elde edilen sonuç 0.32672642 olarak görülmekte yani yaklaşık olarak üçte bir oranda benzerlik bulunmaktadır. Fakat cümleler semantik yaklaşım ile tekrar incelendiğinde çok daha yüksek sonuçların alınması beklenir. Bu örnek ile aşağıdaki kelimelerin tekrar incelenmesi semantik benzerlik çalışmalarının konusuna dahil olmaktadır.

data	→	veri
optimizasyon	→	iyileştirme
algoritma	→	yöntem
çalışma	→	araştırma
elzem	→	gerekli

Eş anlamlı kelimeler, yakın anlamlı kelimeler veya kısaltmalar semantik yaklaşımı metin benzerliklerinde zaruri hale getirmektedir. Zira özellikle intihal çalışmalarında semantik yaklaşımın kullanılmaması halinde doğrudan eşleşme metotlarını manipüle etmek basit değişiklikler ile mümkün hale gelmektedir.

Semantik benzerliğe yönelik algoritmalar temel olarak üç kategoride değerlendirilmektedir. Bu ana kategoriler derlem bazlı algoritmalar, taksonomi bazlı algoritmalar ve hibrit algoritmalar.

1.2.1. Derlem Bazlı Semantik Benzerlik Metrikleri

DDİ çalışmalarında oldukça fazla belgenin bir arada derlenmesi ile oluşturulan belirli bir çalışmaya yönelik dataların toplamına derlem adı verilmektedir. Elde edilen text içerikli büyük data topluluğu birçok sorunun çözümü için kullanılmaktadır. Dilin anlamsal ve kavramsal altyapısına yönelik çalışmalarla metin benzerliği, önemli terim tespiti, yazar tespiti, konu özeti çıkarma gibi birçok konuda derlem bazlı çalışmalar yapılmaktadır. Anlamsal benzerlik hesaplama algoritmalarının bir kısmı da diğer örneklerde olduğu gibi derlem tabanlı çalışmayı tercih etmektedir. Bu tercih; yapılacak çalışmaya özgü alansal veri toplamak, kategorik verileri toplamak ve derlemi oluşturan dataların optimizasyonu gibi işlem gerekliliklerini beraberinde getirmektedir.

a) LSA Semantik Benzerlik Algoritması

Latent Semantic Analysis (LSA), anlam bakımından yakın olan kelimelerin benzer metin parçalarında ortaya çıkacağını varsayan bir yaklaşım sonucu ortaya çıkmıştır [16]. Launder tarafından 1990 yılında ortaya atılan bu fikir ilerleyen yıllarda çok daha popüler şekilde kullanılmıştır. Temel olarak kelimelerin anlamlarını içinde bulundukları cümleler ile, cümlelerin anlamlarını ise barındırdıkları kelimelere göre tahmin eder. Giriş metnini analiz ettikten sonra hangi kelimelerin birlikte kullanıldığını ve hangi ortak kelimelerin farklı cümlelerde görüldüğü gibi bilgileri çıkarır. Cümleler arasında fazla oranda ortak kelime bulunması, bu cümlelerin anlamsal olarak birbirine yakın olduğu varsayımını doğurur. Algoritma arka planda anlamsal vektörler kullanır ve bu vektörlerin doğru anlamlandırılabilmesi için geniş belge koleksiyonlarına ihtiyaç duyar. Bu nedenle derlem bazlı çalışmalar kategorisinde ele alınmaktadır.

Algoritmada paragraf başına kelime sayımlarını içeren bir matris (satırlar benzersiz kelimeleri, sütunlar ise paragrafları temsil eder) kullanılır. Bu matris büyük bir metin topluluğu kullanılarak yapılır ve benzerliği koruyarak sütun sayısını azaltmak için Tekil Değer Ayırıştırması (SVD) adı verilen matematiksel bir teknik kullanılır. SVD

algoritmasının kullanılmasıyla gürültüyü gidermek ve güçlü ilişkileri vurgulamak daha mümkün hale gelecek, beraberinde matrisin boyutsal gösterimi de azalacaktır.

LSA algoritması esasen herhangi bir eğitime veya dış bilgiye ihtiyaç duymayan denetimsiz bir yaklaşımdır. Fakat başarımı arttırmak için bazı çalışmalarda kelimeler daha sonra oluşturulan vektörler arasındaki açının kosinüsü alınarak karşılaştırılabilir. Ayrıca LSA vektörlerinin eğitilmesi için ilave metinler veya farklı bir derlem ile eğitilen vektörlerin kullanılması mümkündür.

1.2.2. Taksonomi Bazlı Semantik Benzerlik Metrikleri

Taksonomiye dayalı anlamsal benzerlik yöntemleri genellikle WordNet[17] gibi semantik ağlar ile hesaplamada edge-counting (kenar sayma) yöntemini kullanırlar. Bir semantik ağın veya bir taksonominin farklı özellikleri metin benzerliğinin hesaplanmasında kullanılabilir. WordNet çalışması taksonomi bazlı yöntemlerin başarısını oldukça arttıran önemli bir kaynaktır. Bu çalışma içerisinde isimler, fiiller, sıfatlar veya her biri farklı bir kavrama karşılık gelen eş anlamlılar gibi farklı grupları barındırır. DDİ çalışmalarında metinler arası ilişkilerin ortaya çıkarılması oldukça önemli bir yere sahiptir. Bu ilişkiler sayesinde normalde makine tarafından anlamlandırılması mümkün görülmeyen, herhangi bir şekilde yazım benzerliği barındırmayan kelime grupları daha anlamlı şekilde yorumlanabilmektedir.

a) Wu-Palmer

ISA taksonomisini kullanan anlamsal benzerlik ölçütlerinden birisi İngilizce fiillerin Mandarin Çincesine çevirisini konu alan bir makalede Wu ve Palmer [18] tarafından önerilmiştir. Algoritmada kavram çiftleri (karşılaştırılacak kelimeler) bir taksonomideki derinlik bilgisine göre kıyaslanır. Kök düğümden kelimelere kadar oluşan derinlik ve kelimelerden köke doğru giden en kısa yol sayesinde anlamsal yakınlık tespiti yapılmaya çalışılmıştır. Bu yöntem kavramsal sözlük tabanlı benzerlik hesaplamalarına ilaveten farklılık ölçümleri için de sıklıkla kullanılan bir yöntemdir [19]. Düğüm odaklı benzerlik tespiti çalışmalarında hiyerarşik bir gruplama söz konusu olmaktadır. Canlı grubuna ait ‘kedi’ ve ‘maydanoz’ kelimeleri diğer birçok benzerlik algoritması için herhangi bir anlam ifade etmemektedir. Bir alt basamak olan bitki grubuna dahil edildiğinde ‘pırasa’ ve ‘maydanoz’ kelimeleri bir önceki örneğe göre daha yüksek oranda anlamsal benzerliğe

sahip olacaktır. Fakat bu kavramlar diğer benzerlik algoritmaları için henüz hiçbir benzerlik anlamı ifade etmemektedir. Bu yaklaşımın sonucu olarak anlamsal derinliğe bağlı ilişkilerin tespiti mümkün olabilmektedir. Algoritmanın WordNet ve diğer alt semantik ağlar üzerinde gösterdiği başarıları Literatür Taraması başlığı altında değinilmiştir.

b) Wan & Angryk

İki kelime arasındaki anlamsal benzerliği hesaplamak için WordNet tarafından sağlanan genişletilmiş bağlam vektörlerinin kullanılması Wan ve Angryk [20] tarafından önerilmiştir. Karşılaştırılma yapılan kelimeler için (formüle göre m_1 ve m_2) birer vektör (v) oluşturulmaktadır. Bu vektörler WordNet'teki eşanlam, karşıtlık, aidiyet, parça-bütün gibi tüm ilişkilerin ve sözlük bilgilerinin yardımıyla oluşturulmuştur. Vektörlerin oluşturulmasından sonraki adımda bu vektörler arasındaki açının kosinüsü kullanılır. Algoritmanın formülü aşağıdaki gibidir.

$$sim_{W\&A}(m_1, m_2) = \cos(v_{m_1}, v_{m_2}) = \frac{v_{m_1} \cdot v_{m_2}}{|v_{m_1}| \cdot |v_{m_2}|} \quad (1.18)$$

1.2.3. Hibrit Metotlar

Hibrit yöntemler, derlem ve taksonomiye dayalı semantik benzerliklerin birleştirildiği yöntemlerdir. Bazı anlamsal benzerlik ölçümlerinde yoğunluk ve düğüm derinliği gibi farklı taksonomi özellikleri kullanılırken bazı yöntemlerde ise anlamsal ilişkilerden elde edilen kelime frekansları kullanılmaktadır. Bu kapsamda iki farklı yöntemi birleştiren algoritmalar alt başlıklar şeklinde aşağıda incelenmektedir.

a) Resnik Algoritması

İki kavram arasındaki benzerliği, ortak bilgi paylaşma dereceleri şeklinde yorumlayan Resnik [21] hem taksonomi bazlı hem de derlem kullanan hibrit bir yöntem önermektedir. Algoritmanın formülü aşağıdaki gibidir.

$$sim_{Res}(m_1, m_2) = IC(lso(m_1, m_2)) = \log^{-1}P(lso(m_1, m_2)) \quad (1.19)$$

Karşılaştırılan iki metin (m_1 ve m_2) aynı taksonomide olduğu için bu kavramların ortak bilgisi formülde 'lso' (lowest super-ordinate) olarak gösterilmektedir. Lso (m_1, m_2) bilgi içeriğini ise derlemde elde etmektedir. Formülde 'P' değeri ile gösterilen olasılık ise m için bir kullanımıyla daha karşılaşma ihtimalidir. Bu değer derlemde bulunan m adedinin o derlem içindeki toplam kavram sayısına bölünmesi ile hesaplanmaktadır. Eşitlikten de anlaşılacağı gibi m kavramının olasılığının artmasıyla beraber o kavrama ait bilgi içeriği (IC) azalır. Taksonominin hiyerarşik yapısına bağlı olarak daha alt katmanlarda (daha yakın ilişki düğümleri) bulunan kavramlar köke yakın üst kavramlardan daha yüksek benzerlik içerir. Dolayısıyla IC(m) değeri taksonomide köke yaklaştıkça düşmektedir.

b) Jiang & Conrath Algoritması

WordNet ile derlem kullanımını birleştiren bir diğer hibrit yöntem Jiang ve Conrath [22] tarafından önerilmiştir. Kavramlar arasındaki semantik benzerlik, edge-based (kenar sayımı) gösterimine ilaveten bilgi içeriği (IC) faktörü eklenerek hesaplanır. Elde edilen sonuç karar faktörü olarak değerlendirilir. Önerilen model; düğüm derinliği, yerel yoğunluk ve bağlantı tipi gibi birçok faktörü göz önünde bulundurmaktadır. Algoritmanın temel aldığı mantık ise, karşılaştırılan m_1 ve m_2 kavramlarının bilgi içeriği toplamı ile en bilgilendirici alt metin arasındaki fark, m_1 ve m_2 arasındaki mesafeyi oluşturur. Bu mesafe ile benzerlik oranı ters orantılıdır. Dolayısıyla mesafe büyüdükçe benzerliğin azalması beklenmektedir. Algoritmanın formül ile gösterimi aşağıdaki gibidir.

$$\begin{aligned} Dist_{J\&C}(m_1, m_2) &= IC(m_1) + IC(m_2) - lso(m_1, m_2) \\ Sim_{J\&C}(m_1, m_2) &= 1 - Dist_{J\&C}(m_1, m_2) \end{aligned} \quad (1.20)$$

1.3. Derin Öğrenme Yöntemleri ile Anlamsal Benzerlik Çıkarımı

Derin öğrenme bir sistemin her katmandaki gösterimini bir önceki katmandan hesaplamak için kullanılan, parametrelerin nasıl değişmesi gerektiğini gösteren ve büyük veri kümelerindeki karmaşık yapıyı geri yayılım algoritması kullanarak çözmeyi hedefleyen bir yöntemdir [23]. Çok katmanlı Yapay Sinir Ağları (YSA) kullanan bir öğrenme yapısına sahip en popüler yapay zeka yaklaşımlarından birisi olarak gösterilmektedir. Günümüz teknolojisinde birçok karmaşık problemin çözümü için

kullanılan yapay zeka ve derin öğrenme algoritmalarının sıklıkla kullanıldığı bir diğer alan da metin çalışmalarıdır. Yapılan çalışmanın konusu gereği bu bölümde Derin Öğrenmeye sadece metin işlemleri kapsamında kullanılan kelime temsili yöntemleri çerçevesinde değinilmiştir.

Kelime Temsili: DDİ çalışmalarında sıklıkla kullanılan dil modellemesi, kelimelerin içeriğini, anlamsal ve dilbilimsel benzerlikleri, diğer kelimeler ile aralarındaki ilişkileri vb. keşfetmek için kullanılmaktadır. Kelimeleri çok boyutlu vektörler ile eşleştirir bu vektörler bir YSA üzerinden eğitilerek Derin Öğrenme kapsamına taşınır [24]. Daha gelişmiş modelleme yöntemleri ile büyük belgeleri parçalamak veya bütün halde vektörel olarak temsil edebilmek mümkündür. YSA yaklaşımında düşük boyutlu vektörleri kullanarak parçadan bütüne genelleme özelliği kazandırılmaktadır. Bu yaklaşımın DDİ çalışmaları için en önemli faydası, elde edilen genelleme gücü sayesinde metinlerin anlamsal sınıflandırmalar şeklinde kümelere dahil edilebilmesidir [24].

Kelime temsili yöntemi ile geliştirilen farklı algoritmalar bulunmaktadır. Bu algoritmalar YSA iç katmanlarında ve kelimelerin vektörler ile ifade edilmesi sürecinde farklı yaklaşımlar taşımaktadır. Bahsi geçen algoritmalarından en popüler kullanılanları, matematiksel altyapıları ile birlikte farklılıkları göz önünde bulundurularak alt başlıklar halinde incelenmiştir.

1.3.1. Word2Vec

Word2Vec [25], Google çalışanı olan Mikalov ve arkadaşları tarafından kelimelerin vektörler halinde gösterimini sağlayan YSA tabanlı bir yöntemdir. YSA için girdi olarak bir metin verilir ve çıktı olarak ise vektör alınır. Kelimelerin vektörler ile ifade edilmesi bu kelimeler üzerinde matematiksel işlemlerin yapılabilmesine olanak tanımaktadır. Elde edilen vektörler ile kelimeler arasında anlamsal veya kavramsal ilişkilerin kurulabilmesi mümkün olacak ve kelimeler arasında antoloji kurulabilecektir. Bu işlemlerin yapılabilmesi vektörler arasındaki uzaklıkların ölçümüyle mümkün olmaktadır.

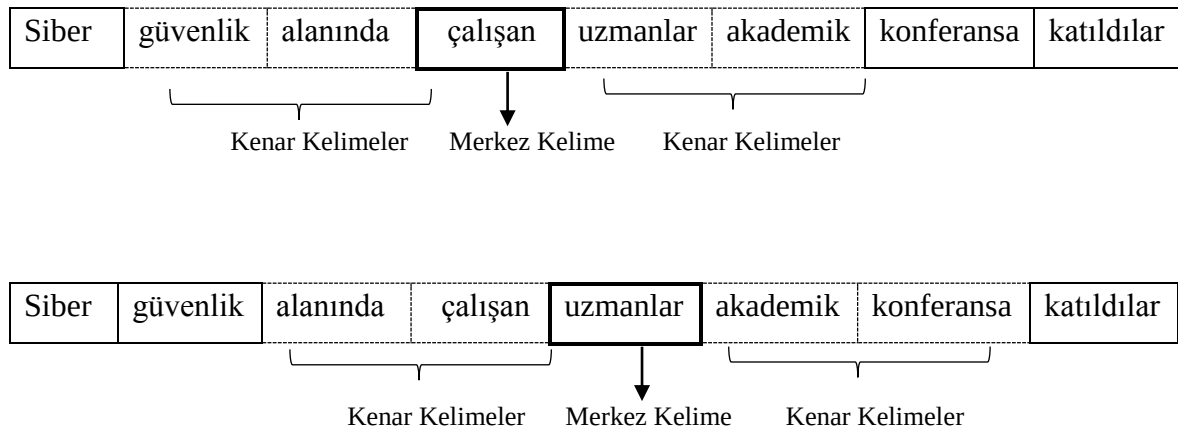
Yöntemin temel mantığı, insan algısında da olduğu gibi bir cümledeki bilinmeyen kelimeyi, cümledeki diğer kelimelerin anlamlarına göre tahmin etmektir. Her zaman tam doğru sonuç vermese dahi genellikle bir kelime, beraberindeki diğer kelimelerin anlamlarına göre yorumlanabilir. Word2Vec metodu da bu algıya benzer bir algoritma ile

çalışmaktadır. Fakat insan algısındaki gibi, o cümledeki diğer kelimelerin biliniyor olduğu varsayımı makineler için mümkün değildir. Bu nedenle bir kelimenin anlamına yönelik çıkarımlar yapmak için aynı işlemi bütün diğer kelimelere de uygulamak gerekmektedir. Böylelikle her bir kelime vektörü daha anlamlı bir hal alacaktır.

Bahsedilen işlemlerin gerçekleştirilmesi sonucunda birbirine benzeyen kelimelerin vektörleri de birbirine yakın olacaktır. Bir kelime için analiz yaparken o derlem içerisinde kelimenin kullanım adedi ne kadar artarsa o kelimeye yönelik oluşturulan vektör de bir o kadar anlamlı sonuçlar doğurur. Bu şekilde tespitlerin yapılabilmesi için büyük derlemlere ihtiyaç duyulması kaçınılmazdır. Büyük derlemlerde çalışmak ise DDİ çalışmalarında performans kaybına veya işlem karmaşıklığına sebep olabilmektedir. Word2Vec yöntemi bu dezavantajı ortadan kaldıracak şekilde büyük veriler ile çalışırken performans kaybı yaşatmamak için tasarlanmış bir gözetimsiz öğrenme yöntemidir.

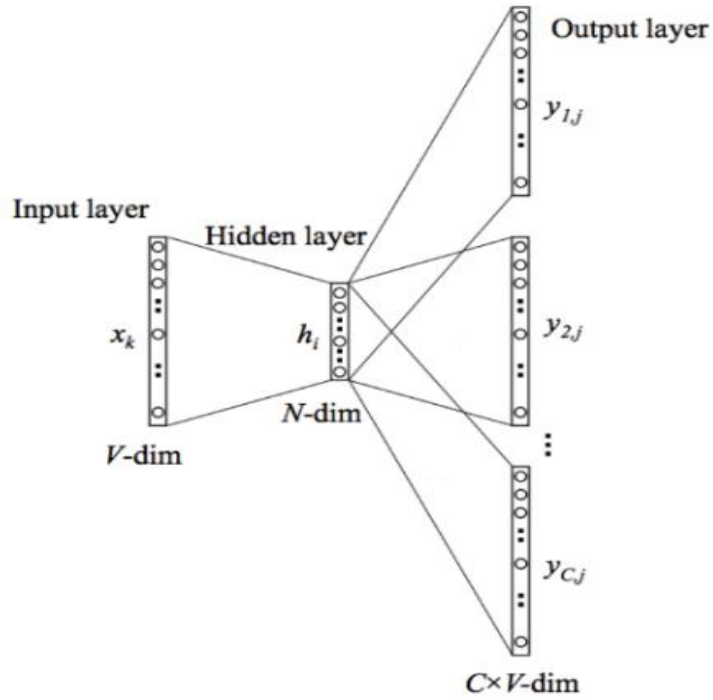
Word2Vec çalışmalarında kelimelerin vektörlere dönüştürülmesi için iki farklı mimari model bulunmaktadır. Bunlar Skip-Gram ve Continuous Bag of Words (CBOW) modelleridir.

Skip-Gram: Bu yöntemin amacı ortadaki bir kelimeye göre yanındaki diğer kelimeleri tahmin etmektir. Algoritmanın her adımında bir ‘merkez kelime’ alınır ve pencere büyüklüğüne göre o kelimenin sağındaki/solundaki diğer kelimeler tahmin edilir. Merkez kelime cümledeki en merkezindeki kelime olmak zorunda değildir zira cümledeki her bir kelime için aynı işlem gerçekleşecek ve her kelime merkez kelime olarak değerlendirilecektir. Bu işlem adımları aşağıdaki örnek üzerinden gösterilmiştir.



Şekil 1.1. Skip-Gram mimarisi yapısı

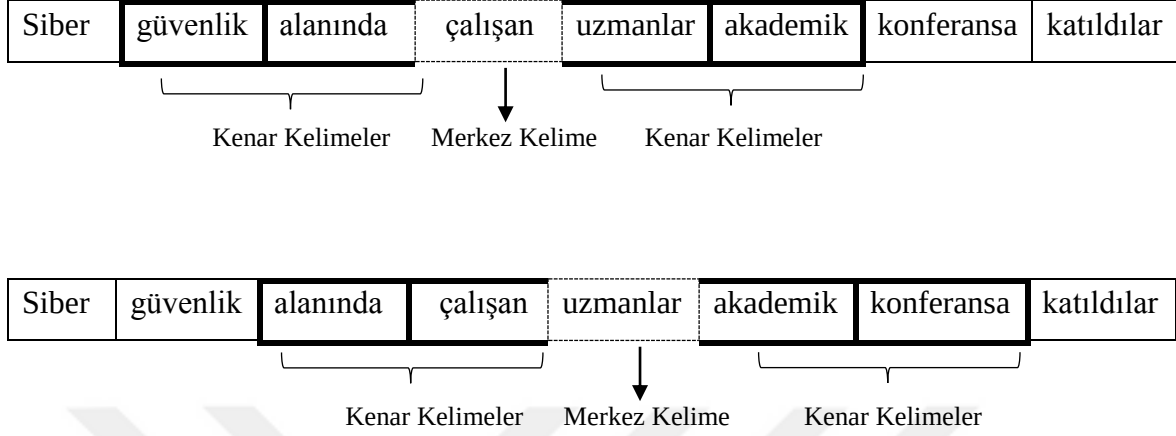
Şekil 1.1'e göre pencere büyüklüğü 2 olarak görülmektedir. Pencere büyüklüğü kavramı, merkez kelimenin sağından ve solundan kaç adet kelimenin tahmin edileceğini belirler. Oluşturulacak Word2Vec modelini örnekteki merkez kelime olan 'çalışan' kelimesine göre sağında ve solundaki ifadelerin neler olabileceğine dair olasılıklar verecektir. Modelin eğitilmesi sırasında bu kelimenin içinde bulunduğu diğer cümleler de incelenecek ve en sık birlikte geçtiği diğer kelimelerin olasılığı yükselecektir. Örneğe göre algoritma 4. ve 5. adımlarda görülmektedir. İlk adımda cümledeki ilk kelimesi olan 'siber', merkez kelime olarak seçilecek fakat pencere boyutuna göre solunda kelime bulunmadığı için sadece sağındaki iki kelime ile işlemler yapılacaktır. Algoritmaya her adımda girdi olarak verilen merkez kelimeye karşılık, çıktı olarak kenardaki kelimelerin tahminleri dönecektir. Skip-gram algoritmasının YSA üzerindeki çalışma mantığı Şekil 1.2'de gösterilmektedir.



Şekil 2.2. Skip-Gram mimarisi modeli [26]

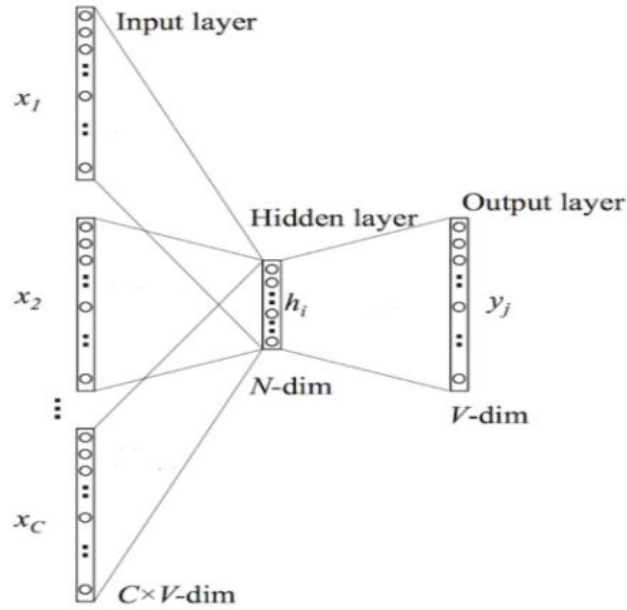
CBOW: Skip-gram modelinin tam tersi şekilde kenardaki kelimelerden ortadaki kelimeyi tahmin etmek üzere kurgulanmıştır. YSA girdi olarak kenardaki kelimeleri alırken çıktı olarak orta kelimeyi verir. Bir cümle için ilk kelimedenden itibaren adım adım skip-gram mantığının tam tersi işlenir. Her bir adımda kayan pencere boyutuna göre kenar kelimeler algoritmada tahmin girdileri olarak kullanılır ve orta kelimeye yönelik tahmin

sonucu alınır. Skip-gram modeli için verilen örnek cümle, CBOW modeli için aşağıdaki gibi incelenmektedir.



Şekil 3.3. CBOW mimarisi yapısı

Şekil 1.3'te pencere boyutu 2 olan bir CBOW modeli görülmektedir. Algoritmanın her adımında bir kelime seçilerek sağından ve solundan ikişer kelime girdi olarak değerlendirilir. Şayet kelimenin sağında veya solunda pencere boyutu kadar kelime yoksa mevcutta olan kelimelerle devam edilir. Modelin çalışma mimarisi Şekil 1.4'te gösterilmektedir.



Şekil 1.4. CBOW mimarisi modeli [26]

Skip-gram ve CBOW modellerinin ikisi de Word2Vec için kullanım alanlarına göre uygun bulunmaktadır. Bu modelleri kıyaslayan Mikolov [25], çalışmasında skip-gram modelini CBOW'a göre daha yavaş fakat küçük çaplı derlemlerde daha başarılı olduğu sonucuna varmıştır. Ayrıca nadir sıklıkta geçen kelimelerin anlamlı şekilde temsilinde CBOW'a göre daha başarılı olduğunu ifade etmektedir. CBOW modelinin avantajları olarak da daha hızlı eğitildiğini ve sık kullanılan kelimelerin temsilinde daha başarılı olduğunu ifade etmiştir. Dezavantaj olarak CBOW ile alınan başarıyı arttırabilmek için Skip-gram modeline göre daha büyük derlemlere ihtiyaç duyulacağı gözlemlenmiştir.

1.3.2. Glove

Kısaltılmamış hali ile 'Global Vectors for Word Representation' şeklinde Pennington ve arkadaşları [27] tarafından yapılan çalışma ile önerilmiştir. Word2Vec benzeri bir yaklaşımla kelimeleri vektörler ile ifade eder. Fakat yapılan çalışmada daha küçük vektör boyutları ve daha küçük derlemler üzerinde başarıyı arttırdığı belirtilmiştir. Glove yöntemini diğer kelime temsili yöntemlerinden ayıran özelliklerinden birisi; çok sık kullanılan kelimelerin hesaplamasında bir üst limit tutmasıdır. DDİ çalışmalarında en sık geçen kelimelerin genellikle 'stopwords' olarak adlandırılan bağlaç, edat vb. şeklinde anlamsal değeri az olan kelimeler olduğu görülür. Hazırlanan derlem için kelimeleri vektörlere çevirirken bu ifadelerin çok sık geçmesi yanındaki diğer kelimelerin tahmini ve temsilinde sapmalara neden olması muhtemeldir. Bu sorunu ortadan kaldırmak için Glove yönteminde sık kullanılan kelimelerin belirli bir sayıdan sonraki kullanımları görmezden gelinir.

Glove modeli Word2Vec'ten farklı olarak derlem içerisindeki tüm kelimeleri inceler ve her bir kelimenin kaç adet tutulduğunu sayarak ilerler. Bu işlem safhasında diğer vektörlerin optimizasyonu yapılmaz. Tüm sayım işlemi gerçekleştikten sonra vektörler güncellenir ve optimize edilir. Fakat Word2Vec modelinde bir kelime ile tekrar karşılaşıldığı her an vektörlerin optimizasyonu anlık yapılarak devam edilir. Yapısal bu farklılık nedeniyle Glove, Word2Vec'ten daha hızlı bir eğitim süresine sahiptir. Modelin formülle gösterimi aşağıdaki gibidir.

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^w f(P_{ij}) (u_i^T \cdot v_j - \log P_{ij})^2 \quad (1.21)$$

Formülde global kelime sayıları ile kelime sayılarından oluşturulan matrisin satır sayısı u , sütun sayısı ise v şeklinde gösterilmiştir. W ile oluşturulan sözlüğün boyutu gösterilirken f fonksiyonu ile ağırlık fonksiyonu ifade edilmektedir.

1.3.3. FastText

FastText modeli 2016 yılında Facebook firmasının Yapay Zeka Araştırma Birimi (Facebook AI Research) tarafından geliştirilmiştir. Word2Vec altyapısını kullandığı için bu modelin bir uzantısı olarak da tanımlanmaktadır. Temel yapıda Word2Vec modelinin izlerini taşısa da farklılaştığı kısmı n -gram temelli bir çalışma prensibine dayanır [28]. Diğer kelime temsili modellerinden farklı olarak bir kelimeyi de kendi içerisinde n -gram mantığı ile parçalar. O kelimeye ait vektör ise kelimenin n -gram parça vektörlerinin toplamından oluşur. Bu yaklaşım sayesinde FastText ile eğitim veri setinde bulunmayan kelimelerin de temsili mümkün olabilmektedir. Nadir geçen kelimelerin n -gramlarının ortaya çıkma ihtimalleri daha düşük olduğu için FastText sayesinde bu kelimelerin daha doğru temsili mümkün olabilmektedir. Ayrıca FastText ve Glove, birçok güncel dil için hazırlanmış kelime vektörlerini geliştiricilere sunmaktadır. Elde edilen bu hazır vektörler proje bazlı farklı eğitim kümelerinde eğitilerek kullanım kolaylığı sağlamaktadır.

1.4. Literatür Çalışması

Yapılan çalışmaya yol göstermesi ve literatürdeki ilerlemelerin takibi açısından geçmiş dönemlerde yapılan araştırma ve çalışmalar bu başlık altında incelenmiştir. Üzerinde çalışılan araştırma Türkçe metinlere odaklandığı için özellikle Türkçe metinler ile yapılmış intihal tespiti, metin eşleştirme, metin sınıflandırma benzeri DDİ çalışmaları incelenmiştir. Türkçe dili için yapılan çalışmalara ilaveten literatüre katkı sunan farklı dil odaklı çalışmalara da yer verilmiştir. Yapılan literatür taramasında anlamsal benzerlik ile yapısal benzerlik tespiti sistemleri bir grupta, derin öğrenme tabanlı kelime temsili modellerinin benzerlik tespitine yönelik çalışmaları ise ayrı bir grupta incelenmiştir.

1.4.1. Yapısal ve Anlamsal Benzerlik Çalışmaları

Yücesoy [29], yaptığı tez çalışmasında Türkçe dokümanlar için benzerlik tespitine yönelik alternatif bir yöntem önermiştir. Bu yöntem kavramsal sözlük kullanmakta ve içeriğini internet ortamından sağlanan datalar ile temin etmektedir. Kullanılan yöntemde çalışılacak metin üzerinde bir gezinti yapılarak metne dair en önemli kelimelerin seçilmesi yenilikçi bir yaklaşım olarak paylaşılmıştır. Bu yaklaşıma ilave olarak Wu-Palmer algoritması eklenmiş ve iki katmanlı bir işlem serisi ile metinler arasındaki anlamsal benzerlik yorumlanmıştır. Çalışmanın sonuçlarına göre kavramsal sözlük kullanmayan algoritmalar ile kıyasında, Kosinüs algoritması daha yüksek sonuçlar vermiş ve bu durum kullanılan sözlüğün yeterli olmaması ile açıklanmıştır.

Soğancıoğlu [30] tarafından yapılan tez çalışmasında biyomedikal dataların anlamsal benzerlik çalışmalarına katılabilmesi için bir veri seti hazırlanmış ve bu veri üzerinde çalışılacak yöntem önerilmiştir. Biyomedikal özelinde çalışılmasının nedeni kavramların günlük veya akademik çalışmalara oranla daha terimsel ve karmaşık olmasıdır. Bu karmaşıklığa rağmen anlamsal ilişkilerin çıkarılabilmesi için, LSA ve paragraf vektör temsili modelleri kullanılmıştır. Kullanılan yöntemler içerisinde paragraf vektör temsili çalışmasının daha başarılı sonuçlar verdiği gözlemlenmiştir.

Amasyalı ve Beken [31] tarafından yapılan çalışmada anlamsal benzerliklerinin ölçülmesi sonrası metin benzerliğinde kullanılmak üzere kelime bazlı benzerlik tespiti üzerine odaklanılmıştır. Bu çalışmanın kelimelere yönelik anlamsal benzerliklerinin çıkarılması sürecinde BoW yaklaşımı kullanılmış ve elde edilen sonuçlar sonrası büyük metinler kelimelerin vektörlerinden faydalanılarak sınıflandırılmıştır. Yapılan çalışmadan çıkan önemli bir sonuç; benzerlik matrisleri bulunurken kullanılan kelimelerin doküman içerisinde birlikte geçme sayısı ile birlikte geçtiği doküman sayısının ayrı ayrı kullanımı sonuçta ciddi bir değişikliğe sebep olmamıştır.

Kairaldeen [32] tarafından yapılan tez çalışmasında anlamsal benzerlik hesaplama algoritmaları karşılaştırılmış ve kategorilere göre başarılı algoritmalar incelenmiştir. Elde edilen sonuçlara göre derlem bazlı algortimaların (LSA vb.) WordNet bazlı çalışan algoritmalarla göre daha başarılı olduğu gözlemlenmiştir. Diğer algoritmalar şeklinde yardımcı yöntemler olarak bahsedilen algoritmalar arasından da Kosinüs Algoritmasının en yüksek başarıyı elde ettiği görülmektedir.

Başak [33] tarafından geliştirilen uygulamada Kosinüs, Jaccard, Dice ve Tonimoto algoritmalarının metin benzerliğindeki performansı kıyaslanmaktadır. Ayrıca N-gram temelli yaklaşımla 1-4 arası harften oluşacak şekilde metinlerin bölünmesi mümkün hale getirilmiştir. Aynı metinler üzerinden farklı N-gram ölçüleri ile yapılan deneyler sonrasında Kosinüs Algoritmasının performansı diğerler algoritmalarla oranla daha başarılı bulunmuştur.

Kaya [34], 2015 yılında yaptığı doktora tezi kapsamında Türkçe metinleri kullanarak metin benzerliği yöntemleri ile bir intihal tespit sistemi önermiştir. Geliştirilen sistem öğrencilerden alınan ödev metinlerinin kıyaslanması ile eğitilmiştir. Yapılan çalışma kapsamında Kosinüs ve Hırslı Metin Eşleme Algoritmaları kullanılmıştır. Bu algoritmalar sırasıyla iki katmanlı eleme yöntemi ile kullanılmıştır. Kosinüs algoritmasının verdiği sonuçlar belirli bir eşiği geçerse bir diğer algoritmadan ikinci test yapılmış ve oradan da eşik üstü değer alınması durumunda intihal tespiti doğrulanmıştır. Çalışmanın sonraki bölümlerinde, gerçek hayatta uygulamanın kullanılması sonucu öğrencilerde görülen intihal hareketliliğinin azaldığı gözlemlenmiştir. Elde edilen sonuçlara göre önerilen yöntemin başarısı üniversite öğrencilerinin verileri ışığında yorumlanmıştır.

Korencic [35] ve arkadaşları tarafından yapılan çalışmada, haber metinleri ile haber başlıklarının uyumu üzerine bir araştırma yer almaktadır. Çalışma, haber bütünlüğü içerisinde konu tutarlılığının tespiti ve önemi üzerine de yoğunlaşmaktadır. Bu doğrultuda hem haber metinleri içerisindeki konuların anlamsal yakınlığı hem de içerik – başlık uyumundaki anlamsal yakınlığın ölçülmesi gerekmektedir. Elde ettikleri sonuçlar incelendiğinde, TF-IDF veya sözcük torbası (Bag of Words) yöntemlerini kullanarak dokümanları vektörleştiren yaklaşımlar oldukça başarılı sonuçlar üretmiştir. Bu yaklaşımlar, yapısal benzerlik algoritmalarına destek amaçlı veya doğrudan ana yöntem olarak kullanılabilirler. Çalışmanın önerdiği yöntem ile metinler arası tutarlılığı kavrayabilmek için, kelime vektörleriyle alınan başarı vurgulanmaktadır.

Metin benzerliğine yönelik çalışmaların çoğunlukla İngilizce için yapılması ve bu amaçla geliştirilen yöntemlerin Türkçe metinlerde yeterli sonucu veremediği düşüncesiyle Dursun ve Sönmez [36] tarafından yeni bir yöntem önerilmiştir. Önerilen yöntemin hedefi Türkçe metinlerin karşılaştırılmasında en sık görülen hataları incelemek ve bu hatalara çözümler üretecek yeni bir yaklaşım getirmektir. Genellikle yazım hatalarından kaynaklı sorunlara ve Türkçe'nin yapısı gereği kelimelerin sondan ek almasına yönelik sorunlara

DDİ metotları ile çözümler önerilmiştir. Ardından temizlenmiş veri üzerinde yapılan testlerde Jaro-Winkler ve Levenstein algoritmaları kullanılmıştır.

Nisar [37] ve arkadaşları, dizge tabanlı ontolojik uyum teknikleri üzerine yaptıkları çalışmada, çeşitli metin benzerliği algoritmalarının performansını kıyaslamaktadır. Bilgisayar bilimlerinin genelinde kullanım alanı bulmasına rağmen semantik web yaklaşımıyla birlikte metinlerin ilişkilendirilmesi ekstra önem kazanmaktadır. Bu kapsamda ilişkilerin ve benzerliklerin tespiti için seçilen algoritma başarısı önemli bir faktör olarak görülmektedir. Yapılan çalışmadan alınan sonuçlar, tekil algoritma kullanımına oranla algoritmaların birlikte kullanımından daha verimli sonuçlar alındığını göstermektedir. Örnekler üzerinde en başarılı sonucu TF-IDF ile desteklenmiş Jaccard Algoritmasının ve N-gram yaklaşımlı algoritmaların verdiği gözlemlenmiştir.

Carmona [38] ve diğerleri anlamsal benzerlik kapsamında intihal tespiti yapmak adına bir çalışma gerçekleştirmiştir. Çalışmada vurgulanan başlıca sorun, cümlelerin aynı anlama gelmesine rağmen türlü değişiklikler ile intihal taramalarına takılmamasıdır. Cümle içerisindeki kelimelerin yer değişimi ve benzer kelime kullanımı, intihal tarama araçlarını yanıltabilmektedir. Bu probleme çözüm olarak mevcut benzerlik algoritmalarının hibrit şekilde kullanılması önerilmektedir. Jaccard ve Levenshtein Algoritmalarını anlamsal benzerlik algoritmaları ile hibrit şekilde kullanarak alınan sonuçlarda ciddi başarı elde edildiği vurgulanmaktadır. Bu çalışma, intihal tespitinde sadece yapısal benzerlik tespiti algoritmalarının yetersiz kaldığını göstererek literatüre önemli bir katkı sunmaktadır.

Mihalcea [39] ve arkadaşları tarafından yapılan araştırmada derlem tabanlı benzerlik sistemleri ile taksonomi tabanlı sistemler kıyaslanmıştır. Algoritmaların tekil kullanımına oranla birleşik kullanımı veya sırayla eleme yöntemiyle kullanımının daha başarılı sonuçlar verdiği gözlemlenmiştir. Yapılan çalışmada Resnik Algoritması ile Wu-Palmer algoritmasının birlikte kullanımı ile başarılı sonuçlar elde edilmiştir. Bu sonuçlara ilaveten vektörel gösterimle Kosinüs Algoritmasının TF-IDF ile kullanılması önerilen bir diğer kombin olmuştur.

Kiourtis [40] ve arkadaşları tarafından yapılan çalışma sağlık verilerinin ontolojik eşleşmesini konu almaktadır. Tıbbi dataların ortak çalışma gerektiren kurumlar veya kişiler arasındaki trafiğinde türlü problemler açığa çıkmaktadır. Bu problemlerden çalışmaya konu olanı ise ortak formata çevrilmesi istenen tıbbi verilerdir. Mevcut çalışmalarda bir verinin istenilen formata veya dile çevrilmesinde birebir çeviri teknikleri kullanılmaktadır. Bu yöntem ise ifadelerin yanlış anlaşılması ve terimlerin algılanamaması sorunlarını

beraberinde getirir. Bu nedenle metinlerin eşleşmelerini kontrol edebilmek ve güvenilir algoritmaları tespit etmek için deneyler yapılmıştır. Birçok benzerlik algoritması içerisinde en güvenilir sonuçlar Levenstein Algoritması ile elde edilmiş, bu algoritmayı Kosinüs, Jaro-Winler ve Jaccard Algoritmaları takip etmiştir.

Mitkov [41] tarafından yapılan çalışma, çoklu kelime barındıran (phrase) deyimlerin bir sözlük kullanılmadan otomatik çevirisini içermektedir. Özellikle İngilizce dilinde görülen çok kelimeli ifadeler için sözlük kullanımını ve çeviri sistemlerini ortadan kaldıran bir öneri sistemi geliştirilmiştir. Bu sistemde İngilizce ve İspanyolca dillerine odaklanılmış ve tatmin edici başarılar gözlemlenmiştir. Bahsi geçen öneri sistemini sunabilmek için kelimelerin anlamsal olarak karşılıklarını bulabilmek gerekmektedir. Bu gereklilik kelime temsili yöntemleri (Word2Vec) ile çözülmüştür. Kelime alternatifleri için öneri almadan önce minimum eşik değeri ise Kosinüs Benzerliği oranıyla 0.45 olarak belirlenmiştir. Çalışma sonucunda İngilizce ve İspanyolca için başarılı tespitlere ulaşıldığı ve diğer diller için de yöntemin başarıyla kullanılabileceği vurgulanmıştır.

Başlıkların eşleştirilmesi amacıyla benzerlik ölçütlerinin kıyaslandığı çalışmada Gali [2] ve arkadaşları 21 farklı algorithmadan alınan sonuçları kıyaslamıştır. Bu algoritmalar önceki bölümlerde bahsedildiği üzere karakter bazlı, N-gram bazlı, token bazlı ve karma algoritmalar olarak ele alınmıştır. Elde edilen sonuçlara göre Soft TF-IDF karma metotlar arasında en verimli sonuçları veren algoritma olarak gözlemlenmiştir.

Sun [42] ve arkadaşları tarafından yapılan ontoloji uyumu çalışması için benzerlik ölçütü metrikleri kıyaslanmıştır. Yapılan çalışma ile elde edilen sonuçların bir diğer benzer çalışmayla [37] aynı doğrultuda olduğu gözlemlenmektedir. Çıkan sonuçlardan birisi ise TF-IDF kullanımının vektörel gösterimde elde edilen başarıyı arttırdığıdır.

Sravanthi [43] tarafından yapılan çalışmada cümlelerin semantik benzerliklerini ölçen algoritmalar karşılaştırılmıştır. Çalışmada WordNet tarafından sunulan işlenmiş veriler ile birlikte çeşitli İngilizce test cümleleri kullanılmıştır. Karşılaştırılan yöntemler arasında Wu-Palmer Algoritmasının en düşük performansı gösterdiği gözlemlenmiştir. Semantik destekli Kosinüs Benzerliği ile özellik çıkarımına dayanan ilişkisel yöntem ise test verilerinde en yüksek başarı skorlarına sahip algoritmalar olmuştur.

Türkçe metinlerin sınıflandırılmasına yönelik anlamsal benzerlik algoritmaları ve Kosinüs Algoritmasını hibritleyen bir metot Yücesoy [44] ve Öğüdücü tarafından önerilmiştir. Bu çalışmanın Türkçe metinlerin arasındaki anlamsal benzerlikleri inceleyen ilk çalışma olduğu ve Kosinüs Algoritması ile en etkili sonuçların alındığı vurgulanmıştır.

1.4.2. Kelime Temsili Tabanlı Benzerlik Çalışmaları

Sosyal medya profillerinin benzerliklerini ölçmek üzere Erdoğan [45] tarafından yapılan tez çalışmasında DDİ tekniklerine ilaveten Word2Vec metodu kullanılmıştır. Kelime temsili yaklaşımı ile önce kelime vektörlerinin çıkarılması ardından ise paragraf vektörlerin çıkarılması mümkün olmuştur. Çalışmada sosyal medya kullanıcılarının benzer ifadeleri kullanma sıklıkları incelenmiş ve anlamsal ilişkiler yorumlanmıştır. Kelime temsili ve paragraf temsili yöntemlerinin mevcut başarımı önemli oranda yükseltmediği gözlemlenmiş ve bu durum eğitim verisinin az oluşuyla ilişkilendirilmiştir.

Özer [46] ve arkadaşları tarafından yapılan çalışmada Türkçe yazılan twitter verileri için diyakritik restorasyonlar önerilmektedir. Kelimelerin Türkçe özel karakterler barındırması durumunda sıklıkla hatalı kullanıldığı gözlemlenmiştir. Bahsedilen özel karakterler yerine alfabedeki noktasız halinin veya yakın bir diğer harfin kullanımı anlamı değiştirmektedir. Cümle içerisindeki bu hatalı yazımların tespiti ve yerine doğru kelimelerin getirilebilmesi için yazarlar tarafından Word2Vec metodu kullanılmıştır. Word2Vec ile kelimenin anlamı tespit edilmiş, yerine alternatif kullanılacak kelimeler önerilmiştir. Elde edilen sonuçlara göre, %94 doğruluk performansı ile hataların %37'sinin düzeltilmesi sağlanmıştır.

Türkçe haber metinlerinin sınıflandırılması konulu çalışmaları ile Acı [47] ve Çırak, kelimelerin semantik değerlerini de çalışmaya katabilmek için Word2Vec kullanarak mevcut çalışmalardan daha yüksek bir başarı elde etmişlerdir. Bu başarıyı yorumlarken; Türkçe metinler için yeterli DDİ çalışmasının olmamasından ötürü kelimelere yönelik anlamsal ilişkilerin ortaya konduğu çalışmaların eksikliğine işaret edilmiştir. Bu problemi çözebilmek için önceden eğitilmiş Word2Vec vektörlerinden faydalanılmıştır.

Öztürk [48] ve Can tarafından yapılan çalışmada sözcük köklerinin sözdizimsel kümelemesini yapmak için olasılıksal yöntemler ve Word2Vec ayrı ayrı kullanılarak kıyaslanmıştır. Çalışmadan çıkarılan bulgulara göre köklerin birbirine benzer kategorilerde isim, fiil vb. şekilde sınıflandırılması konusunda Word2Vec modeli daha başarılı sonuçlar vermiştir. Word2Vec modeli içerisinde CBOW ve Skip-gram modelleri de ayrı ayrı denenmiş ve Skip-gram yöntemi ile daha başarılı sonuçlar aldığı gözlemlenmiştir. Ayrıca Skip-gram için pencere büyüklüğünün azaltılması ile Türkçe kelimeler için başarının arttığı görülmüştür.

Türkçe tweetleri değerlendirerek duygu analizi çıkarımı konusunda çalışmalar yürüten Ayata [49] ve Saraçlar, çalışmalarında makine öğrenmesi yöntemlerine ilave olarak Word2Vec kullanışlardır. Kelime temsiliyle tüm kelimelerin vektörleştirilmesi sonrasında Rastgele Orman ve Karar Destek Makinesi Algoritmaları kullanılmıştır. Kelime temsili desteği ile iki algoritmanın da yalın kullanımına oranla elde ettiği başarının arttığı gözlemlenmiştir.

Şahin [50] tarafından yapılan araştırmada Türkçe dokümanların sınıflandırılması amacıyla Word2Vec kullanımına başvurulmuştur. Yapılan çalışma ile öncelikle kelimeler, ardından da metinler vektörler ile gösterilmiştir. Metinlerin vektörel gösterimi, barındırdığı kelime vektörlerinin ortalaması alınarak hesaplanmıştır. Elde edilen metin vektörleri, makine öğrenmesi metotları yardımıyla sınıflandırılmış ve 0.92 F ölçüm değeri gibi yüksek bir sonuç alınmıştır. Çalışmadan çıkan temel sonuç ise Word2Vec yaklaşımının klasik TF-IDF tabanlı sınıflandırma yaklaşımına göre daha başarılı olduğunun görülmesidir.

Bilgin [51] ve Köktaş tarafından Türkçe metinlerden duygu çıkarımı için Makine Öğrenmesi temelli çıkarımlar yapılan bir çalışma gerçekleştirilmiştir. Duygu analizi, DDİ çalışmalarına sıklıkla konu olan bir alandır. Fakat bu çalışmada ilk kez Makine Öğrenmesi algoritmalarına ilaveten Word2Vec yöntemi kullanılmıştır. Elde edilen bulgular bu yaklaşımın oldukça başarılı olduğunu kanıtlar niteliktedir. Çünkü çalışmada kullanılan 5 farklı sınıflandırma algoritmasının tamamında da Word2Vec desteği ile başarımın arttığı gösterilmektedir.

Türkçe kısa mesajların sınıflandırılmasına yönelik yapılan araştırmada Karasoy [52] ve Ballı sınıflandırma algoritmalarına gönderilecek veriyi Word2Vec kullanarak eğitmişlerdir. Word2Vec ile kelimelerin vektörler ile temsili sağlanmış ve anlamsal nitelikler çıkarılmıştır. Bu işlemlerin sonrasında Rastgele Orman Algoritması ile gelen mesajların spam olup olmadıklarına yönelik kararlar alınması mümkün hale gelmiştir. Kelimelerin vektörler ile gösterilmesi sayesinde her bir kelimenin spam içeren metinlere görece matematiksel uzaklığı hesaplanmış ve bu bilgiye dayanarak sınıflandırma işleminde %94'ün üzerinde bir başarı sağlanmıştır.

Arabacı [53] ve diğerleri tarafından yapılan çalışmada Türkçe için benzer cümlelerin tespitine odaklanılmıştır. Bu amaç doğrultusunda anlamsal ve kavramsal yakınlığa dikkat ederek yakın cümleleri bulmaya yönelik bir uygulama geliştirilmiştir. Yöntem olarak Word2Vec ile birlikte Fisher kodlaması kullanılmıştır. Elde edilen sonuçlarda Word2Vec içi yaklaşımlardan Skip-gram yöntemi ile daha yüksek başarı değerlerine ulaşılmıştır.

Firmalar için süreç yönetimini kolaylaştırmak ve büyük iş süreçlerini gereksinim analizi dokümanlardan çıkarabilmek için Shahzad [54] ve diğerleri tarafından anlamsal çıkarım testleri yapılmıştır. Süreç modeli oluştururken, farklı tarifler ile aynı iş parçasına işaret edilebileceği çıkarımıyla sözcükleri anlamak ve benzer ifadeleri saptamak için türlü algoritmalar denenmiştir. Yapısal benzerlik algoritmaları arasında Kosinüs ve Jaccard Algoritmaları önde olsa dahi kelime temsili yöntemleri ile diğer 8 benzerlik algoritmasından daha başarılı sonuçlar elde edilmiştir. Çalışmada kullanılan 4 farklı derlem ile aynı algoritmalar sınanmış, Word2Vec, Glove ve Fasttext F1 başarı ölçütünde en üstte bulunmuştur.

Bilimsel yayınlarda kullanılan anahtar kelimeler arasındaki anlamsal ilişkilerin sonuçlarını araştıran Hu [55] ve arkadaşları, çalışmalarında Word2Vec'i kullanmaktadır. Word2Vec sayesinde çıkarılan anlamsal ilişkiler ile birlikte anahtar kelimelerin yakınlığını ölçmek mümkün olmaktadır. Bu yakınlıkların tespiti sayesinde, literatür taraması ve atıf alma sürecinde konunun popülerliğinin artırılabilmesi öngörülmüştür. Coğrafi doğa tehlikelerini içeren eğitim setiyle elde edilen bulgular, çalışma ekibini doğrular nitelikte sonuçlar vermiştir. Anahtar kelimelerin doğru sınıflandırılması ile beraberinde kullanılan diğer anahtar kelimelerin popülerliğini arttırdığı gözlemlenmiştir.

Dillenberger [56] tarafından yapılan tez çalışmasında doğrudan Word2Vec ile ölçülebilir ilişkiler incelenmektedir. Çalışma kapsamında Word2Vec yöntemine 100.000 üzeri kelime ile veri sunulduğunda benzerlik ve yakın anlam tahminlerinin başarılı sonuçlar verdiği belirtilmiştir. Bir diğer bulgu olarak, nadir kullanılan kelimelerin analojisi için Word2Vec modelinin oldukça kullanışlı olduğu vurgulanmaktadır. Derin Öğrenmede sık kullanılan hiper parametrelerden taksonomi bazlı benzerliklerin tespiti ve eş anlam tespitinde Word2Vec yönteminin performansı, semantik yakınlık çalışmalarına göre daha başarılı bulunmuştur.

Orkphol [57] ve Yang, kelimelerin anlamsal ayrımlarını ve yakınlıklarını ölçmek için yaptıkları çalışmada Word2Vec ile Kosinüs Teoremini ortak olarak kullanmaktadır. Kelimelerin eğitiminde hazır vektörler sunan WordNet ve ilave duygu çıkarımı odaklı derlemler kullanılmıştır. Çalışmada, Word2Vec modelinden alınan sonuçların hem performans hem de anlamsal ilişki olarak LSA'dan daha iyi sonuçlar verdiği gözlemlenmiştir. Fakat performansı arttırabilmek için önemli bir etken olarak sadece belirli bir eşiği geçen benzerlikler üzerine odaklanılmıştır. Bu eşiği belirlemek ve sonuçta elde edilen vektörlerin yakınlığını ölçmek için çalışmada Kosinüs Teoremi kullanılmaktadır.

Duygu çıkarımı çalışmalarında kelime temsili yöntemlerinin başarısı, Lauren [58] ve arkadaşları tarafından diğer Makine Öğrenmesi temelli çalışmalar ile kıyaslanmaktadır. Bu kapsamda, Word2Vec ve Glove yöntemleri hem kendi aralarında hem de diğer algoritmalar ile ayrı ayrı karşılaştırılmıştır. Varılan sonuçlarda tahminler için Word2Vec yönteminin oldukça başarılı sonuçlar verdiği gözlemlenmiştir.

Zhu [59] ve arkadaşları Biyomedikal alanda bilgi alma ve tavsiye sistemleri oluşturabilmek için milyonlarca kelimedenden oluşan bir derlem oluşturmuştur. Bu derlem tamamen Biyomedikal çalışmalardan alınan kelimeleri kapsamaktadır. Oluşturulan bu derlem ile terimler arasındaki semantik ilişkilerin tespiti, terimlerin anlam benzerliklerinin karşılaştırılması gibi türlü testler yapmışlardır. Terimlerin anlamsal yakınlıklarını saptamak için Word2Vec yöntemi tercih edilmiştir. Vektörel gösterime sahip kelimelerin benzerlik tespitinde ise Kosinüs Benzerliği Algoritması kullanılmıştır. Elde edilen bulgulara göre bir noktadan sonra artan derlem büyüklüğü sonuçları etkilememektedir. Oldukça geniş bir derlem üzerinde yeterli anlamsal bağlantıların çıkarıldığı neredeyse %100'e yakın doğrulukla benzerlik tespitlerinin yapıldığı belirtilmiştir.

Bir diğer Biyomedikal alana özgü anlamsal benzerlik tespiti çalışması Chen [60] ve arkadaşları tarafından yapılmıştır. Bu çalışmada benzer çalışmalardan [59] farklı olarak, eğitim setini oluşturan derlem iki farklı yöntem ile hazırlanmıştır. İlk derlemde Wikipedia üzerinden sadece sağlık odaklı başlıkların altındaki metinler toplanmıştır. İkinci derlemde ise WordNet ile Wikipedia'dan konu gözetmeksizin genel başlık altından veriler toplanmıştır. Oluşturulan derlemler ile eğitilen Word2Vec ve Glove modellerinin başarıları kıyaslanırken, konu odaklı çalışma ile genel çalışma arasındaki başarı farkı izlenmiştir. Alınan sonuçlar, sağlıkla ilişkili oluşturulan derlemden eğitilen modellerin daha başarılı semantik eşleşmeler yaptığını göstermektedir.

Eş anlamlı sözcüklerin yanlış kullanımını ve eş anlamı bulunamayan ifadelerle yönelik yakın anlamlı sözcükleri tespit etmek amacıyla Yan [61] ve arkadaşları tarafından bir çalışma yapılmıştır. Yapılan çalışmada İngilizce kelimeler için Reuters haberlerinden oluşan bir derlem ile sağlık verilerinden oluşan ikinci farklı bir derlem kullanılmıştır. Bu derlemler üzerinden eğitilen Word2Vec modeli ile, eş anlamlılar sözlüğüne ihtiyacı gidermek veya tam aksine eş anlamı aranan sözcüklere alternatifler sunmak hedeflenmiştir. İki ayrı çalışma için de başarılı sonuçlar elde edildiği çalışma sonunda vurgulanmıştır.

Sitikhu [62] ve arkadaşları tarafından yapılan çalışmada metin benzerliğine yönelik semantik çalışmaların kıyaslaması yapılmıştır. Bu çalışma kapsamında 3 ayrı güncel

yöntem karşılaştırılmıştır. Bunlardan ilki TF-IDF yardımcı Kosinüs Algoritması, ikincisi kelime temsili yardımcı Kosinüs Algoritması, üçüncüsü ise kelime temsili yardımıyla Soft Kosinüs Algoritmasıdır. Bu hibrit yöntemler içerisinde kısa metinler için en etkili sonucu TF-IDF destekli Kosinüs Algoritması vermiştir. Metinler uzadıkça kelime temsili yöntemlerinin başarıya etkisi artmaktadır.

Yapılan çalışmada Türkçe metinler arasında doğrudan benzerlik ile anlamsal benzerliği birlikte kontrol eden hibrit bir model üzerine odaklanılmıştır. Literatürde Türkçe odaklı çalışmaların ve çalışmalara altyapı hazırlayacak derlem eksikliğinden dolayı Bilgi Güvenliği alanında Türkçe metinlerden oluşan bir derlem hazırlanacaktır. Alan özelinde çalışma gereksinimi, geçmiş çalışmalarda görülen derlemin büyüklüğünün önemine binaen ortaya çıkmaktadır. Bir başlık üzerinde odaklanan metinlerin kullanılması ile birlikte o alana dair geniş bir derlem oluşturup Word2Vec metodu ile daha başarılı sonuçların alınabileceği öngörülmüştür.

Hazırlanan çalışmada, doğrudan benzerlik tespitine yönelik TF-IDF destekli Kosinüs Benzerliği yönteminin kullanılması kararlaştırılmıştır. Önceki çalışmaların büyük çoğunluğunun sonuçlarından yola çıkarak Kosinüs teoreminin cümle bazlı benzerlik ölçümünde başarılı sonuçlar verdiği gözlemlenmiştir. Ayrıca benzer çalışmalarda, TF-IDF ile desteklendiğinde Kosinüs teoremi ile başarımın arttığı gözlemine dayanarak önerilen yöntemde TF-IDF yaklaşımı da dahil edilecektir. Bu kapsamda vektörel gösterimle anlamsal ve kavramsal yakınlıkları, Kosinüs Benzerliği ve TF-IDF yaklaşımıyla da yazım benzerliklerinin birlikte denetleneceği yeni bir yöntem motivasyonu oluşmuştur. Yapılan çalışmanın öncelikli hedefi, yapısal ve semantik benzerliği birlikte kullanarak yeni bir benzerlik yöntemi sunmaktır. Başarılı olunması halinde; eş anlamlı veya yakın anlamlı kelimelerin kullanımı, kısaltmaların veya farklı dilde terimlerin kullanımı gibi faktörler benzerlik oranı bulmada yanıltıcı olmaktan çıkacaktır.

2. MATERYAL VE METOT İLE BULGULAR

2.1. Bilgi Güvenliđi Alanında Türkçe Derlemin Oluřturulması

Yapılan alıřma Türke metinleri hedef aldıđı iin Türke ieriđe sahip geniř bir veri kumesine ihtiya duyulmuřtur. Bu kapsamda Bilgi Güvenliđi konusu altında yazılan akademik alıřmalar derlenerek yeni bir Türke derlem oluřturulmuřtur. Bir alanda geecek teknik terim ve literatür dilinin bilgisayar tarafından iyi öğrenilmesi iin derlem ierisinde bu ifadelerin sık bulunması önemlidir. Bir terim ne kadar sık kullanılırsa, o kelimeye yönelik oluřturulacak vektörün temsili de daha dođru olacaktır. Dođru temsil edilen vektörler ise DDİ alıřmalarında makinenin o dili algılama becerisini üst seviyeye ıkaracaktır. Bu nedenle konulardan bağımsız genel bir derlem hazırlamak yerine sadece bir alandan seilen datalar ile hazırlamak daha uygun görölmüřtür.

Literatürde daha önceden bu alanda hazırlanmıř bir derlem ile karřılařılmadıđı ve bu konuda fazlaca teknik ağırlıklı terim bulunduđu göz önüne alınarak Bilgi Güvenliđi alanında bir derlem hazırlanılmasına karar verilmiřtir. Hazırlanan derlemin tamamı Türke yayınlardan oluřmaktadır. Fakat bu yayınların ierisinde bulunan İngilizce anahtar kelimeler derlemde ıkarılmamıřtır. İngilizce anahtar kelimelerin ođunlukla sık kullanılan terimlerden oluřturulduđu ve Türkeleřtirilmeden de kullanıldıđı varsayımı ile bu kelimeler de vektörleřtirilmiřtir.

Konu kapsamında geen terim sayısının fazla olması, bu kelimelerin makine tarafından algılanmasını zorlařtıracak bir faktördür. ünkü bu terimler genellikle Türke ilk anlamları ile aynı anlamda kullanılmamaktadır. Anlamsal benzerlik alıřmalarında, sık kullanılmayan kelimelerin makine tarafından anlařılabilmesi oldukça zordur. Örnek olarak tıbbi literatür ile öğrenme yapan makineye mühendislik dataları ile sunulacak test verisinden elde edilecek bařarı, aynı alanda verilen test örneđine göre daha düşük sonuçlar verecektir [63]. Önerilen yöntemin eř anlamlı ifadelerde veya terim kullanılan ifadelerde de benzerlik tespiti yapabildiđini ölçmek adına terim sayısının fazla olduđu bir alan tercih edilmiřtir. Bu konu özelinde bařarı oranının kayda deđer olması durumunda konu bazlı diđer alıřmalarda veya ok daha geniř derlem ile konulardan bağımsız genel benzerlik alıřmalarında aynı yöntemin kullanımı mümkün görölecektir.

Oluřturulan derlem alıřmada hazırlanacak Word2Vec modelinin kullanılması iin gereklilik göstermektedir. Derlem ierisinde geen tüm kelimeler, bir sonraki bölümde

anlatılacak ön işleme adımlarından sonra vektörel bir gösterime sahip olacaktır. Modelin başarılı ilişkiler kurabilmesi için Bölüm 1.4 içerisinde bahsedilen şartlara uygun ve tutarlı belgeler kullanılmıştır. Text haline dönüştürülen belgelerin birleştirilmesi sonucunda oluşturulan derlemin büyüklüğü aşağıda maddeler halinde gösterilmiştir.

- 56888 adet cümle
- 692004 adet kelime
- 106919 adet farklı token

Her bir token bir kelimeye işaret etmekte fakat Türkçe kelimelerin sondan eklemeli dil yapısı göz önüne alınmamaktadır. Bu nedenle 106919 adet farklı kelimenin bulunduğu sonucuna varılamaz. Bu kelimelerin sadece köklerinin hesaba katılması için DDİ işlemlerinden kök çözümleme ve gövdeleme tekniklerine başvurulacaktır. Böylelikle eklerinden arındırılmış farklı kelime sayısı önışleme adımlarından sonra tekrar değerlendirilecektir.

2.2. Veri Önışleme Adımları

Dokümanlar üzerinde yapılan çalışmaların başarımını doğru ölçebilmek için veri kalitesinin üst seviyede olması önemlidir. Üzerinde DDİ çalışmaları yapılacak veri kümesinin doğrudan işlemlere tabi tutulması oldukça yanıltıcı sonuçlar doğurabilmektedir. Bu nedenle verilerden elde edilecek sonuçların anlamlı olabilmesi için veri önışleme faslı DDİ çalışmalarında ve veri madenciliğinde büyük önem taşımaktadır [64].

Önışleme adımlarına ihtiyacı zaruri hale getiren başlıca sebepler aşağıda maddeler halinde sıralanmaktadır.

- Yazım yanlışları
- Nümerik verilerin çokluğu
- Gereksiz görülen sözcükler (stopwords)
- Metin aralarındaki boşluklar, görseller veya ikonlar
- Uygun karakter kümesine göre (utf-8 gibi) çevrilemeyen ifadeler

Yukarıdaki maddelere ilave olarak çalışmanın gereklerine göre kelimelerin köklere ayrılması, farklı dildeki sözcüklerden arındırılması, kısaltmalardan arındırılması gibi yöntemler de önışleme adımları arasında bulunur.

Yapılan çalışmada hazırlanan derlem üzerindeki önışleme adımları alt başlıklar halinde açıklanmıştır. Bu işlemler sonrasında veri kümesindeki tüm kelimeler için kelime temsili yöntemleri kullanılacaktır.

2.2.1. Verinin Sayısal İfadeler ve Noktalamalardan Arındırılması

Yapılan çalışma Türkçe metinlerdeki yapısal ve anlamsal benzerliklerin ölçülmesini hedeflemektedir. Bu hedef doğrultusunda sadece sözcükler üzerinde işlemler yapılabilmesi için veri kümesi sayısal ifadelerden arındırılmıştır. Bu noktada cümleler arasında geçen sayısal istatistikler veya sayısal açıklamaların cümleden çıkarılması bazı anlamsal değişikliklere neden olabilmektedir. Fakat bu şekilde değerli görülebilecek veri kaybı, anlamı bozmayan sayısal ifade kullanımına göre oldukça düşük orandadır. Bu nedenle nümerik ifadelerin çıkarılması nedeniyle oluşabilecek sapmalar göz ardı edilmektedir.

Veri kümesi oluşturulurken kullanılan makale ve diğer bilimsel çalışmalar haliyle birçok noktalama işareti barındırmaktadır. Kelimelerin işlenmesi sürecinde her bir karakter, makine dilinde karşılığı olan bir ASCII değeri ile bilgisayar tarafından algılanmaktadır. Haliyle bilgisayarın bir noktalama işaretini ve boşluk karakterini ilk okumada bir harften ayırması mümkün olmamaktadır. Bu durum, bazı kelimelerin bitişik diğer noktalama işaretleri ile birlikte bütüncül algılanması problemini doğurur. Böyle problemlerle karşılaşmamak için diğer DDİ çalışmalarının birçoğunda olduğu gibi veri kümesi noktalama işaretleri ve gereksiz boşluklardan arındırılmıştır.

2.2.2. Verinin Tamamen Küçük Harflere Dönüştürülmesi

Veri kümesindeki bütün kelimelerin vektörler ile ifade edilebilmesi için tüm kelimelerin başarılı şekilde çıkarılması gerekmektedir. Çıkarılan her bir kelime için farklı bir vektör oluşacaktır. Bu noktada mükerrer girdilere dikkat edilmelidir. Cümle başlarında veya başlıklarda geçen kelimelerin büyük harf ile başlaması bahsedilen problemin en temel sebebi olarak görülmektedir. Bilgisayar tarafından aynı harfin büyük ve küçük hali farklı karakterler olarak algılandığı için aynı kelime iki farklı kelime olarak algılanmaktadır. Bu nedenlerden dolayı tüm metin küçük harflerden oluşacak şekilde tekrar güncellenmiştir.

2.2.3. Verinin Gereksiz Kelimelerden Arındırılması

Metin işleme çalışmalarında kelimelerin veri kümesi içerisinde geçme frekansları çoğunlukla kilit seviyede önemlidir. Elde edilecek istatistiksel sonuçları veya çıkarımları doğrudan etkilemektedir. Fakat çalışılan dilin yapısında anlamsal karşılık bulunmayan veya gereksiz görülen kelimeler bulunabilmektedir. Bu kelimelere örnek olarak edatlar

veya bağlaçlar ele alınabilir. Bu gibi anlam bütünlüğünü tamamlayıcı özellikte olup tek başına anlam barındırmayan kelimeler DDİ çalışmalarında ‘stopwords’ olarak adlandırılmıştır [65]. Anlamli verilerden oluşturulan geniş veri kümeleri incelendiğinde en çok kullanılan kelimeler arasında bu kelimelerin üst sıralarda olduğu gözlemlenir [66].

Geçmiş çalışmalarda birçok dil için hazırlanan ‘stopwords’ listeleri hazır listeler halinde dil işleme çalışmalarında kullanılabilmektedir. Bu çalışmada Türkçe metinler üzerinde işlemler yapılacağı için Tablo 2.1’de listelenen kelime grubu veri kümesinden çıkarılmıştır.



Tablo 2.1. Türkçe stopwords listesi

acaba	böyle	gene	kez	onlar	şeyde
altı	böylece	gibi	kırk	onlara	şeyden
altmış	bu	hâlâ	ki	onlarda	şeye
ama	bu arada	hangi	kim	onlardan	şeyi
ancak	bu yüzden	hangisi	kimde	onları	şeyler
artık	buarada	hani	kimden	onların	şimdi
asla	buna	hatta	kime	onların	şöyle
aslında	bunda	hem	kimi	onu	şu
az	bundan	henüz	kimin	onun	şuna
bana	bunu	hep	kimisi	orada	şunda
bazen	bunun	hepsi	kimse	otuz	şundan
bazı	burada	hepsinde	madem	oysa	şunlar
bazıları	burda	hepsinden	mı	oysaki	şunu
bazısı	buyüzden	hepsine	mı	öbürü	şunun
belki	bütün	hepsini	mi	ön	tabi
ben	çoğu	hepsinin	milyar	önce	tamam
bende	çoğuna	her	milyon mu	ötürü	trilyon
benden	çoğunu	her biri	mu	öyle	tüm
beni	çok	herbiri herkes	mü	rağmen	tümü
benim	çünkü	herkese	mü	sana	üç
beş	da daha	herkesi	nasıl	sanki	üzere
bile	dahi	hiç	ne	sekiz	var
bin	de	hiç kimse	ne kadar	seksen	ve
bir	defa	hiçbir	ne zaman	sen	veya
biri	değil	hiçbiri	nekadar	sana	veyahut
birçoğu	demek	hiçbirine	nezaman	sekiz	ya
birçok	diğer	hiçbirini	neden	sen	ya da
birçokları	diğeri	hiçkimse	nedir	sende	yada
biri	diğerleri diye	için	nerde	senden	yani
birisi	doksan	içinde	nerede	seni	yedi
birkaç birkaçı	dokuz	iki	nereden	senin	yerine
birkez	dolayı	ile	nereye	siz	yetmiş
birşey	dört	ise	nesi	sizde	yine
birşeyi	elbette	işte	neyse	sizden	yirmi
biz	elli	kaç	niçin	size	yoksa
bizde	en	kadar	niye	sizi	yüz
bizden	fakat	katrilyon	on	sizin	zaten
bize	falan	kendi	ona	son	zira
bizi	felan	kendine	onda	sonra	
bizim	filan	kendini	ondan	şayet	
				şey	

Çalışmada kullanılacak Word2Vec yönteminin algoritması gereği; bir kelimenin derlemde bulunma sıklığı, yanındaki diğer kelimelerin vektörlerinin oluşturulmasında doğrudan kullanılmaktadır. Gereksiz kelimelerin cümlelerden ayrılması durumunda bu kelimeler oldukça yanıltıcı sonuçlara sebep olacaktır.

Key	Type	Size	Value
ve	int	1	21960
bir	int	1	10839
bu	int	1	8541
olarak	int	1	4901
ile	int	1	4850
için	int	1	4590
siber	int	1	3584
bilgi	int	1	3555
güvenlik	int	1	2897
da	int	1	2831
olan	int	1	2266
de	int	1	2253
şifre	int	1	2126
gibi	int	1	2019
daha	int	1	1836
ise	int	1	1605
tarafından	int	1	1587
saldırı	int	1	1567
veya	int	1	1557
olduğu	int	1	1543

Şekil 2.1. Bilgi güvenliği derleminde en sık geçen kelimeler

Şekil 2.1’de derlemde en sık geçen 20 kelimenin listesi görülmektedir. Görsel incelendiğinde derlemde en sık geçen kelimelerin çoğunlukla gereksiz kelimelerden (stopwords) oluştuğu görülmektedir. Liste içerisinde çerçeveye alınmış kelimeler haricindeki 14 kelime bağlaç veya edatlardan oluşmaktadır. Bu kelimelerin veri kümesinden çıkarılması sonrasında sık geçen kelimeler tekrar hesaplanmış ve Şekil 2.2’de gösterilmiştir.

Key	Type	Size	Value
siber	int	1	3584
bilgi	int	1	3555
güvenlik	int	1	2897
şifre	int	1	2126
tarafından	int	1	1587
saldırı	int	1	1567
ilgili	int	1	1235
veri	int	1	1185
önemli	int	1	1133
şekilde	int	1	1121
internet	int	1	1092
ortaya	int	1	1083
tablo	int	1	1048
elde	int	1	960
bilişim	int	1	940
diğer	int	1	938
farklı	int	1	934
yönelik	int	1	920
sahip	int	1	909
yapılan	int	1	900

Şekil 2.2. Bilgi güvenliği derleminden stopwords’lerin silinmesi ile en sık geçen kelimeler

2.2.4. Verinin Yabancı Kelimelerden Arındırılması

Çalışma için hazırlanan derlemin önışleme adımlarına uğramadan önce yaklaşık %8’i İngilizce metinlerden oluşmaktadır. Yabancı dille yazılmış bu ifadelerin tamamını derlemden çıkarmak bir seçenek olarak değerlendirilebilir. Fakat bu yabancı ifadeler içerisinde Bilgi Güvenliği alanına özgü terimlerin bulunduğu gözlemlenmiştir. Çalışma konusunun yabancı terimler barındırması birçok alan için kaçınılmazdır. Bu terimlerin Türkçeye doğrudan çevrilememesi beraberinde bazı anlamsal sorunlar getirmektedir. Bu terimler için Türkçe karşılıklar belirlenmiş olsa dahi genel kullanım alışkanlıkları

nedeniyle orijinal terimin kullanılmasına sık rastlanılmaktadır. Aynı kavramın bir belgede Türkçe sözcüklerle, başka bir belgede yabancı sözcüklerle ifade edilmesi durumunda bu kelimelerin vektör değerleri yanıltıcı sonuçlar içerecektir.

Anlatılan sorunlardan ötürü derlemdeki tüm yabancı kelimeler çıkarılmamıştır. Ayıklama yöntemi olarak, anahtar kelimeler ve derlemde kelimenin kullanım sıklığı baz alınarak filtreleme yapılmıştır. Yabancı bir kelime veya kelime grubu bilimsel çalışmanın ‘anahtar kelimeler’ kısmında geçiyorsa atılmaması gereken bir terim olduğu varsayımı yapılmıştır. Aynı şekilde İngilizce özet kısımlarıyla birlikte sunulan ‘keywords’ bölümünde geçen kelimeler derlemde çıkarılmamıştır. Bu çalışmaların ardından derlem içerisinde en fazla geçen kelimeler gözden geçirilmiş ve ilk 500 kelime arasında bulunan yabancı kelimeler derlemde tutulmuştur.

2.3. Önerilen Yöntemin Açıklanması

Hazırlanan çalışma iki Türkçe metin arasındaki benzerliği oranlar ile ifade edecek şekilde tespit etmeyi hedeflemektedir. Yöntem olarak ise yapısal ve anlamsal benzerlik algoritmalarından faydalanılacaktır. Önerilen yöntemde, metinler öncelikle yapısal olarak karşılaştırılacak ve bire-bir benzerlikleri tespit edilecektir. Bu tespiti yaparken Kosinüs Benzerlik Algoritması kullanılacaktır. Yapısal benzerlik oranı belirli bir eşiğin (0.3) üzerinde çıktığı takdirde aynı metinler ilave olarak bir de anlamsal benzerlik testine tabi tutulacaktır. Önerilen yöntemde anlamsal ilişkilerin tespit edilebilmesi için Semantik Benzerlik Algoritmalarına alternatif olarak Word2Vec yöntemi kullanılacaktır.

Bilgi güvenliği alanında bilimsel metinlerden bir derlem oluşturulmuştur. Bu derlem gerekli ön işleme adımlarıyla DDİ çalışmalarına hazır hale getirilmiştir. Yapılan hazırlıktan sonra eldeki veri kümesi kullanılarak bir Word2Vec modeli geliştirilecek ve kelimelerin anlamsal vektörleri çıkarılacaktır.

Kosinüs benzerliği ölçülen iki ayrı metin için sonuç 0.3 ve üzeri çıkarsa bu metinler algoritmanın anlamsal yakınlık ölçümüne gönderilecektir. Bu noktada metinlerden bir tanesi orijinal haliyle tutulurken diğer metin cümlelerine ayrılmış bir liste olarak tutulacaktır. Ardından bu liste kelimelerine ayrılarak token seviyesinde parçalanmış hale gelir. Her bir kelime, daha önce eğitilen Word2Vec modeline girdi olarak sokulur ve bu kelimeye en yakın 5 kelime çıkarılır. Ardından esas kelime yerine, modelden gelen alternatif kelimeler koyularak her cümle için tekrar Kosinüs Benzerliği uygulanır. Bu işlem cümledeki tüm kelimeler için tekrarlanır. Elde edilen benzerlik oranları ilk orandan daha

yüksek çıkarsa yapısal benzerliğin yetersiz kaldığı ölçümde önerilen yöntem ile başarı artırılır. Şayet benzerlik oranları ilk ölçüme göre kayda değer bir artış göstermez veya daha düşük olursa, iki cümlenin anlamsal olarak da birbirinden uzak olduğu sonucuna varılır. Önerilen yönteme ait bir örnek ve işlem adımları aşağıda gösterilmektedir.

metin1 = Siber saldırı tespitine yönelik çalışmalar devam ediyor.

metin2 = Siber casusluk tespiti için araştırmalar devam ediyor.

1. Adım : İki metin Kosinüs Benzerlik Algoritması ile karşılaştırılır. Benzerlik oranı bulunur. Benzerlik Oranı 1 (BO1) hesaplanır.

$$Kos_{(m1,m2)} = BO1 = 0.4285714285714285$$

2. Adım : Elde edilen BO1 değeri 0.3'ten büyük olursa metinlerden biri seçilerek ikinci metinden farklı olan kelimeleri çıkarılır.

Çıkan kelimeler : saldırı – yönelik – çalışmalar

Bu kelimelerin Word2Vec modelinden sırayla benzerleri bulunur ve benzeri ile değiştirilerek metin1 güncellenir. Her bir kelime için, modelden 5 en yakın kelime önerisi alınır. Alınan kelimeler sırası ile cümle içerisinde denenerek, güncel her bir hali için yeni benzerlik oranı hesaplanır.

```
In [14]: model.wv.most_similar('saldırı',topn=5)
Out[14]:
[('saldırıları', 0.7828900814056396),
 ('saldırıları', 0.7550871968269348),
 ('saldırının', 0.7435573935508728),
 ('casusluk', 0.7153679132461548),
 ('saldırıya', 0.7146600484848022)]
```

Yeni metin1 = “Siber saldırılar tespitine yönelik çalışmalar devam ediyor.”

BO2 = 0.4285714285714285

Yeni metin1 = “Siber saldırıları tespitine yönelik çalışmalar devam ediyor.”

BO2 = 0.4285714285714285

Yeni metin1 = “Siber saldırının tespitine yönelik çalışmalar devam ediyor.”

BO2 = 0.4285714285714285

Yeni metin1 = “Siber casusluk tespitine yönelik çalışmalar devam ediyor.”

BO2 = 0.5714285714285713

Yeni metin1 = “Siber saldırıya tespitine yönelik çalışmalar devam ediyor.”

BO2 = 0.4285714285714285

Yukarıdaki işlemler sonucunda ‘saldırı’ kelimesinin yerine modelin önerdiği ‘casusluk’ kelimesi kullanılarak benzerlik oranı arttırılmıştır. Bu aşamadan sonra metin üstünde yapılacak işlemlerde ‘casusluk’ kelimesi kullanılacak şekilde metin güncellenir.

3. Adım = 2. Adımda yapılan işlem sırasıyla ‘yönelik’ kelimesi için tekrar edilecektir. Benzerlik oranının yükseldiği her adımda metin1 güncellenir.

```
In [26]: model.wv.most_similar('yönelik',topn=5)
Out[26]:
[('ilişkin', 0.7315554022789001),
 ('güvenliğine', 0.6359898447990417),
 ('neler', 0.6147449016571045),
 ('görüşleri', 0.6105387806892395),
 ('dönük', 0.60867840051651)]
```

Yeni metin1 = “Siber casusluk tespitine ilişkin çalışmalar devam ediyor.”

BO2 = 0.5714285714285713

Yeni metin1 = “Siber casusluk tespitine güvenliğine çalışmalar devam ediyor.”

BO2 = 0.5714285714285713

Yeni metin1 = “Siber casusluk tespitine neler çalışmalar devam ediyor.”

BO2 = 0.5714285714285713

Yeni metin1 = “Siber casusluk tespitine görüşleri çalışmalar devam ediyor.”

BO2 = 0.5714285714285713

Yeni metin1 = “Siber casusluk tespitine dönük çalışmalar devam ediyor.”

BO2 = 0.5714285714285713

4. Adım = 3. Adımda ‘yönelik’ kelimesi için önerilen alternatif kelimeler total benzerlik oranını yükseltmemiştir. Bu nedenle kelime olduğu gibi tutularak cümle güncellenmez. Bu aşamadan sonra ‘çalışmalar’ kelimesi için aynı işlemler tekrarlanacak ve sonuç benzerlik oranı elde edilecektir.

```
In [37]: model.wv.most_similar('çalışmalar',topn=5)
Out[37]:
[('araştırmalar', 0.8409428596496582),
 ('çalışmaların', 0.8376559615135193),
 ('yürütmüştür', 0.787592887878418),
 ('çalışmalara', 0.7831051349639893),
 ('çalışmaları', 0.7650936841964722)]
```

Yeni metin1 = “Siber casusluk tespitine yönelik araştırmalar devam ediyor.”

BO2 = 0.7142857142857141

Yeni metin1 = “Siber casusluk tespitine yönelik çalışmaların devam ediyor.”

BO2 = 0.5714285714285713

Yeni metin1 = “Siber casusluk tespitine yönelik yürütmüştür devam ediyor.”

BO2 = 0.5714285714285713

Yeni metin1 = “Siber casusluk tespitine yönelik çalışmalara devam ediyor.”

BO2 = 0.5714285714285713

Yeni metin1 = “Siber casusluk tespitine yönelik çalışmaları devam ediyor.”

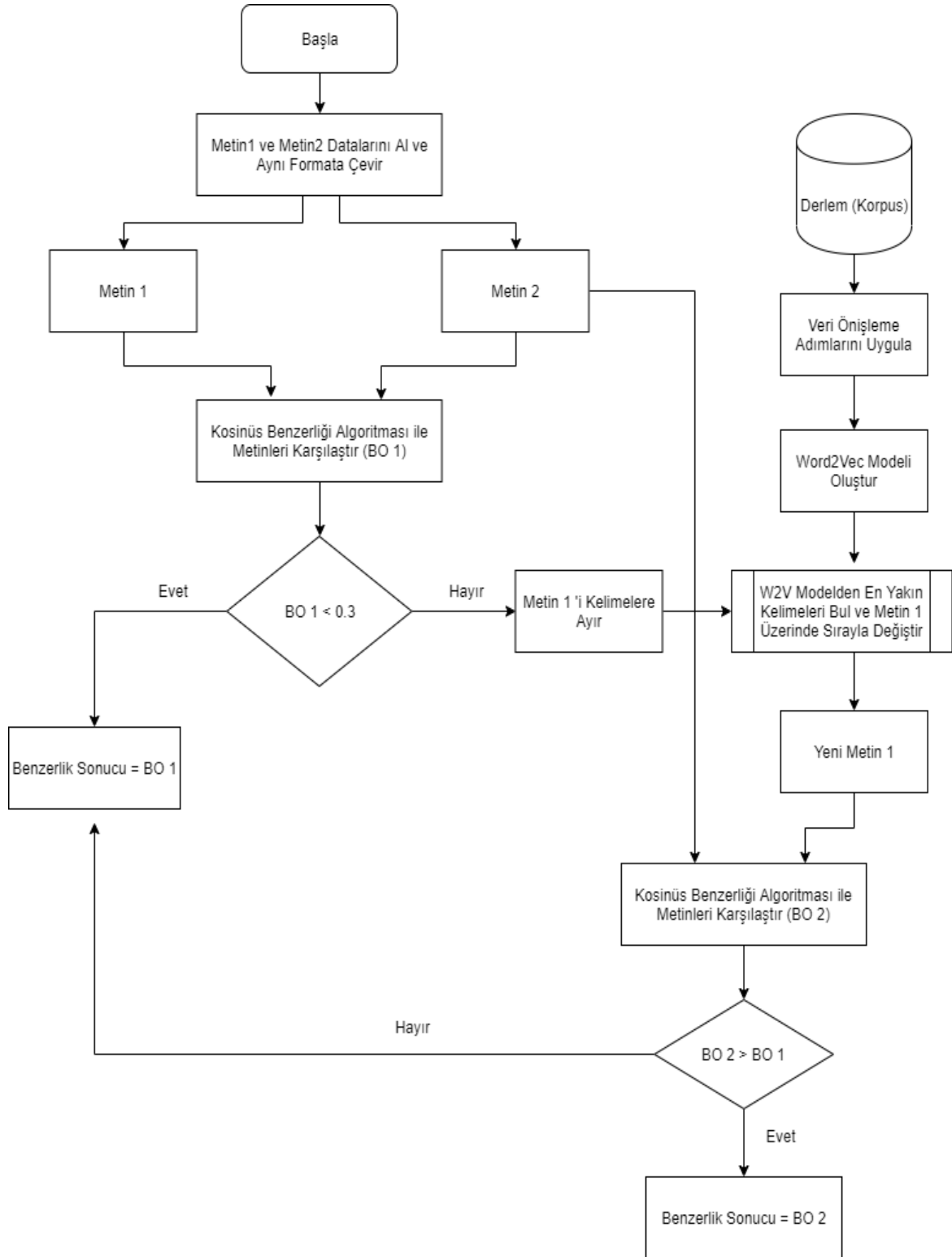
BO2 = 0.5714285714285713

5. Adım = Değiştirilecek kelime kalmadığı için işlemler sırasında bulunan en yüksek BO2 değeri olan **0.7142857142857141** nihai oran kabul edilir.

Örnekte yapılan işlem adımlarından sonra iki cümle arasındaki benzerlik oranı ciddi oranda yükselmiştir. Cümlelerin anlamsal yakınlıkları değerlendirildiğinde ilk değer olan BO1 değeri (0,43), BO2 (0.71) ile güncellenmiştir. Böylelikle yapısal ve anlamsal benzerlik hibritlenmiş, yapısal benzerlik oranına göre daha başarılı bir sonuç alınmıştır.

Elde edilen sonuçların başarılı olması için Word2Vec modelinin tutarlı sonuçlar vermesi oldukça önemlidir. Çalışmada kullanılan model Bilgi Güvenliği alanında hazırlandığı için, özellikle bu alana ait terimlerde yüksek başarı beklenmektedir. Örnek incelendiğinde ‘siber saldırı’ ve ‘siber casusluk’ yakın anlamlar olarak model tarafından önerilmiştir. Toplam benzerliğe etki etmese dahi önerilen yeni kelimelerin esas kelimeye yakınlığı, modelin tutarlılığını göstermektedir.

Geliştirilen yöntemin algoritması Şekil 2.3’de gösterilmektedir. Şekilde algoritmanın akış diyagramı adımlar halinde gösterilmiştir. Diyagram üzerinde kısaltılmış olarak gösterilen BO1, Word2Vec kullanılmadan önceki benzerlik oranını ifade eder. Yine kısaltılmış olarak gösterilen BO2 ise Word2Vec modeli ile desteklenmiş sonuçlarla elde edilen benzerlik oranını göstermektedir.



Şekil 2.3. Önerilen yöntemin algoritma akış şeması

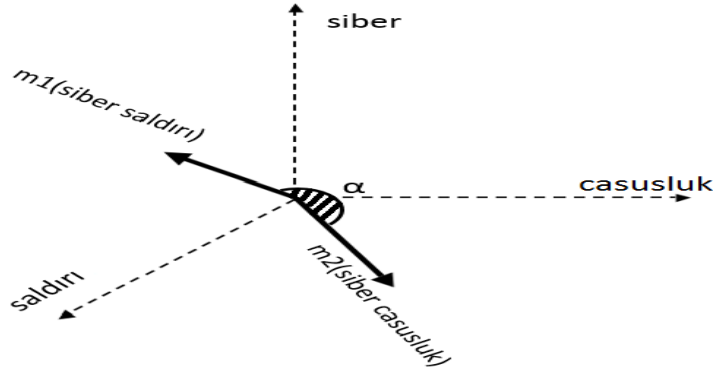
2.4. Kosinüs Benzerliğinin Veriye Uygulanması

Geliştirilen yöntemin eleme ve karar verme süreçlerinde Kosinüs Benzerliği Algoritmasından faydalanılmıştır. Algoritmaya bağlı kalınarak cümleler öncelikle kelimelere bölünür. Kelimelerin ise vektörler halinde gösterilebilmesi için Python kütüphanelerinden Counter kütüphanesi ve CountVectorizer modeli kullanılmıştır. Bu yöntemler sayesinde yazı içerikli metnin bitler seviyesinde matematiksel gösterimleri mümkün olmaktadır. Ardından her bir bit dizisi bir vektör olarak ele alınır ve nokta çarpımı ile aralarındaki uzaklıkları bulunur.

Çalışmada yapısal benzerliğin bulunması için Kosinüs Benzerliğinin tercih edilmesi iki temel sebebe dayanmaktadır. Bunlardan ilki Kosinüs Benzerliğinin algoritması gereği birbirinden farklı uzunluktaki metinler için tutarlı sonuçlar verebilmesidir. İkinci önemli faktör ise literatürde yapısal metin benzerliği için yapılan kıyaslamalarda Kosinüs Algoritmasının sıklıkla önerilmesidir [67].

Yapılan çalışmada Bölüm 2.3'te detaylıca anlatıldığı üzere en az iki kez Kosinüs Benzerliği yöntemi kullanılmaktadır. İlk etapta karşılaştırılması için verilen iki metin doğrudan benzerlik algoritmasına tabi tutulur. Word2Vec modelinden alınan destekle birlikte ölçülen iki benzerlik oranında yine Kosinüs Benzerliğine başvurulmuştur. İki farklı ölçüm metriği ile çalışmak durumunda oluşacak hesap tutarsızlıkları böylece engellenmiştir. Geliştirilen yöntemden bağımsız olarak başarıyı arttırıcı unsur olarak kullanılan Word2Vec algoritması da iç mekanizmasında Kosinüs Benzerliğine başvurmaktadır. Derlem içerisindeki kelimelerin anlamsal yakınlığı ölçülürken vektörler kullanılır. Bu vektörler üç boyutlu uzayda yerleştirilir ve ikili olarak aralarındaki açının kosinüsü hesaplanarak anlamsal yakınlık ölçülür.

Bölüm 2.3'te verilen örnek cümlelerden ilk üç kelime için kosinüs teoreminin gösterimi Şekil 2.4'deki gibi olmaktadır.



Şekil 2.4. Kosinüs Benzerlik Algoritması vektörel gösterimi

2.5. Word2Vec Modelinin Oluşturulması

Geliştirilen yöntemde yapısal benzerlik tespitlerini daha ileri seviyeye çıkarmak için Word2Vec kullanılmıştır. Çünkü metinlerin yapısal olarak yakınlıkları ölçülürken elde edilen sonuçlar, genellikle yeterli olmamaktadır. Metinlerin benzer anlamlara veya neredeyse aynı anlama gelecek şekilde alternatif kelimeler ile oluşturulmasına yapısal benzerlik algoritmaları çözüm üretememektedir. Özellikle eş anlamı ifadeler veya yakın anlamlı farklı kelimeler kullanıldığında cümlelerin anlamı değişmezken benzerlik oranı düşmektedir [68]. Yapılan çalışma ile bu problemi azaltmak ve anlamsal ilişkileri yakalamak için Word2Vec yöntemi ek bir katman olarak algoritmaya dahil edilmiştir.

Word2Vec modelinin oluşturulması için kullanılan derlem ‘Bilgi Güvenliği’ alanındaki dokümanlardan oluşturulmuştur. Bu nedenle modelin test edilmesi aşamasında alana özgü kavramlar kullanılacaktır. Eğitilen model sonucunda her bir kelimenin vektör şeklinde ifade edilmesi beklenir. Bu aşamada her bir vektör 100 elemandan oluşacak, yani vektörlerin uzunluğu 100 olacak şekilde kodlama yapılmıştır. Bu aşamadan sonra kelimenin vektör şeklinde sayısal görünümü Şekil 2.5’deki gibidir.

```
In [119]: model.wv['şifre']
Out[119]:
array([ 0.11466274, -0.3289032 , -0.02667928, -0.1204628 ,  0.19672088,
        0.1265576 ,  0.19944131,  0.24819596, -0.33593646, -0.041422 ,
        0.02554098,  0.27869484,  0.11043143,  0.28894627, -0.14747195,
       -0.04411989, -0.2932412 , -0.07456031, -0.14868188, -0.35199627,
        0.8047915 ,  0.23939052, -0.26240128,  0.33436835, -0.21632688,
       -0.08330528, -0.27190852,  0.61777145, -0.25148556, -0.24559163,
       -0.23763344,  0.3463697 ,  0.31782869, -0.63234055, -0.32512397,
       -0.13431883,  0.19596544,  0.2994541 , -0.5650457 , -0.04188585,
       -0.32449552, -0.4192738 , -0.20080253, -0.09355183, -0.5676993 ,
        0.0781574 ,  0.01369642,  0.41796476, -0.1117946 , -0.2085556 ,
       -0.5098273 ,  0.03353371, -0.3248955 ,  0.01600536, -0.15491037,
        0.3676037 , -0.3787443 ,  0.41756922,  0.54818326,  0.11422514,
       -0.01386378,  0.65000093,  0.29061753,  0.20892769, -0.16504914,
       -0.08766646, -0.28906742, -0.4231175 , -0.26927796, -0.09960197,
       -0.25727594, -0.04855007, -0.09283647,  0.08646671, -0.0734171 ,
        0.12140502, -0.38843086, -0.42501917, -0.04414474, -0.06131436,
        0.02616532, -0.7195856 , -0.479443 , -0.28602773,  0.10686725,
        0.43381146,  0.11832491,  0.33256912,  0.00674286,  0.09746746,
       -0.64582676,  0.1646138 ,  0.22128555, -0.00118487, -0.21379058,
       -0.01523906, -0.2675745 , -0.16139947, -0.09985736, -0.35873896],
      dtype=float32)
```

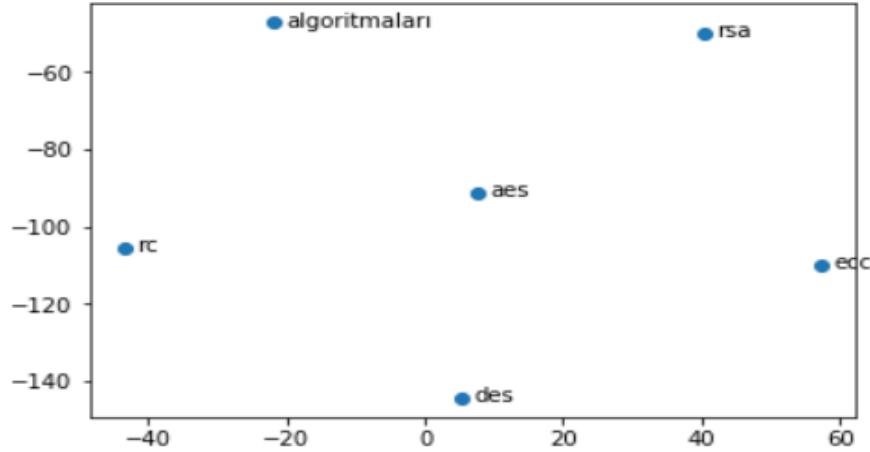
Şekil 2.5. Kelime vektörlerinin gösterimi

Bölüm 1.4.4’te detaylıca anlatıldığı üzere Word2Vec modeli oluşturulurken pencere boyutu belirlenir. Belirlenen boyut, kelimenin sağında ve solundaki kaç adet kelimeyle ilişkilendirileceğini belirler [69]. Çalışmada kullanılan modelde pencere boyutu 5 olarak belirlenmiştir. Böylelikle ortadaki kelimenin sağından ve solundan 5 adet kelime dikkate alınacaktır.

Oluşturulan model 800.000 adetten daha fazla kelime üzerinde eğitileceği için hatalı girdilere dikkat etmek sonuçlar açısından önemlidir. Yazım yanlışlarının modeli etkilememesi için derlem içerisinde en az 5 adet geçen kelimeler eğitime katılacaktır. Böylelikle bir kelime hatalı yazılmış olsa dahi göz önüne alınmamış olur. 5 adetten az sayıda geçen doğru kelimelerin ise çok az sıklıkta kullanıldığı veya çok istisnai kelimeler olduğu varsayımı yapılır. Bu kelimeler de eğitim setinde kullanılmamıştır.

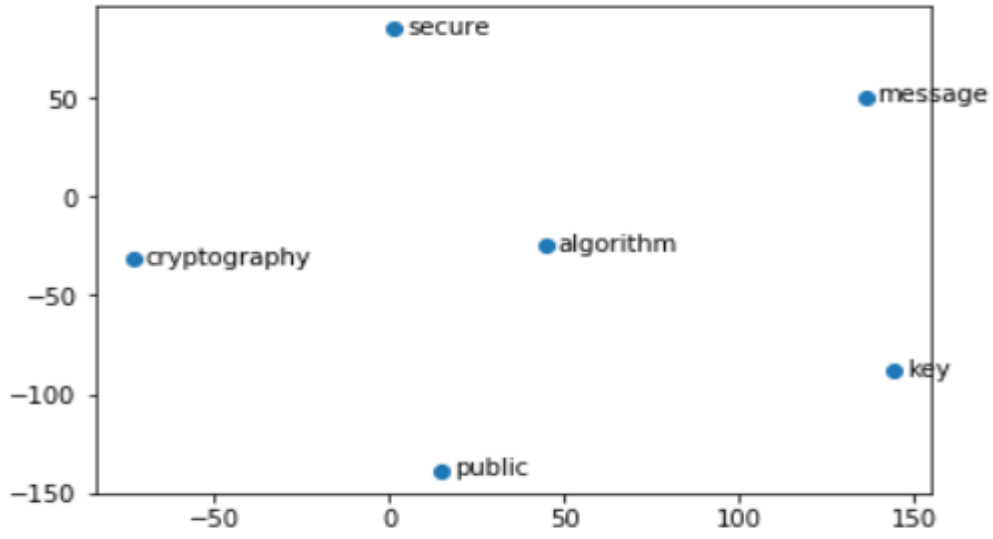
Word2Vec modelinin eğitilmesi için kullanılan iki yöntem bulunmaktadır. Bu yöntemler CBOW ve Skip-gram algoritmalarıdır. Yapılan çalışmada iki ayrı algoritma da denenmiş ve Skip-gram ile derlem üzerinde daha başarılı sonuçlar alındığı gözlemlenmiştir. Bu nedenle Skip-gram algoritması ile eğitim tamamlanmıştır. Eğitilen modelin sonuçları incelendiğinde anlam ilişkilerinin iyi seviyede yakalandığı gözlenmiştir. Modelin ürettiği sonuçlardan bazıları görselleriyle birlikte paylaşılmıştır.

Şekil 2.6'daki görsel, bir şifreleme algoritması olan 'RSA' algoritması için yakın kelimeleri göstermektedir. Eğitilen modelden alınan sonuçlar, RSA kelimesi için yakın sayılabilecek tutarlı sonuçlar vermiştir. RSA gibi şifreleme algoritmaları olan AES, DES, ECC gibi örnekler bulunmuştur. Ayrıca RSA'nın bir algoritma olduğu düşünülürse, 'algoritmalar' kelimesi modelin anlamlı ilişkiler kurduğunu yansıtmaktadır.



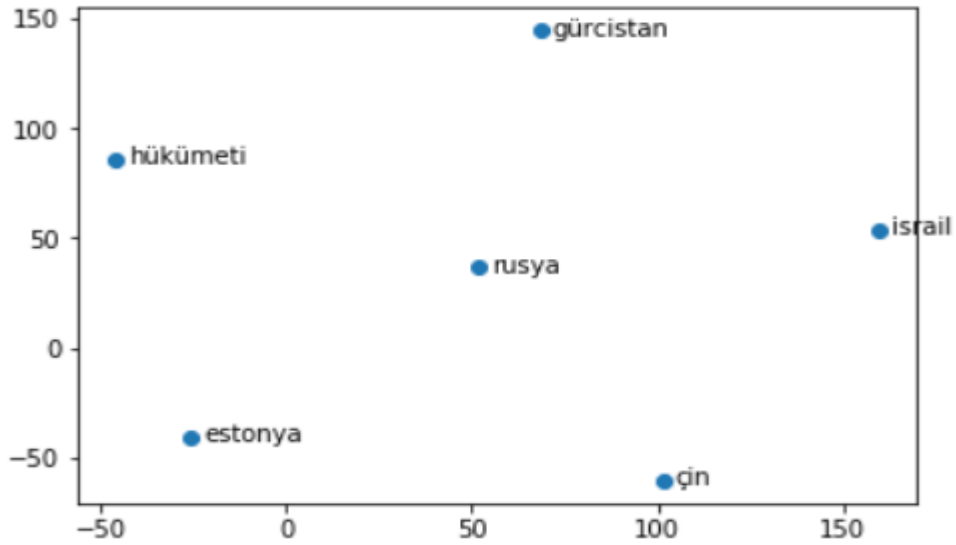
Şekil 2.6. RSA kelimesi için Word2Vec ile bulunan yakın kelimeler

Şekil 2.7'deki görsel, Türkçe karşılığı anahtar olan 'key' kelimesi için yakın kelimeleri göstermektedir. Hazırlanan derlemde, İngilizce kelimelere nadir de olsa yer verildiği Bölüm 2.1'de anlatılmıştır. Derlemdeki İngilizce kelime oranı çok az olmasına rağmen model anlamlı ilişkiler kurabilmiştir. Bilgi güvenliğinde sıklıkla kullanılan 'key' tabiri bir terim olarak kabul edilmektedir. Kriptografide şifreli içeriği çözmek için kullanılan anahtar tipleri incelendiğinde, 'private' ve 'public' şeklinde iki sınıf öne çıkar. Gizli veya açık anahtar tabirlerine karşılık, 'private' ve 'public' terimleri kullanılır. Eğitilen modelin önerdiği kelimelere bakılırsa bu anlamsal bağlantı yakalanmıştır.



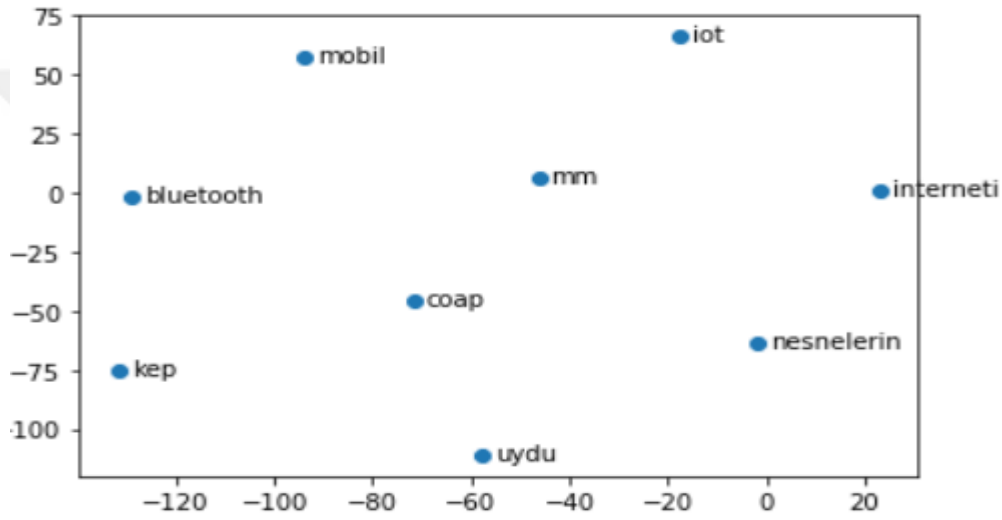
Şekil 2.7. Key kelimesi için Word2Vec ile bulunan yakın kelimeler

Bir diğer örnekte ise Şekil 2.8’de gösterildiği gibi modelden ‘Rusya’ kelimesi için yakın anlamlı kelimeler istenmiştir. Rusya’nın bir ülke olduğu model tarafından algılanmış ve benzeri ülkeler sıralanmıştır. Bu anlam ilişkisi önceki örneklerden farklı olarak, eş anlam veya karşıt anlama odaklanmayıp ortak bir sınıfa ait olma özelliğini kullanmıştır. Böylelikle, bir metin içerisinde kullanılan özel isme karşılık farklı bir özel ismin kullanılması gibi küçük değişimleri saptama imkanı doğmaktadır.



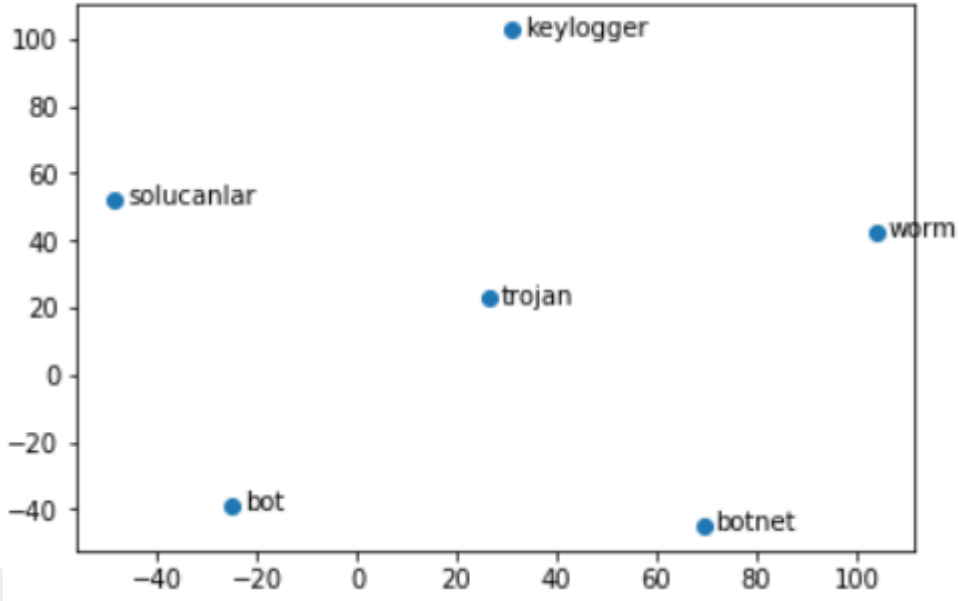
Şekil 2.8. Rusya kelimesi için Word2Vec ile bulunan yakın kelimeler

Şekil 2.9’da ‘IoT’ kelimesi ile ilişkili kelimeler gösterilmiştir. Iot, nesnelerin interneti olarak kullanılan bir kısaltmadır. Bu terim uzun şekliyle ‘Internet of Things’ şeklinde kullanılabilir fakat literatürde çoğunlukla ‘IoT’ olarak geçmektedir. Eğitilen modelin önerilerinde ise terimin Türkçe karşılığı olan ‘nesnelerin’ ‘interneti’ kelimeleri bulunmaktadır. Ayrıca makineler arası iletişimi gösteren ‘M2M’ terimi sayılardan arındırıldığı için ‘mm’ olarak gösterilmiştir. Önerilen diğer kelimelere de bakarak, alana özgü terimlerin yakalanması ve anlam ilişkilerinin kurulmasında modelin başarılı sonuçlar verdiği görülmektedir.



Şekil 2.9. IoT kelimesi için Word2Vec ile bulunan yakın kelimeler

Şekil 2.10’da ‘botnet’ terimi için önerilen yakın anlamlı kelimeler gösterilmiştir. Çoklu bilgisayar kullanımı ile yapılan siber saldırılarda daha önceden sızılmış bot (zombi) bilgisayarlar kullanılmaktadır. Bu saldırılarda kullanılacak zombi bilgisayarlar Botnet denilmektedir [70]. Örnekteki gibi bazı terimlerin tam olarak Türkçe karşılıkları bulunmamaktadır. Eğitilen modelden alınan sonuçlar ise bu terimin anlamsal olarak algılanabildiğini gösterir. Botnet oluşturabilmek için saldırganların çoğunluklu olarak kullandığı teknikler model tarafından önerilmiştir. Bu teknikler keylogger, solucan (worm), truva atı (trojen) gibi aracı sızma yöntemleridir. Ayrıca botnetin bot bilgisayarlardan oluştuğu bilgisi modelin önerdiği kelimelerin tutarlılığını göstermektedir.



Şekil 2.10. Botnet kelimesi için Word2Vec ile bulunan yakın kelimeler

Örneklerde gösterildiği gibi eğitilen model eğitim alanı üzerinde başarılı sonuçlar vermiştir. Bu süreçte modelin eğitilmesi için Python kütüphanelerinden Gensim, şekillerle gösterilmesi için ise TSNE ve matplotlib kütüphanelerinden faydalanılmıştır. Geliştirme ortamı olarak, Anaconda Navigator ile sunulan Spyder ve Jupyter geliştirme araçları kullanılmıştır.

2.6. Kosinüs Benzerliğinin Word2Vec Modeli ile Desteklenmesi

Önerilen yöntem anlamsal ve yapısal benzerliğin hibritlenmesi üzerine kurgulanmış ve deneyler bu doğrultuda yapılmıştır. Önceki başlıklarda açıklandığı üzere yapısal benzerlik yöntemi olarak Kosinüs Benzerliği Algoritması kullanılmıştır. Bu algorithmadan alınan sonuçların daha optimize hale gelmesi için Word2Vec modelinden faydalanılmıştır. Yöntemde Word2Vec kullanımının başlıca nedenleri maddeler halinde sıralanmıştır.

- Eş anlamlı ifadeler kullanılması ile metin benzerliği oranının manipüle edilmesi hatalı sonuçlar alınmasına neden olur.
- Eş anlamlı olmayan fakat yakın anlamlı (özel isimler, örneklemelerde kullanılan ortak sınıf elemanları gibi) olan sözcüklerin cümle içerisinde tercih edilmesi ile benzerlik oranı düşer. Böylelikle olması gerekenden daha düşük benzerlik bulunur.
- Alana özgü terimlerin farklı dillerle ifadesinde karşılaşılan benzerlik oranının düşmesi hatalı ölçüm oranlarına neden olur.

- Tanımlayıcı ifadelerin veya terimlerin kısaltmalar halinde farklı bir kelimeyle gösterimi, benzerlik oranının olduğundan daha düşük çıkmasına neden olur.
- Kelimelere gelen ekler sayesinde değişen anlamın, yapısal benzerlik algoritmaları ile algılanması zorlaşır. Bu zorluk nedeniyle olduğundan daha yüksek benzerlik oranlarının çıkması yanıltıcı sonuçlar doğurur.
- Yazımı aynı veya çok yakın ifadelerin kullanılması, anlamsal olarak birbirinden uzak metinlerin yüksek benzerlikte çıkmasına neden olur.

Yukarıdaki maddeler metin benzerliği tespitinde en sık karşılan problemler içerisinde. Bu sorunların sonuca etkisini minimuma çekmek için Word2Vec modeli kullanılmıştır.

Önerilen yöntemde algoritma iki adımda işlemektedir. Adımların ilkinde metinler kosinüs benzerlik algoritması ile kıyaslanır. Bu işlem sonucunda 0.3 benzerlik barajını aşan metinler için Word2Vec modelinden alınan alternatif kelimelere başvurulur. Modelden alınan kelimeler metindeki alternatifi yerine kullanılır ve benzerlik oranı tekrar hesaplanır. Hesaplamanın ardından oranın yükselmesi, anlamsal benzerliğin ilk hesaptan daha yüksek seviyede olduğunu göstermektedir. İki algoritmanın birleştirilmesine yönelik açıklama Bölüm 2.3'te detaylıca yapılmıştır.

2.7. Önerilen Yöntem ile Deneysel Çalışmalar

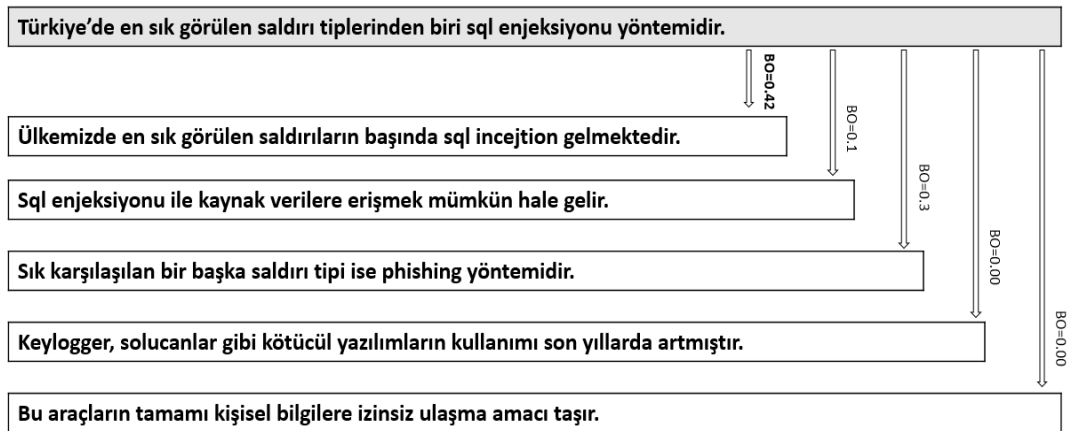
2.7.1. Bilgi Güvenliği Verileri ile Deney 1

Önerilen yöntemi test edebilmek için Bilgi Güvenliği alanına dahil olacak bir metin kullanılmıştır. Bu metne yakın anlamlı ikinci bir metin türetilmiştir. Karşılaştırma yapılacak metinler Tablo 2.2'de gösterilmektedir. Ardından bu metinler ile yapılacak işlemler ve elde edilen sonuçlar açıklanmaktadır.

Tablo 2.2. Deney 1 için seçilen metinler

metin 1	Türkiye’de en sık görülen saldırı tiplerinden biri sql enjeksiyonu yöntemidir. Bu yöntem ile sunucudaki verilere ulaşılmaktadır. Sık görülen bir diğer saldırı tipi ise ortalama yöntemidir. Worm, trojan gibi zararlı uygulamaların kullanımı da son yıllarda artmaktadır. Bu uygulamaların tümü kişisel verilere erişme amacıyla kullanılır.
metin 2	Ülkemizde en sık görülen saldırıların başında sql injection gelmektedir. Sql enjeksiyonu ile kaynak verilere erişmek mümkün hale gelir. Sık karşılaşılan bir başka saldırı tipi ise phishing yöntemidir. Keylogger, solucanlar gibi kötücül yazılımların kullanımı son yıllarda artmıştır. Bu araçların tamamı hassas bilgilere izinsiz ulaşma amacıyla kullanılır.

Tabloda gösterilen metinlerin karşılaştırılması için öncelikle cümleler haline getirilmesi gerekmektedir. Ardından her bir cümle pivot cümle olarak seçilecek ve diğer cümleler ile sırayla karşılaştırılacaktır. Elde edilen benzerlik oranları arasında en yüksek olanı değerlendirmeye alınır. Bu oran 0.3 üzeri ise önerilen yöntemin ikinci aşamasına geçilecek ve Word2Vec modeli kullanılacaktır. Çalışmanın ilk adımı metin1’in ilk cümlesi ile başlamaktadır. Bu cümle için yapılan benzerlik bulma adımları Şekil 2.11’de adım adım gösterilmiştir.



Şekil 2.11. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 1)

Görsel incelendiğinde, metin1’den alınan pivot cümle ile en yüksek benzerlik metin2’nin ilk cümlesi arasında görülmektedir. Bu oran 0.421 olup, 0.3 üzerinde olduğu için Word2Vec modelinden cümlede geçen kelimelere yakın diğer kelimeler alınacaktır. Bu safhada, iki cümlede ortak geçen kelimeler atlanır ve algoritma farklı kelimelerin birbirine benzer olma ihtimalleri üzerine odaklanır. Bu işlem basmağı Şekil 2.12’de adımları ile birlikte gösterilmektedir.

Türkiye’de	en	sık	görülen	saldırı	tiplerinden	biri	sql	enjeksiyonu	yöntemidir
ülkümüzde (BO = 0.527)	Word2Vec modelinin önerileri				x	birisi (BO = 0.527)		sql (BO=0.577)	sorguları (BO=0.632)
itibariyle (BO = 0.421)					x	biridir (BO = 0.527)		enjeksiyon (BO = 0.527)	tabanlı (BO=0.632)
dünyadaki (BO = 0.421)					x	unsurları (BO = 0.527)		çalıştırma (BO = 0.527)	modellenmiş (BO=0.632)
yıllarında (BO = 0.421)					x	konuları (BO = 0.527)		atı (BO = 0.527)	girişinin (BO=0.632)
alanındaki (BO = 0.421)					x	özellikleri (BO = 0.527)		injection (BO=0.632)	okuyucuları (BO=0.632)

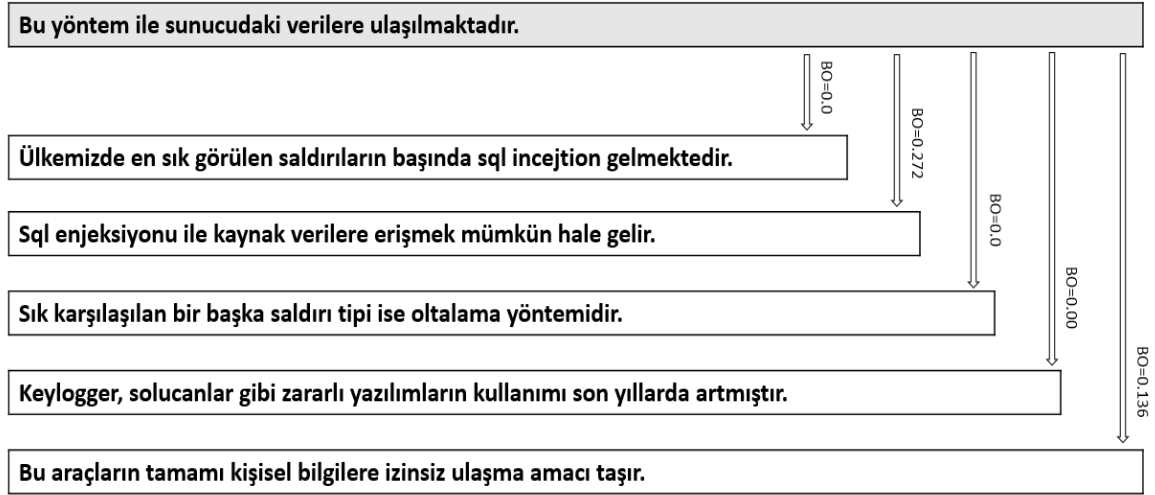
Şekil 2.12. Deney 1 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım1)

Şekil incelendiğinde, pivot cümlede bulunup da karşılaştırılan cümlede bulunmayan her kelime için Word2Vec modelinden öneriler alındığı görülmektedir. Bu öneriler ilk kelimeden başlayarak sırasıyla son kelimeye kadar alınır. Her bir kelime, önerilen yeni kelimeler ile yer değiştirerek cümle yeniden oluşturulur. Oluşan cümle ile karşılaştırılan cümledeki yeni benzerlik oranı eskisinden yüksek ise kelime cümlede bırakılır. Güncellenen benzerlik oranı ise yeni benzerlik oranı olarak saklanır. Yapılan işlemler incelendiğinde “Türkiye’de” kelimesi yerine “ülkümüzde” kelimesinin kullanımı benzerlik oranını yükseltmiştir. Sonraki adımda “tiplerinden” kelimesi için model öneri sunamamıştır. Bir sonraki adımda “biri” kelimesi için modelden öneriler sunulmuş fakat bu öneriler benzerlik oranında değişime sebep olmamıştır. Bir adım sonra “enjeksiyonu” kelimesi için öneriler alınmış ve ilk artış “sql” kelimesi kullanıldığında gözlemlenmiştir. Fakat öneriler içerisinde daha yüksek artışa neden olan “injection” kelimesi bulunduğu için en yüksek benzerlik skorunu veren “injection” kelimesi yeni cümlede yerini almıştır. Son adımda ise “yöntemidir” kelimesine modelden alternatifler sunulmuş fakat bu öneriler sonuca etki etmediği için değerlendirilmemiştir. Yapılan işlemler incelendiğinde, sonucun ciddi oranda değiştiği görülmektedir. İlk haliyle 0.421 olan benzerlik oranı, nihai haliyle

0.632'ye yükselmiştir. İşlemlerin ardından `metin1`'e ait pivot cümle güncellenerek aşağıdaki hale çevrilmiştir.

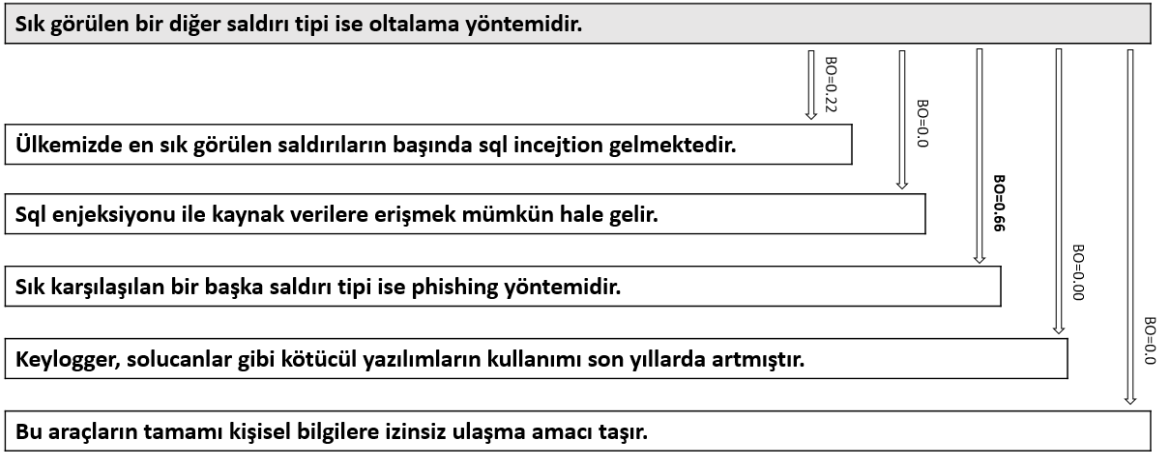
pivot cümle = “Türkiye’de en sık görülen saldırı tiplerinden biri sql enjeksiyonu yöntemidir.”
yeni cümle = “**Ülkemizde** en sık görülen saldırı tiplerinden biri sql **injection** yöntemidir.”

Bu aşamanın ardından bir sonraki cümle pivot cümle seçilir ve aynı işlemler sıradaki pivot cümle için tekrarlanır. Sıradaki pivot cümle için gerçekleşen işlemler Şekil 2.13’de adımlar halinde gösterilmektedir.



Şekil 2.13. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 2)

Pivot cümle ile diğer cümlelerin benzerlik sonuçları incelendiğinde oranların tamamı 0.3'ten küçük görünmektedir. Bu nedenle bu cümlelerin metin2 içerisindeki herhangi bir cümle ile yeterli yakınlıkta olmadığı kanaatine varılır. İşlemlere devam etmek için pivot cümle bir kaydırılarak metin1 içerisinde sıradaki cümle seçilir. Seçilen yeni pivot cümle için benzerlik işlemleri detaylı şekilde Şekil 2.14'te gösterilmiştir.



Şekil 2.14. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 3)

Şekil 2.14 incelendiğinde pivot cümle ile metin2’de bulunan 3. cümle arasında 0.66 oranında benzerlik olduğu göze çarpmaktadır. Algoritmanın ikinci adımı için seçim değeri olan 0.3 değerini aştığı için bu cümleler Word2Vec modeli yardımıyla tekrar ele alınacaktır. Modelden gelen öneriler doğrultusunda pivot cümlelerin değişimi Şekil 2.15’te gösterilmektedir.

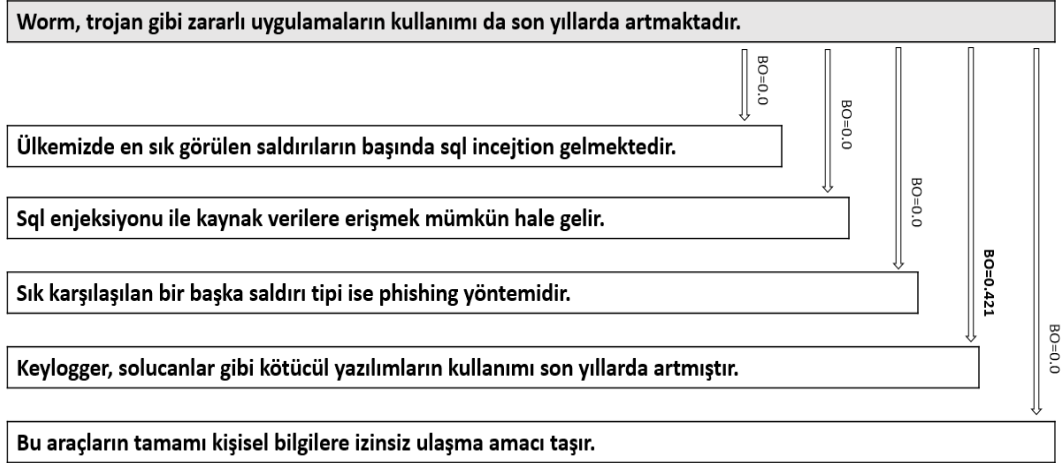
Sık	görülen	bir	diğer	saldırı	tipi	ise	ortalama	yöntemidir
	incelenen (BO=0.66)		öte (BO=0.66)				phishing (BO=0.88)	
	yöntemin (BO=0.66)		başka (BO=0.77)				spam (BO = 0.77)	
	çalışmasında (BO=0.66)		yandan (BO=0.66)				içerikli (BO = 0.77)	
	süreleri (BO=0.66)		mensuptur (BO=0.66)				sahte (BO = 0.77)	
	yönteminin (BO=0.66)		tanesi (BO=0.66)				bot (BO=0.77)	

Şekil 2.15. Deney 1 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım3)

İşlemler incelendiğinde; başlangıç benzerlik oranı 0.66 bulunan cümlelerin, anlamsal benzerlikle birlikte 0.88 seviyesinde benzer çıktıkları görülmektedir. Bu süreçte pivot cümlede bulunan “diğer” kelimesi, model tarafından önerilen “başka” kelimesi ile değiştirilmiştir. Bu işlem sonucunda benzerlik oranı 0.77’ye yükselmiştir. Son adımda ise pivot cümlede bulunan “ortalama” kelimesi “phishing” olarak değiştirilmiş ve benzerlik oranı 0.88 seviyesine kadar çıkmıştır. Bu oran iki cümlelerin neredeyse aynı anlamda

olduğunu göstermektedir. Değişen kelimeler incelendiğinde, eş anlamlı sözcük kullanımı ve terimlerin Türkçeleştirilmesi nedeniyle yakalanamayan benzerliklerin tespit edildiği rahatlıkla görülmektedir.

Bir sonraki adımda pivot cümle olarak metin1'den sıradaki cümle seçilir. Seçilen cümle ile metin2'den alınan cümlelerin benzerlik oranları Şekil 2.16'da görülmektedir.



Şekil 2.16. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 4)

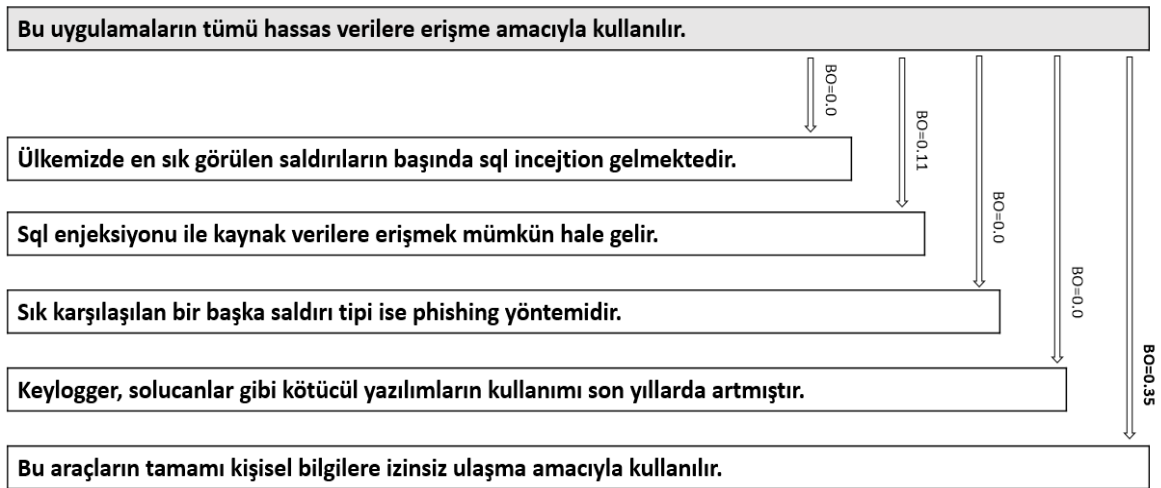
Seçilen pivot cümle ile metin2'ye ait cümleler arasında sırayla benzerlik oranları bulunmuştur. Çıkan sonuçlara göre metin2'den 4. cümle ile pivot cümle arasında 0.421 oranında yapısal benzerlik olduğu görülmektedir. Bu benzerlik oranı eşik değer olan 0.3'ten daha yüksek olduğu için önerilen yönteme ait algorithma ikinci adıma geçilir. Algoritmanın ikinci adımında yapılacak anlamsal benzerlik tespitleri ile benzerlik oranı tekrar hesaplanacaktır. Bu işlemler Şekil 2.17'de incelenmektedir.

Worm	trojan	gibi	zararlı	uygulamaların	kullanımı	son	yıllarda	artmaktadır
solucanlar (BO=0.52)	solucanlar (X) (BO=0.57)		yazılımlar (BO=0.52)	süreçlerine (BO=0.63)				geçtikçe (BO=0.63)
trojan (BO=0.38)	worm (BO=0.52)		virüs (BO=0.52)	araçların (BO=0.63)				sıfırıncı (BO=0.63)
ıdap (BO=0.42)	yığın (BO=0.52)		kötücül (BO=0.63)	süreçler (BO=0.63)				yaşamı (BO=0.63)
sazan (BO=0.42)	kaydedici (BO=0.52)		casus (BO=0.52)	sağlaması (BO=0.63)				biraz (BO=0.63)
cüzdanlar (BO=0.42)	bedava (BO=0.52)		spam (BO=0.52)	işlerin (BO=0.63)				düşünülrse (BO=0.63)

Şekil 2.17. Deney 1 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 4)

Şekil 2.17 incelendiğinde pivot cümle ile metin2’den alınan cümle arasındaki başlangıç benzerlik değerinin daha da yükseldiği görülmektedir. Bu cümle için diğer örneklerden farklı olarak yeşil renk ile boyanan “solucanlar” kelimesi dikkat çekmektedir. Bu kelime anlık benzerlik oranını yükseltmiş fakat bu artış göz önünde bulundurulmamıştır. Sebebi bir önceki kelimenin (worm), yine “solucanlar” kelimesi ile yer değiştirmesidir. Solucanlar kelimesi artık yeni cümleye ait bir kelime olduğu için değerlendirmeye alınmamıştır. Çünkü yapısal benzerlik oranı veren algoritmalar, metinler arasındaki ortak kelimenin birden fazla kez kullanılmasıyla oranı yükseltirler. Bu nedenle bir kelime cümle içerisinde bir kez kullanıldıktan sonra model önerisi ile tekrar kullanımına izin verilmemektedir. Aynı sebepten ötürü cümlelerde ortak geçen kelimeler için Word2Vec modelinden alternatif istenmez. Sonraki adımda “zararlı” kelimesi yerine önerilen “kötücül” kelimesi yer değiştirmiştir. Bu değişim ile benzerlik oranı 0.63 seviyesine yükselmiştir. Cümle içerisinde alternatif aranan diğer kelimeler toplam benzerlik oranına etki etmemiştir. Bu cümle özelinde değişen kelimelerle, eş anlamlı kullanım (zararlı-kötücül), benzer kategoriye ait kelimeler (trojan-solucan) ve yabancı terimlerin Türkçeleştirilmesi (worm-solucan) gibi sorunlara çözüm üretildiği görülmektedir.

Algoritmanın bir sonraki adımında pivot cümle olarak son cümle seçilecektir. Bu cümle ile metin2 içerisindeki cümlelerin benzerlikleri Şekil 2.18’de gösterilmektedir.



Şekil 2.18. Deney 1 için pivot cümle ile metin2 arasındaki benzerlikler (adım 5)

Pivot cümle ile metin2’nin benzerlik oranları incelendiğinde son cümle ile 0.35 oranında yapısal benzerlik bulunduğu görülmektedir. Bu cümle ile anlamsal benzerlik

destekli yeni bir oran çıkarabilmek için Word2Vec modelinden faydalanılır. Word2Vec modelinin pivot cümlede geçen kelimelere sunduğu alternatif kelimeler ve sonuca etkisi Şekil 2.19’da gösterilmiştir.

Bu	uygulamaların	tümü	kişisel	verilere	erişme	amacıyla	kullanılır
süreçlerine (BO=0.35)	yerleştirme (BO = 0.47)	hassas (BO=0.58)	bilgilere (BO=0.70)	erişimini (BO=0.70)			
araçların (BO=0.47)	tamamı (BO = 0.47)	verilerin (BO=0.47)	yollara (BO=0.58)	sağlayabilir (BO=0.70)			
süreçler (BO=0.35)	kopyası (BO = 0.47)	mahremiyeti (BO=0.47)	işlemlerin (BO=0.58)	değişiminin (BO=0.70)			
sağlaması (BO=0.35)	kaydedip (BO = 0.47)	bilgilerin (BO=0.47)	silinmesi (BO=0.58)	hesapların (BO=0.70)			
işlerin (BO=0.35)	elemanların (BO = 0.47)	gizliliği (BO=0.47)	erişimini (BO=0.58)	başkalarının (BO=0.70)			

Şekil 2.19. Deney 1 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 5)

Cümle içerisindeki değişiklikler ile benzerlik oranı 0.35’ten 0.70 seviyesine yükselmiştir. Pivot cümle ile karşılaştırılan cümlelerin anlamsal benzerliği yapısal benzerliğine göre daha yüksek sonuçlar vermektedir. Algoritmanın ikinci kısmında Word2Vec modeli tarafından önerilen kelimeler incelendiğinde eş anlamlı veya yakın anlamlı kelimeler olduğu görülecektir. Bu önerilerden ilki “uygulamaların” kelimesi yerine “araçların” kelimesidir. İki kelime tam olarak aynı anlamda olmasa dahi anlam bütünlüğünde yakın görülmektedir. Bir diğer öneri “kişisel” kelimesi yerine “hassas” kelimesi gelecek şekilde yapılmıştır. Bu iki kelime ilk bakışta yakın görülmemektedir. Fakat Bilgi Güvenliği başlığı altında “kişisel veriler” ifadesi “hassas veriler/bilgiler” ifadesi ile yakın görülebilir. Çünkü güvenlik alanında ‘kişisel olan’ yani ‘özel olan’ içeriğin hassas içerik olduğu sonucu yakın bir tahmindir. Son olarak ise “verilere” kelimesine karşılık “bilgilere” kelimesi önerilmiştir. Burada doğrudan eş anlamlı sözcüklerin kullanıldığını gözlemlemek mümkündür. Cümle içerisinde eş anlamlı farklı bir sözcüğün kullanılmasıyla atlanan benzerlik saptanmış ve yeni oran arttırılmıştır.

2.7.2. Deney 1 Sonucunda Elde Edilen Bulgular

Deney 1 kapsamında yapılan işlemler sonucunda, kullanılan metinler ve metin1’in yeni hali Tablo 2.3’te gösterildiği gibi güncellenmiştir. Altı çizili ve koyu yazılan

kelimeler metin1'den önerilen yöntem aracılığıyla yakalanan gizli benzerlikleri ifade etmektedir.

Tablo 2.3. Deney 1 sonucunda metinlerin yeni hali ve değişen kelimeler

metin1	Türkiye'de en sık görülen saldırı tiplerinden biri sql enjeksiyonu yöntemidir. Bu yöntem ile sunucudaki verilere ulaşılmaktadır. Sık görülen bir diğer saldırı tipi ise ortalama yöntemidir. Worm, trojan gibi zararlı uygulamaların kullanımı da son yıllarda artmaktadır. Bu uygulamaların tümü kişisel verilere erişme amacıyla kullanılır.
yeni metin1	Ülkemizde en sık görülen saldırı tiplerinden biri sql injection yöntemidir. Bu yöntem ile sunucudaki verilere ulaşılmaktadır. Sık görülen bir başka saldırı tipi ise phishing yöntemidir. Solucanlar , trojan gibi kötücül uygulamaların kullanımı da son yıllarda artmaktadır. Bu araçların tümü hassas bilgilere erişme amacıyla kullanılır.
metin2	Ülkemizde en sık görülen saldırıların başında sql injection gelmektedir. Sql enjeksiyonu ile kaynak verilere erişmek mümkün hale gelir. Sık karşılaşılan bir başka saldırı tipi ise phishing yöntemidir. Keylogger, solucanlar gibi kötücül yazılımların kullanımı son yıllarda artmıştır. Bu araçların tamamı hassas bilgilere izinsiz ulaşma amacıyla kullanılır.

Önerilen yöntemi kullanmadan önceki haliyle metin1 ve metin2 arasındaki Kosinüs Benzerliği 0.525 olarak hesaplanmıştır. Metinler üzerinde önerilen yöntemin kullanılması ile birlikte yeni benzerlik oranı 0.686 olarak tekrar hesaplanmıştır. Elde edilen sonuçlar incelendiğinde iki metnin anlamsal yakınlığının daha yüksek olduğu görülmektedir. Bu sonuçların gösteriminde metin1 yerine m1, metin2 yerine m2 ve yeni metin1 için de ym1 ifadeleri kullanılmıştır.

$$Kos_{(m1,m2)} = BO1 = 0.525$$

$$Kos_{(ym1,m2)} = BO2 = 0.686$$

Yapılan deney sonrasında metin1 ve metin2 için cümle bazında değişen benzerliklerin oranları Tablo 2.4'te gösterilmektedir.

Tablo 2.4. Deney 1 sonucunda benzerlik oranlarının cümle başına değişimi

metin1	İlk Benzerlik Oranı (BO1)	Son Benzerlik Oranı (BO2)	Değişim (%)
cümle 1	0.42	0.63	50
cümle 2	0.27	0.27	0
cümle 3	0.66	0.88	33.3
cümle 4	0.42	0.63	50
cümle 5	0.35	0.70	100
total	0.52	0.68	30

Tablo 2.4'te metin1'e ait cümleler ve metnin tamamı (total satırı) için benzerlik oranındaki değişim gösterilmektedir. Önerilen yöntemin kullanılmasıyla metinler arasındaki yapısal benzerliğin %100'e kadar sapmadan kurtulduğu görülmektedir. Anlam olarak birbirine oldukça yakın olan cümleler için, yapısal yaklaşımlar ile yakalanamayan benzerlikler önerilen yöntem ile yakalanmıştır. İki metni bütün olarak yapısal yöntemlerle karşılaştırdığımıza elde edilen benzerlik oranının %30 oranında saptığı gözlemlenmiştir. Önerilen yöntem ile birlikte nihai benzerlik oranı ilk ölçülen orana göre %30 artmıştır. Böylelikle anlamsal olarak birbirine oldukça yakın görülen metinlerin benzerliği daha tutarlı şekilde tespit edilmiştir. Yukarıda gösterilen değerlerin doğruluğunu incelemek için son olarak metin1'den yeni metin1'e geçişte değiştirilen kelimeler incelenmelidir. Bu kelimelerin listeli hali Tablo 2.5'te gösterildiği gibidir.

Tablo 2.5. Deney 1 sonucunda algoritma tarafından saptanan yakın kelimeler

Kelimenin metin1'deki Hali	Yöntemin Önerdiği Alternatif
Türkiyede	Ülkemizde
eneksiyonu	injection
diğer	başka
oltalama	phishing
worm	solucanlar
zararlı	kötücül
uygulamaların	yazılımların
kişisel	hassas
verilere	bilgilere

Tablo 2.5 incelendiğinde Word2Vec modeli tarafından önerilen kelimelerin anlamsal olarak cümle akışına uygun olduğu görülmektedir. Geliştirilen yöntemin amacına uygun olarak eş anlam, yakın anlam, terim karşılığı ve yabancı ifade karşılığı bulmada modelin başarılı sonuçlar verdiği gözlemlenmiştir.

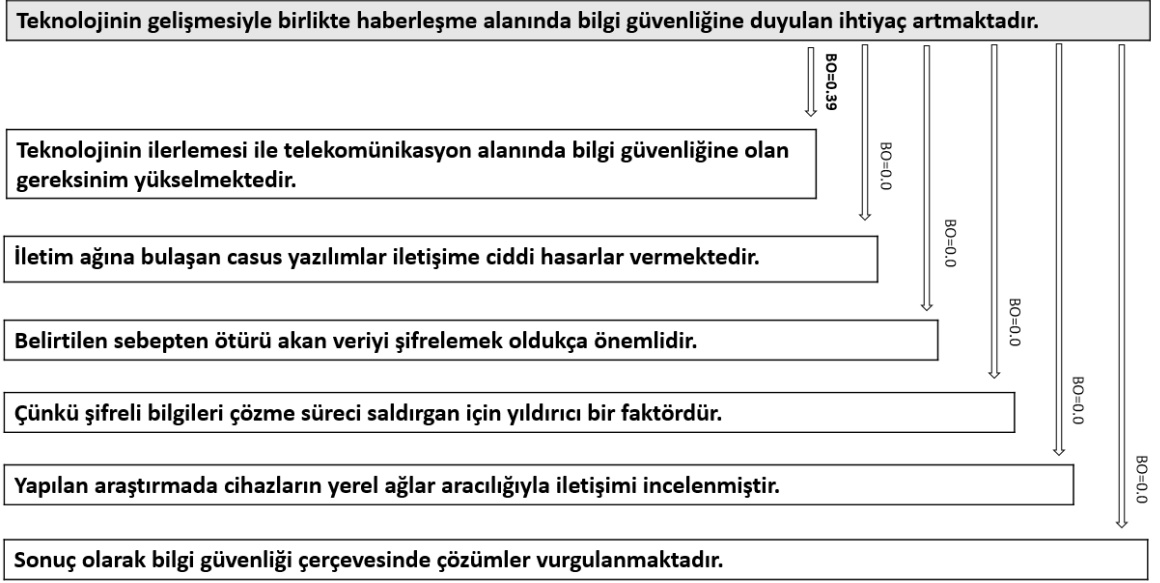
2.7.3. Bilgi Güvenliği Verileri ile Deney 2

Bu bölümde, yapılan çalışmanın başarımını test edebilmek için Deney 1'de kullanılan veri setinden daha uzun iki metinle deneyler yapılmıştır. Kullanılan metinler yine Bilgi Güvenliği alanına dahil olan teknik terimler içermektedir. Önerilen yönteme ait Word2Vec modeli Bilgi Güvenliği alanından dokümanlarla eğitildiği için test verileri bu alandan seçilmiştir. Deneyde kullanılan metnin farklı bir konuda ve mümkün olduğunca Deney 1'de kullanılmayan kelimelerden oluşmasına dikkat edilmiştir. Bu metinler Tablo 2.6'da gösterilmektedir.

Tablo 2.6. Deney 2 için seçilen metinler

metin1	Teknolojinin gelişmesiyle birlikte haberleşme alanında bilgi güvenliğine duyulan ihtiyaç artmaktadır. İletim ağına sızan zararlı yazılımlar bu alanda en önemli tehdit kaynağıdır. Bu nedenle aktarılan verinin kriptolu olması oldukça önemlidir. Çünkü şifreli verileri deşifreleme aşaması saldırgan için caydırıcı olabilmektedir. Yapılan çalışmada, cihazların lokal ağlar üzerinden bağlantısı incelenmiştir. Bu ağların güvenlik seviyeleri ve saldırıya açıklıkları tartışılmıştır. Sonuç olarak, bu alanda bilgi güvenliği kapsamında alınabilecek önlemler vurgulanmıştır.
metin2	Teknolojinin ilerlemesi ile telekomünikasyon alanında bilgi güvenliğine olan gereksinim yükselmektedir. İletim ağına bulaşan casus yazılımlar iletişime ciddi hasarlar vermektedir. Belirtilen sebepten ötürü akan veriyi şifrelemek oldukça önemlidir. Çünkü şifreli bilgileri çözme süreci saldırgan için yıldırıcı bir faktördür. Yapılan araştırmada cihazların yerel ağlar aracılığıyla iletişimi incelenmiştir. Sonuç olarak bilgi güvenliği çerçevesinde çözümler vurgulanmaktadır.

Yapılacak deneyde seçilen metinler Deney 1'e göre daha fazla cümle içermekte ve daha uzun cümlelerden oluşmaktadır. Sabit tutulacak metin, metin2 olarak seçilmiştir. Öncelikle metinlerin cümlelere ayrılma işlemi yapılmış ve metin1'den cümleler sırasıyla pivot cümle işlemlerine tabi tutulmuştur. Bu işlemler metin1'in ilk cümlesiyle birlikte başlamaktadır. İlk pivot cümlenin metin2'den cümleler ile benzerlik oranları Şekil 2.20'de gösterilmektedir.



Şekil 2.20. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 1)

Seçilen pivot cümle ile diğer cümlelerin benzerlik oranları karşılaştırıldığında metin2'ye ait ilk cümle ile 0.39 oranda benzerlik öne çıkmaktadır. Bu oran sonucunda, önerilen yöntemin ikinci aşamasından gelen kelime önerileri ile anlamsal yakınlık testleri yapılacaktır. Yapılan işlemler Şekil 2.21'de gösterilmektedir.

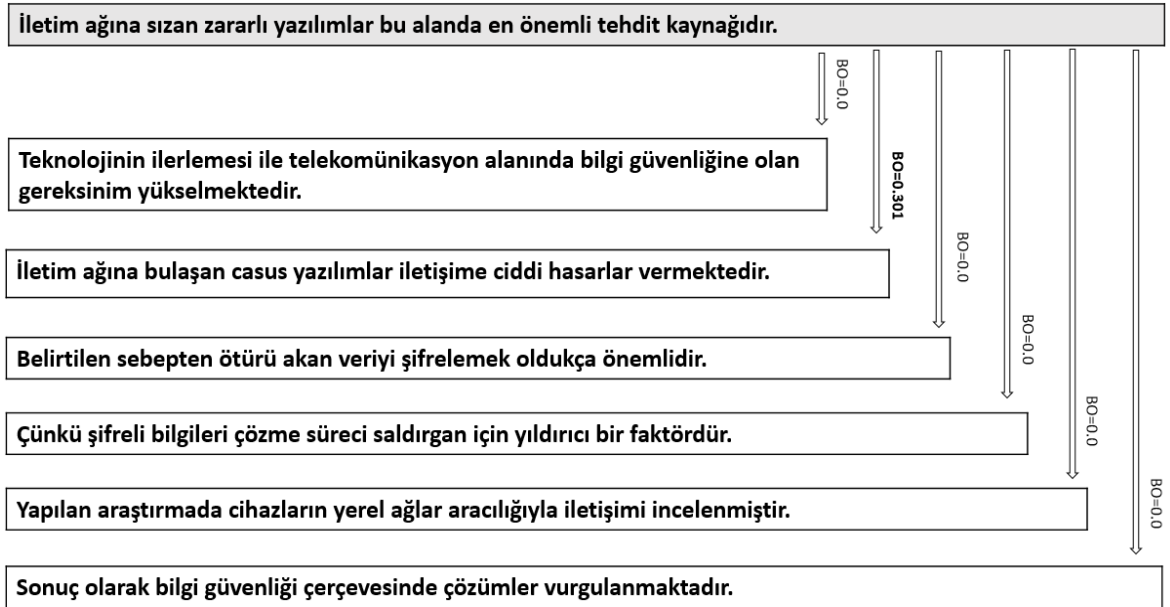
Teknolojinin	gelişmesiyle	birlikte	haberleşme	alanında	bilgi	güvenliğine	duyulan	ihtiyaç	artmaktadır
başlanmasıyla (BO = 0.39)	beraber (BO = 0.49)	servisleri (BO = 0.49)					ihtiyaç (BO = 0.60)	duyulan (BO = 0.60)	geçtikçe (BO = 0.70)
gelişimiyle (BO = 0.39)	kavramı (BO = 0.49)	telekomünikasyon (BO = 0.60)					duyduğu (BO = 0.60)	gereksinim (BO = 0.70)	yaşamı (BO = 0.70)
ilerlemesi (BO = 0.49)	gelişmeler (BO = 0.49)	ulaştırma (BO = 0.49)					mutlaka (BO = 0.60)	duymaktadır (BO = 0.60)	diliminde (BO = 0.70)
ilerlemelerle (BO = 0.39)	teknolojinin (BO = 0.49)	bakanlığı (BO = 0.49)					kullanıcılara (BO = 0.60)	zorunlu (BO = 0.60)	sıfırıncı (BO = 0.70)
teknolojideki (BO = 0.39)	insanların (BO = 0.49)	denizcilik (BO = 0.49)					verimliliği (BO = 0.60)	duyduğu (BO = 0.60)	artmıştır (BO = 0.70)

Şekil 2.21. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 1)

Pivot cümlelerin karşılaştırıldığı cümleyle başlangıçtaki benzerlik oranı 0.39 olarak görülmüştür. Word2Vec modelinden gelen öneriler ile 3 kelimenin değişmesi uygun bulunmuştur. Bu kelimeler ve modelin önerdiği alternatifler aşağıda maddeler halinde gösterilmektedir.

- gelişmesiyle → ilerlemesi
- haberleşme → telekomünikasyon
- ihtiyaç → gereksinim

Maddelere göre modelin önerdiği kelimeler doğrudan metin2'deki ilk cümlede geçen kelimeler olmuştur. Bu kelimeler anlamsal yakınlık ve eş anlamlılık barındırmaktadır. Modelin seçtiği kelimeler arasında anlamı bozan bir kelime göze çarpmamaktadır. Yapılan değişikliklerden sonra cümleler arası benzerlik oranı 0.70'e çıkmıştır. Bir sonraki adımda pivot cümle metin1'in ikinci cümlesi olacaktır. Bu cümle için metin2 ile benzerlik oranları Şekil 2.22'de gösterilmektedir.



Şekil 2.22. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 2)

Şekil 2.22'de pivot cümle ile diğer cümlelerin benzerlik oranları görülmektedir. Diğer tüm cümleler için benzerlik oranı 0 iken sadece ikinci cümlede 0.301 benzerlik oranı bulunmuştur. Bu değer algoritmanın ikinci adımına geçmek için sınır üzerindedir. Bu nedenle cümle için kelime önerileri alma safhasına geçilecektir. Bu aşama Şekil 2.23'te detaylıca gösterilmektedir.

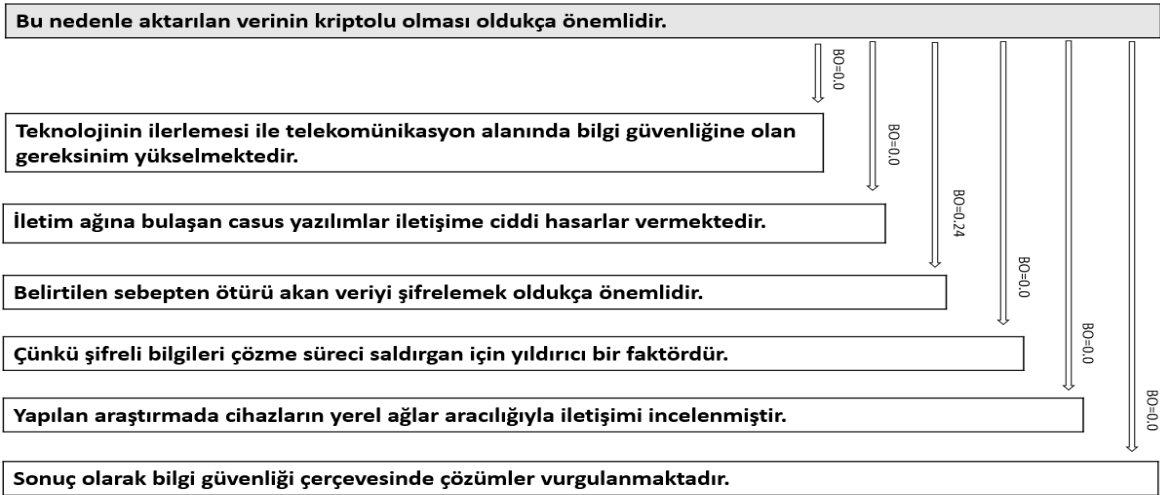
iletim	ağına	sızan	zararlı	yazılımlar	bu	alandaki	en	önemli	tehdit	kaynağıdır
	arayan (BO = 0.30)	yazılımlar (BO = 0.40)			x	ülkede (BO = 0.50)	x	konulardan (BO = 0.50)	gelebilecek (BO = 0.60)	?
	karşıdaki (BO = 0.30)	kötücül (BO = 0.40)			x	gelişmelerin (BO = 0.50)	x	biridir (BO = 0.50)	teşkil (BO = 0.60)	?
	hesabınız (BO = 0.30)	virüs (BO = 0.40)			x	alandaki (BO = 0.50)	x	unsurlarından (BO = 0.50)	potansiyel (BO = 0.60)	?
	bulaşma (BO = 0.30)	programlar (BO = 0.40)			x	gelişme (BO = 0.50)	x	ciddi (BO = 0.60)	tehlikeye (BO = 0.60)	?
	bulaşan (BO = 0.40)	casus (BO = 0.50)			x	gelişmeleri (BO = 0.50)	x	oynamıştır (BO = 0.50)	tehlike (BO = 0.60)	?

Şekil 2.23. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 2)

Pivot cümle üzerinde yapılan değişiklikler incelendiğinde 3 kelimenin değiştiği görülmektedir. Bu kelimelere ilaveten “?” ile gösterilen bir sütun bulunmaktadır. Bu sütunda “kaynağıdır” kelimesine karşılık Word2Vec modelinden herhangi bir öneri gelmediği görülür. Modelin eğitim için kullandığı derlemde bu kelimenin 5’ten az geçmesi bu soruna neden olmaktadır. Ayrıca “bu” ve “en” kelimeleri için de öneri alınmamıştır. Bu kelimeler ise derlemde gereksiz kelimeleri (stopwords) ayıklama esnasında silinmiştir. Bu nedenle bu kelimeler için modelden bir öneri alınamamaktadır. Esas önerilerin alındığı ve benzerlik oranını arttıran kelimeler aşağıda maddeler halinde gösterilmektedir.

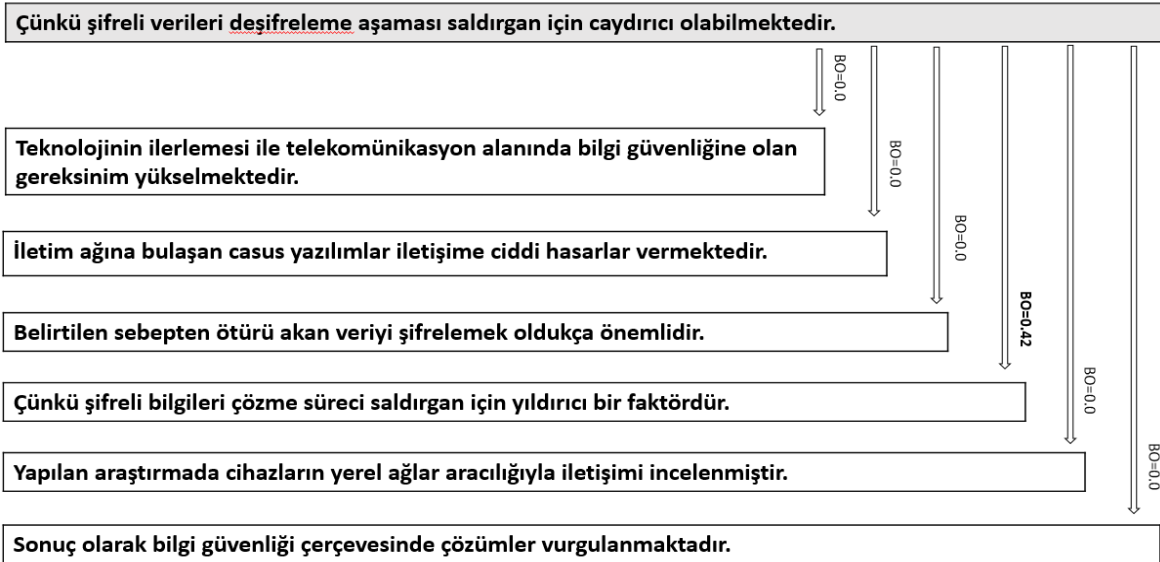
- sızan → bulaşan
- zararlı → casus
- önemli → ciddi

Değişen kelimeler incelendiğinde anlamı bozacak herhangi bir öneriye rastlanmamaktadır. Bu değişimler sonucunda cümlelerin benzerlik oranları 0.60 seviyesine çıkmıştır. Bir sonraki adımda pivot cümle değişecektir. Sıradaki pivot cümlelerin benzerlik oranları Şekil 2.24’te gösterilmiştir.



Şekil 2.24. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 3)

Pivot cümle ile metin2’den gelen cümleler incelendiğinde, hiçbir cümle ile yeterli oranda (0.3) benzerlik taşımadığı görülmektedir. En yakın olarak metin2’den 3. cümle ile 0.24 oranında yapısal benzerliğe sahiptir. Fakat bu oran önerilen yöntemin ikinci aşamasına geçmek için yeterli değildir. Bu nedenle pivot cümle olarak metin1’den bir sonraki cümle seçilecek ve işlemler devam edecektir. Seçilen pivot cümleye ait benzerlik oranları Şekil 2.25’te gösterilmiştir.



Şekil 2.25. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 4)

Şekil 2.25’te görüldüğü gibi pivot cümle ile metin2’den 4. cümle arasında 0.42 oranında yapısal benzerlik bulunmaktadır. Elde edilen sonuçlar doğrultusunda

algoritmanın ikinci adımına geçilecektir. Bu aşamada pivot cümle için önerilen alternatif kelime grupları Şekil 2.26’da gösterilmektedir.

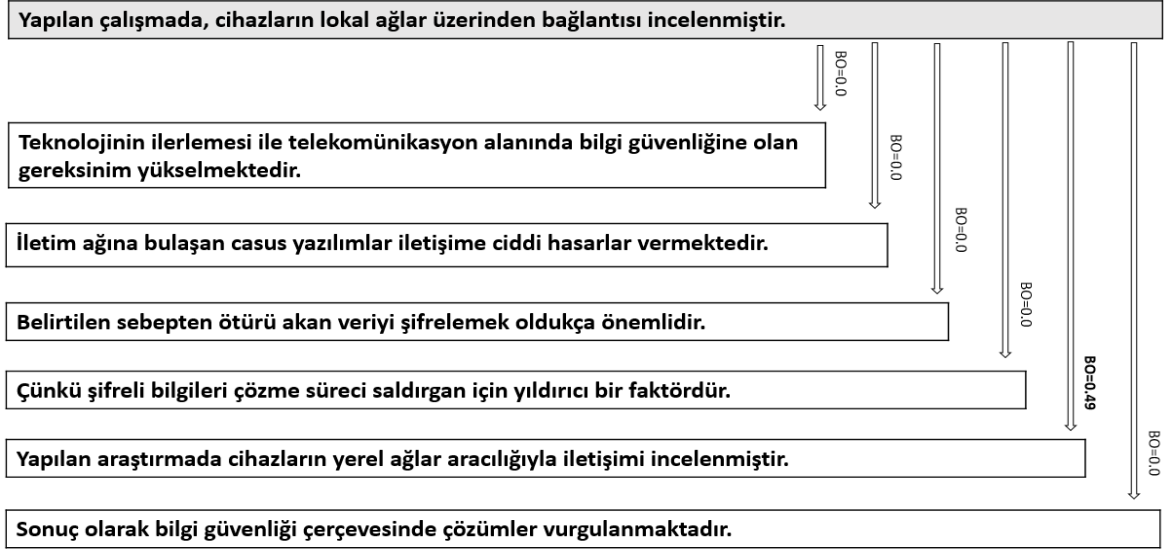
Çünkü	şifreli	verileri	deşifreleme	aşaması	saldırgan	için	caydırıcı	olabilmektedir
		verilerinin (BO = 0.42)	<u>tea</u> (BO = 0.52)	<u>laplace</u> (BO = 0.63)			ulaşılmasını (BO = 0.63)	zararlara (BO = 0.63)
		bilgileri (BO = 0.52)	eliptik (BO = 0.52)	gömme (BO = 0.63)			yapmaktadır (BO = 0.63)	olabilecek (BO = 0.63)
		kötüye (BO = 0.42)	algoritmasının (BO = 0.52)	<u>serpent</u> (BO = 0.63)			çağrısında (BO = 0.63)	yanlarını (BO = 0.63)
		hassas (BO = 0.42)	<u>aria</u> (BO = 0.52)	şifresindeki (BO = 0.63)			bulunmasına (BO = 0.63)	açan (BO = 0.63)
		veriler (BO = 0.42)	çözme (BO = 0.63)	genişletilmiş (BO = 0.63)			sekteye (BO = 0.63)	olmaktadır (BO = 0.63)

Şekil 2.26. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 4)

Alınan sonuçlar incelendiğinde pivot cümleden iki kelimenin değişimi sağlanmıştır. Word2Vec modeli tarafından önerilen bu kelimeler aşağıda maddeler halinde gösterilmiştir.

- verileri → bilgileri
- deşifreleme → çözme

Kullanılan yöntemin önerdiği kelimelerden “veri” ve “bilgi” kelimeleri Deney1’de de karşılaşılan bir örnektir. Fakat “deşifreleme” kelimesine karşılık olarak “çözme” kelimesinin önerilmesi modelin başarısına bir örnek teşkil etmektedir. Çünkü “çözme” kelimesini, “şifre çözmek” anlamında kullanılan “deşifreleme” kelimesine karşılık önermiştir. Bu öneri, Bilgi Güvenliği alanına dair modelin tutarlı tavsiyeler verdiğini göstermektedir. Yapılan işlemlerin ardından pivot cümle olarak sıradaki cümle seçilecektir. Bu cümleye ilişkin benzerlik oranları Şekil 2.27’de gösterilmiştir.



Şekil 2.27. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 5)

Elde edilen sonuçlara göre, pivot cümle ile metin2'den 5. cümle yapısal olarak 0.49 oranında birbirlerine benzemektedir. Bu oranın anlamsal benzerlik desteğiyle daha üst seviyelere çıkma ihtimali algoritmanın ikinci adımında değerlendirilecektir. Pivot cümlelerin içerdiği kelimeler için model tarafından önerilen alternatifler Şekil 2.28'de gösterilmektedir.

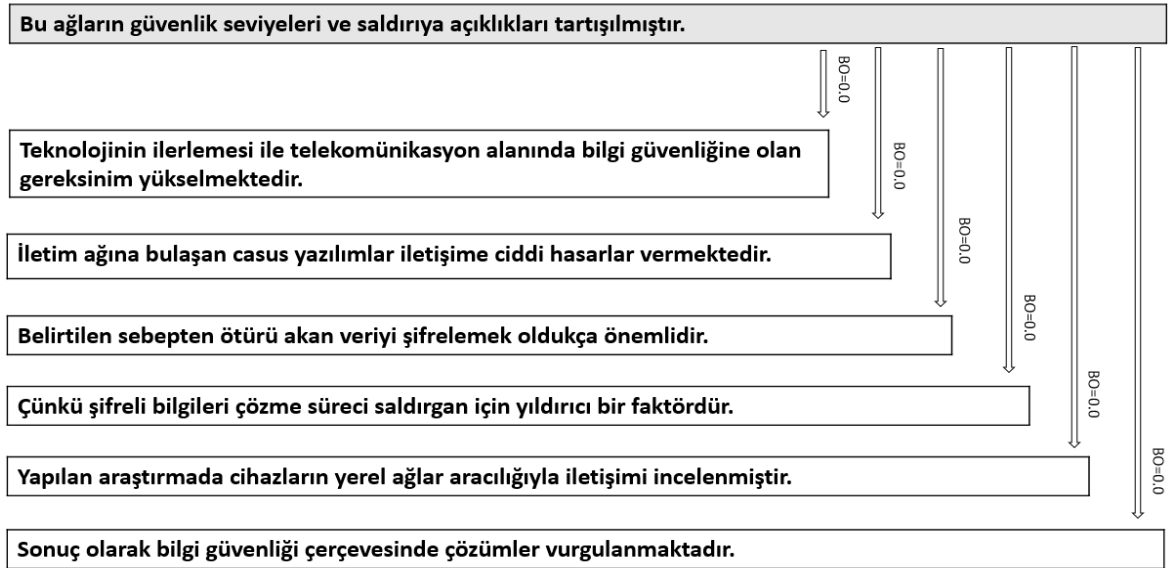
Yapılan	çalışmada	cihazların	lokal	ağlar	üzerinden	bağlantısı	incelenmiştir
	çalışmasında (BO = 0.49)		saatler (BO = 0.62)		aracılığıyla (BO = 0.74)	birbirleriyle (BO = 0.74)	
	araştırmada (BO = 0.62)		yönlendiriciler (BO = 0.62)		vasıtasıyla (BO = 0.62)	sunucular (BO = 0.74)	
	arkadaşlarının (BO = 0.49)		saldırganlar (BO = 0.62)		sayesinde (BO = 0.62)	bağlanarak (BO = 0.74)	
	bölümde (BO = 0.49)		mesafede (BO = 0.62)		tarayıcı (BO = 0.62)	gömülü (BO = 0.74)	
	makalede (BO = 0.49)		spesifik (BO = 0.62)		site (BO = 0.62)	iletişimi (BO = 0.87)	

Şekil 2.28. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 5)

Önerilen yöntemden gelen veriler incelendiğinde, pivot cümle için 3 adet yeni kelime önerildiği görülmektedir. Bu üç kelimenin cümle içerisinde değişimi ile hedef cümle arasındaki benzerlik oranı yükselecektir. Pivot cümle ile hedef cümlelerin başlangıçta 0.49 bulunan benzerlik oranı, yapılan işlemler ile 0.87 oranına yükselmiştir. Bu oran iki cümlelerin anlamsal olarak çok yakın olduğunu göstermektedir. Cümle içerisinde değiştirilmesi önerilen kelimeler liste halinde aşağıda sunulmuştur.

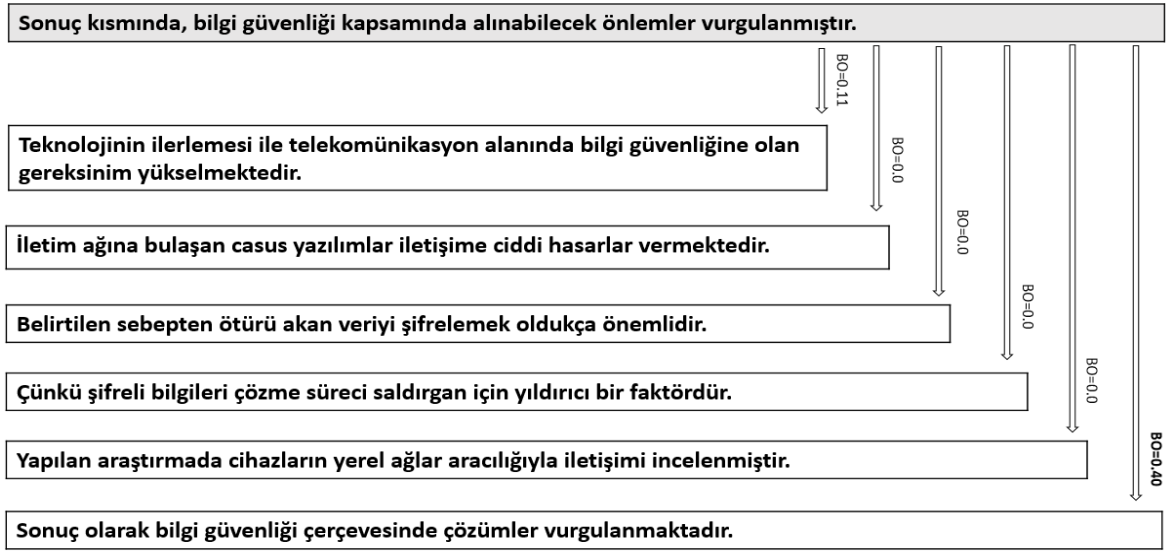
- çalışmada → araştırmada
- üzerinden → aracılığıyla
- bağlantısı → iletişimi

Değişim için önerilen kelimelerin anlamca yakın kelimeler olduğu gözlemlenmektedir. Bu kelimelerin cümle içerisindeki değişimi anlamsal bozukluğa sebep olmamaktadır. Böylelikle eş anlamlı veya yakın anlamlı kelime kullanımından kaynaklanan hesap sapmaları düzeltilmiştir. Bir sonraki aşamada pivot cümle değişecek ve benzer işlemlere tabi tutulacaktır. Yeni pivot cümle için metin2 ile benzerlik oranları Şekil 2.29'da gösterilmektedir.



Şekil 2.29. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 6)

Pivot cümle ile metin2 içerisindeki hiçbir cümlelerin benzemediği sonucu alınmıştır. Sonucun teyit edilmesi için pivot cümlede geçen kelimeleri metin2'de geçen tüm kelimelerle karşılaştırmak gerekmektedir. Yapılan karşılaştırma sonrasında metin2'ye ait kelime havuzundan herhangi birinin pivot cümle içerisinde geçmediği görülmüştür. Böylelikle algoritmada ikinci adıma geçmeden bir sonraki pivot cümle seçilir. Seçilen cümleye ait benzerlik oranları Şekil 2.30'da gösterilmektedir.



Şekil 2.30. Deney 2 için pivot cümle ile metin2 arasındaki benzerlikler (adım 7)

Pivot cümle ile metin2'ye ait son cümle arasındaki 0.40 benzerlik oranı, algoritmanın ikinci adımına geçmeyi gerektirmektedir. Bu iki cümle arasındaki benzerliğin daha doğru hesaplanabilmesi için Word2Vec modelinden kelime önerileri alınacaktır. Modelden gelen kelime önerileri ve bunların benzerlik oranına etkileri Şekil 2.31'de gösterilmektedir.

Sonuç	kısımında	bilgi	güvenliği	kapsamında	alınabilecek	önlemler	vurgulanmıştır
	ayrıntıları (BO = 0.40)			çerçevesinde (BO = 0.53)	risklere (BO = 0.53)	alınabilecek (X) (BO = 0.47)	vurgulanmaktadır (BO = 0.59)
	geliştirilirken (BO = 0.40)			ışığında (BO = 0.40)	önlemler (X) (BO = 0.47)	önlemlerin (BO = 0.53)	yapılmasının (BO = 0.53)
	işbirlikli (BO = 0.40)			bakanlığında (BO = 0.40)	önlemlerin (BO = 0.53)	önlemleri (BO = 0.53)	düzeye (BO = 0.53)
	merkezinde (BO = 0.40)			doğrultusunda (BO = 0.40)	risklerine (BO = 0.53)	riskleri (BO = 0.53)	kaldırılması (BO = 0.53)
	yararlanılarak (BO = 0.40)			kapsamındaki (BO = 0.40)	hazırlıklı (BO = 0.53)	alınmasını (BO = 0.53)	atılacak (BO = 0.53)

Şekil 2.31. Deney 2 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 7)

Son cümle ile birlikte yapılan çalışma Şekil 2.31'de görüldüğü gibi sonuçlandırılmıştır. Elde edilen sonuçlara göre son pivot cümleden iki kelimenin değişimi anlamsal yakınlığı arttırmaktadır. Yapısal benzerliği 0.40 olarak bulunan cümlelerin, anlamsal öneriler ile birlikte sonuç benzerliği 0.59 seviyesine yükselmiştir. Değişmesi tavsiye edilen kelimeler aşağıda maddeler halinde gösterilmektedir.

- kapsamında → çerçevesinde
- vurgulanmıştır → vurgulanmaktadır

Değişen kelimeler incelendiğinde, eş anlamlı kullanım olarak tanımlanacak “kapsamında” ve “çerçevesinde” kelimeleri görülmektedir. Bir diğer kelime değişimi ise “vurgulanmıştır” kelimesi ile “vurgulanmaktadır” kelimeleri arasında olmuştur. Bu kelimelerin kökleri aynıdır. Fakat aldıkları eklerle birlikte belirttikleri zaman değiştiği için anlamsal olarak aynı görülmemektedir. Dolayısıyla bu kelimeler de anlamsal yakınlığa dahil edilmiş ve değişim ile benzerlik oranını yükseltmiştir.

2.7.4. Deney 2 Sonucunda Elde Edilen Bulgular

Deney 2 kapsamında yapılan işlemler sonucunda, kullanılan metinler ve metin1’in yeni hali Tablo 2.7’de gösterildiği gibi güncellenmiştir.

Tablo 2.7. Deney 2 sonucunda metinlerin ve kelimelerin değişimi

metin1	Teknolojinin gelişmesiyle birlikte haberleşme alanında bilgi güvenliğine duyulan ihtiyaç artmaktadır. İletim ağına sızan zararlı yazılımlar bu alanda en önemli tehdit kaynağıdır. Bu nedenle aktarılan verinin kriptolu olması oldukça önemlidir. Çünkü şifreli verileri deşifreleme aşaması saldırgan için caydırıcı olabilmektedir. Yapılan çalışmada, cihazların lokal ağlar üzerinden bağlantısı incelenmiştir. Bu ağların güvenlik seviyeleri ve saldırıya açıklıkları tartışılmıştır. Sonuç kısmında, bu alanda bilgi güvenliği kapsamında alınabilecek önlemler vurgulanmıştır.
yeni metin1	Teknolojinin ilerlemesi birlikte telekomünikasyon alanında bilgi güvenliğine duyulan gereksinim artmaktadır. İletim ağına bulaşan casus yazılımlar bu alanda en ciddi tehdit kaynağıdır. Bu nedenle aktarılan verinin kriptolu olması oldukça önemlidir. Çünkü şifreli bilgileri çözme aşaması saldırgan için caydırıcı olabilmektedir. Yapılan araştırmada cihazların lokal ağlar aracılığıyla iletişimi incelenmiştir. Bu ağların güvenlik seviyeleri ve saldırıya açıklıkları tartışılmıştır. Sonuç kısmında, bilgi güvenliği çerçevesinde alınabilecek önlemler vurgulanmaktadır .
metin2	Teknolojinin ilerlemesi ile telekomünikasyon alanında bilgi güvenliğine olan gereksinim yükselmektedir. İletim ağına bulaşan casus yazılımlar iletişime ciddi hasarlar vermektedir. Belirtilen sebepten ötürü akan veriyi şifrelemek oldukça önemlidir. Çünkü şifreli bilgileri çözme süreci saldırgan için yıldırıcı bir faktördür. Yapılan araştırmada cihazların yerel ağlar aracılığıyla iletişimi incelenmiştir. Sonuç olarak bilgi güvenliği çerçevesinde çözümler vurgulanmaktadır.

Yapılan deney öncesinde metin1 ve metin2 aralarındaki yapısal benzerlik oranı 0.33 olarak hesaplanmıştır. Bu oran metinlerin sadece yapısal olarak benzerliklerini ifade etmektedir. Fakat metinlerin anlamsal yakınlıkları ölçüldüğünde bu orandan daha yüksek bir sonuç çıkması beklenir. Çünkü iki metinde de birbirine yakın konulardan bahsedilmektedir. Önerilen yöntemin metinler üzerinde kullanımıyla birlikte yapısal benzerlik, anlamsal benzerlikle desteklenecek ve yeni bir benzerlik oranı üretilecektir. Bu işlemlerin ardından elde edilen yeni benzerlik oranı yaklaşık olarak 0.57 seviyesinde ölçülmüştür. Beklendiği üzere anlamsal yakınlıkları bilinen metinlerin benzerlik oranları, yapısal benzerlikten daha yüksek çıkmıştır.

$$Kos_{(m1,m2)} = BO1 = 0.334$$

$$Kos_{(ym1,m2)} = BO2 = 0.569$$

Deneyden elde edilen bulgulara göre, pivot cümlelere ait benzerlik oranlarındaki değişim Tablo 2.8’de gösterilmektedir.

Tablo 2.8. Deney 2 sonucunda benzerlik oranlarının cümle bazlı değişimi

metin1	İlk Benzerlik Oranı (BO1)	Son Benzerlik Oranı (BO2)	Değişim (%)
cümle 1	0.39	0.70	79
cümle 2	0.30	0.60	100
cümle 3	0.24	0.24	0
cümle 4	0.42	0.63	50
cümle 5	0.49	0.87	77
cümle 6	0.00	0.00	0
cümle 7	0.40	0.59	48
total	0.33	0.56	70

Elde edilen bulgular incelendiğinde, metinler arası total benzerliğin ilk orana göre %70 daha yükseldiğini göstermektedir. İki metnin yapısal olarak karşılaştırılması sonucu benzerlikleri 0.33 olarak ölçülmüştür. Önerilen yöntem ile yapılan benzerlik testinde ise sonuç 0.56 olacak şekilde güncellenmiştir. Metinler arasındaki konu ve anlatım benzerliği incelendiğinde, ikinci sonucun daha tutarlı olduğu görülmektedir. Yapılan çalışmayla metinler arasında yapısal incelemeye saptanamayan benzerlikler daha yüksek oranla tespit

edilmiştir. Elde edilen bulguların doğruluğunu kontrol etmek için önerilen kelimeler Tablo 2.9’da sunulmaktadır.

Tablo 2.9. Deney 2 sonucunda algoritma tarafından saptanan yakın kelimeler

Kelimenin metin1’deki Hali	Yöntemin Önerdiği Alternatif
gelişmesiyle	ilerlemesi
haberleşme	telekomünikasyon
ihtiyaç	gereksinim
sızan	bulaşan
zararlı	casus
önemli	ciddi
verileri	bilgileri
deşifreleme	çözme
çalışmada	araştırmada
üzerinden	aracılığıyla
bağlantısı	iletişimi
kapsamında	çerçevesinde
vurgulanmıştır	vurgulamaktadır

Çalışılan yöntemle değişen kelimeler incelendiğinde, tutarsız bir öneri bulunmadığı görülmektedir. Değişen kelimelerin, akan cümlede neredeyse hiç anlam bozukluğuna sebep olmaması, yöntemin bir diğer başarısı olarak gösterilebilir. Elde edilen bulgular kapsamında önerilen yöntemin Bilgi Güvenliği alanında başarılı sonuçlar verdiği gözlemlenmiştir. Daha büyük bir veri seti ile gerçekleştirilecek eğitim ile daha iyi sonuçların alınabileceği öngörülmektedir.

2.7.5. Genel Türkçe Derlem ile Konu Bağımsız Deney 3

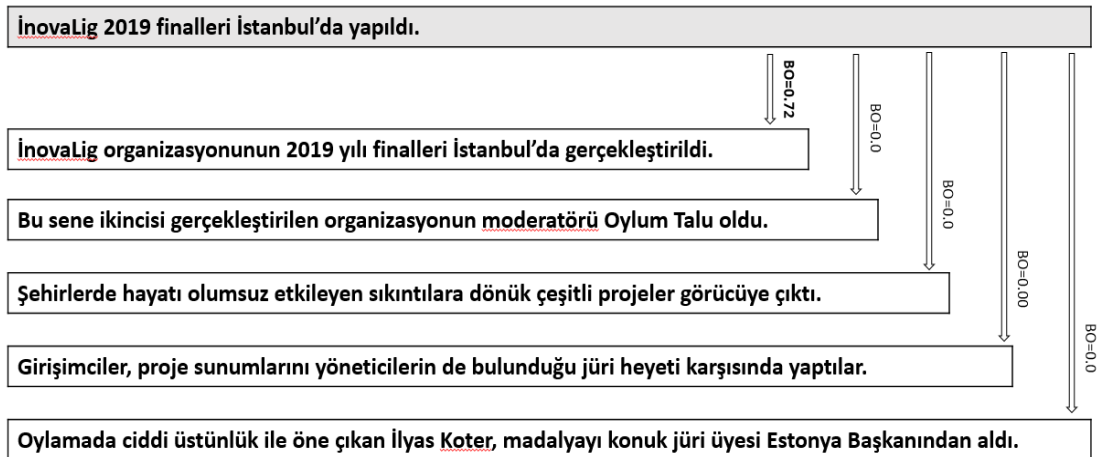
Yapılan deneyde önceki iki deneyden farklı olarak, bir alan üzerine odaklanmadan genel Türkçe metinler için benzerlik tespiti çalışması yapılmıştır. Çalışmada kullanılan derlem de önceki deneylerden farklı olarak haber metinleri ve Wikipedia’dan alınan Türkçe metinlerden oluşturulmuştur [71]. Deney kapsamında oluşturulan derlemde 400.000’den fazla kelime bulunmaktadır. Ayrıca bu deney için modelin önerdiği kelime

sayısı 5'ten 10'a çıkarılmıştır. Böylece modelden alınan öneri sayısı ile elde edilen başarının etkilenme oranı ölçülecektir. Deneyde kullanılmak üzere seçilen metinler Tablo 2.10'da gösterilmektedir.

Tablo 2.10. Deney 3 için seçilen metinler

metin1	İnovaLig 2019 finalleri İstanbul'da yapıldı. Bu yıl ikincisi düzenlenen organizasyonu Oylum Talu yönetti. Kentlerde görülen temel sorunlara yönelik çeşitli projeler görücüye çıktı. Yarışmacılar, projelerini siyasilerin de bulunduğu jüri heyeti karşısında sundular. Oylamada rakiplerine karşı büyük başarı gösteren İlyas Koter, ödülünü misafir jüri üyesi Estonya Başkanından aldı.
metin2	İnovaLig organizasyonunun 2019 yılı finalleri İstanbul'da gerçekleştirildi. Bu sene ikincisi gerçekleştirilen organizasyonun moderatörü Oylum Talu oldu. Şehirlerde hayatı olumsuz etkileyen sıkıntılara dönük çeşitli projeler görücüye çıktı. Girişimciler, proje sunumlarını yöneticilerin de bulunduğu jüri heyeti karşısında yaptılar. Oylamada ciddi üstünlük ile öne çıkan İlyas Koter, madalyayı konuk jüri üyesi Estonya Başkanından aldı.

Deney 3 için hazırlanan metin1 pivot cümlelerin seçileceği pivot metin olarak kullanılacaktır. Bu metinden alınan her bir cümle ikinci metinde geçen cümleler ile karşılaştırılacaktır. Elde edilen bulgular ile konudan bağımsız çalışmanın etkilerini tartışmak mümkün hale gelecektir. İlk adımda birinci metnin ilk cümlesi pivot cümle olarak seçilir ve metin2 ile olan benzerlikleri hesaplanır. Bu hesaplama Şekil 2.32'de gösterilmektedir.



Şekil 2.32. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 1)

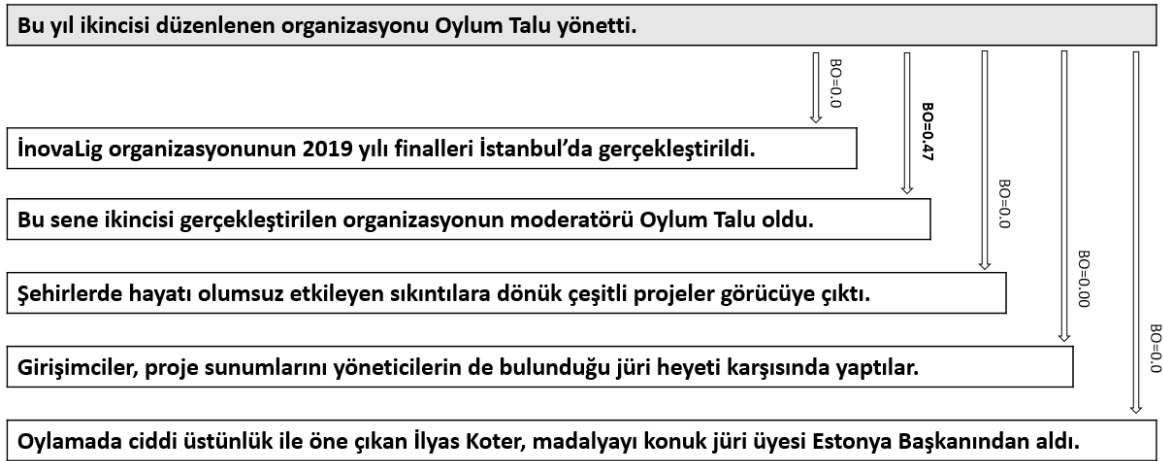
Şekil 2.32’de görüldüğü üzere pivot cümle ile ikinci metnin ilk cümlesi arasında 0.72 oranında yapısal benzerlik bulunmaktadır. Bu ölçüm Kosinüs Benzerlik Algoritması ile gerçekleştirilmiştir. Bu benzerlik oranı, anlamsal yaklaşım ile desteklendiğinde sonucun değişimi nihai benzerlik oranını verecektir. Anlamsal benzerlik önerileri için eğitilen Word2Vec modelinden öneriler alınmaktadır. Pivot cümlelerin kelimeleri için alınan öneriler Şekil 2.33’te gösterilmektedir.

İnovaLig	2019	finalleri	İstanbul’da	yapıldı
				gerçekleştirildi (BO = 0.86)

Şekil 2.33. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 1)

Pivot cümleden sadece bir kelimenin diğer cümle ile farklı olması, mevcut benzerliği daha fazla arttırabilmek için dezavantaj içermektedir. Fakat değişik görülen tek kelimenin de öneri yöntemiyle metin2’deki anlamı bire bir karşıladığı görülmüştür. Pivot cümle için başlangıç benzerlik oranı 0.72 iken, nihai benzerlik oranı 0.86’ya yükselmiştir.

Bu işlem ardından pivot cümle değişecek ve yeni cümle için metin2 ile yapısal benzerlik ölçümleri yapılacaktır. Bu ölçümlerden alınan sonuçlar, Şekil 2.34’te gösterildiği gibidir.



Şekil 2.34. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 2)

Pivot cümle ile en yakın yapısal benzerliğe sahip olan cümle, 0.47 oranında benzerlik göstermektedir. Bu benzerlik oranı, önerilen algoritmanın ikinci adımına geçiş için

yeterlidir. Algoritmanın ikinci adımında gerçekleşen cümle içi değişiklikler Şekil 2.35’te gösterilmektedir.

Bu	yıl	ikincisi	düzenlenen	organizasyonu	Oylum	Talu	yönetti
	sene (BO = 0.60)		gerçekleştirilen (BO = 0.70)				

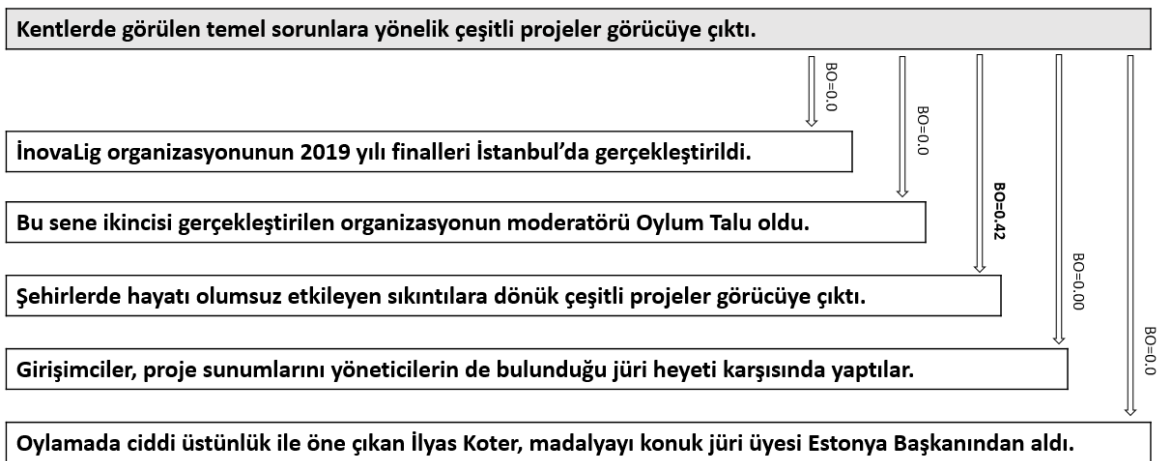
Şekil 2.35. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 2)

Pivot cümle içerisindeki kelimelerden iki tanesi model tarafından önerilen yeni kelimeler ile değiştirilmiştir. Önerilere uygun değişiklikler ile hesaplanan benzerlik oranı 0.47’den 0.70’e yükselmiştir. Model tarafından değişimi uygun görülen kelimeler aşağıda maddeler halinde gösterilmektedir.

- yıl → sene
- düzenlenen → gerçekleştirilen

Yapılan değişiklikler incelendiğinde eş anlamlı ve yakın anlamlı kelimelerin tespit edildiği görülmektedir. Önerilen kelimelerin anlamda değişikliğe neden olmaması, modelin tutarlı sonuçlar verdiğini göstermektedir.

Bir sonraki adımda değişen pivot cümlelerin metin2 ile yapısal benzerlik oranları Şekil 2.36’da gösterilmektedir.



Şekil 2.36. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 3)

Elde edilen yapısal benzerlik oranları incelendiğinde, sadece bir cümle ile 0.42 oranında yapısal benzerlik bulunduğu gözlemlenir. Bu oran algoritmanın ikinci adımına geçmek için yeterli seviyededir. Pivot cümle ile benzer cümle arasındaki anlamsal benzerliğin hesaplanabilmesi için Word2Vec modelinden öneriler alınmıştır. Bu öneriler içerisinde benzerliği tespit eden başarılı sonuçlar Şekil 2.37’de gösterilmektedir.

Kentlerde	görülen	temel	sorunlara	yönelik	çeşitli	projeler	görülmeye	çıkı
Şehirlerde (BO = 0.52)			sıkıntılara (BO = 0.63)	dönük (BO = 0.73)				

Şekil 2.37. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 3)

Sonuçlar incelendiğinde pivot cümle içerisinde üç kelimenin değişikliğe uygun olduğu görülmektedir. Bu sözcüklere alternatif olarak sunulan yeni kelimelerle, doğrudan ikinci metinde geçen kelimeler bulunmuştur. Değişmesi uygun görülen kelimeler maddeler halinde aşağıdaki gibi görülmektedir.

- kentlerde → şehirlerde
- sorunlara → sıkıntılara
- yönelik → dönük

Değişen kelimeler incelendiğinde eş anlamı ve yakın anlamlı eşleşmelerden oluştuğu görülmektedir. Bu değişimler neticesinde, başlangıçta 0.42 olan benzerlik hesabı 0.73 olarak güncellenmiştir. Sıradaki pivot cümlelerin metin2 ile yapısal benzerlik oranları Şekil 2.38’de gösterilmektedir.

Yarışmacılar, projelerini siyasilere de bulunduđu jüri heyeti karşısında sundular.	
InovaLig organizasyonunun 2019 yılı finalleri İstanbul’da gerçekleştirildi.	BO=0.0
Bu sene ikincisi gerçekleştirilen organizasyonun moderatörü Oylum Talu oldu.	BO=0.0
Şehirlerde hayatı olumsuz etkileyen sıkıntılara dönük çeşitli projeler görülmeye çıktı.	BO=0.52
Girişimciler, proje sunumlarını yöneticilerin de bulunduđu jüri heyeti karşısında yaptılar.	BO=0.8
Oylamada ciddi üstünlük ile öne çıkan İlyas Koter, madalyayı konuk jüri üyesi Estonya Başkanından aldı.	

Şekil 2.38. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 4)

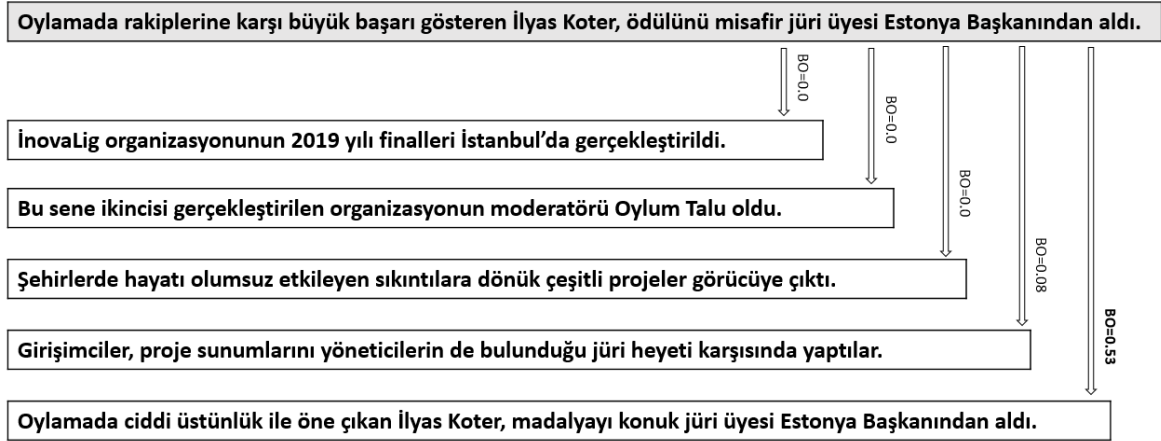
Elde edilen benzerlik oranları incelendiğinde, metin2’den 5. cümle ile pivot cümle arasında 0.52 oranında yapısal benzerlik gözlemlenmektedir. Bu oranı anlamsal benzerlik testleriyle birlikte genel benzerlik oranına dönüştürmek için bir dizi işlem yapılır. Bu süreçte Word2Vec modelinden cümle içindeki tüm kelimeler için alternatifler önermesi beklenir. Pivot cümle için önerilen kelimeler ve bu kelimelerin oransal değişime katkısı Şekil 2.39’da gösterilmiştir.

Yarışmacılar	projelerini	siyasilerin	de	bulunduğu	jüri	heyeti	karşısında	sundular
		yöneticilerin (BO = 0.63)						

Şekil 2.39. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 4)

Pivot cümle için modelden önerilen kelimelerin sadece bir tanesi cümledeki benzerlik oranını arttırmıştır. Modelin “siyasiler” yerine “yöneticiler” kavramını önermesi cümlede anlam değişikliğine yol açmadığı gibi ikinci metinle doğrudan eşleşme sağlamıştır. Yakın anlamlı iki kelimenin saptanmış olması modelin tutarlılığını göstermektedir. Yapılan değişiklik sonucunda benzerlik oranı 0.52 seviyesinden 0.63’e yükselmiştir.

Pivot cümle son değişikliği ile birlikte metin1’e ait son cümle seçilecektir. Bu cümle metin2 ile benzerlik oranları Şekil 2.40’da sunulmaktadır.



Şekil 2.40. Deney 3 için pivot cümle ile metin2 arasındaki benzerlikler (adım 5)

Pivot cümle önceki deneyler ve testlerde görülen cümlelere oranla daha uzun bir cümledir. 15 kelimedenden oluşan bu cümle ile metin2’den en yakın olduğu cümle arasında 0.53 oranında benzerlik bulunmaktadır. Bu oranın güncellenmesi için algoritmanın ikinci

aşamasında kullanılacak Word2Vec modeline ihtiyaç duyulmaktadır. Modelden gelen öneriler Şekil 2.41’de gösterilmektedir.

Oylamada	rakiplerine	karşı	büyük	başarı	gösteren	İlyas	Koter
			ciddi (BO = 0.60)	üstünlük (BO= 0.66)			

ödülünü	misafir	jüri	üyesi	Estonya	Başkanından	aldı
madalyayı (BO= 0.73)	konuk (BO = 0.80)					

Şekil 2.41. Deney 3 için pivot cümlede algoritma tarafından yapılan değişiklikler (adım 5)

Sonuçlar incelendiğinde, pivot cümleden 4 kelime için değişiklik önerilerinin olumlu katkı verdiği gözlemlenmiştir. Bu 4 kelime maddeler halinde aşağıda sunulmaktadır.

- büyük → ciddi
- başarı → üstünlük
- ödülünü → madalyayı
- misafir → konuk

Değişen kelimelerin birbirine oldukça yakın anlamlar taşıdığı görülmektedir. Bu kelimelerin anlamda bir aksamaya sebebiyet vermemesi modelin başarısını gösteren bir etmendir. Kelime değişimleri sonucunda 0.53 olarak bulunan benzerlik oranı 0.80 seviyesine yükselmiştir. Cümlelerin neredeyse aynı anlamı taşıdığı düşünülürse, yapılan güncellemenin sonucu daha doğru seviyeye getirdiği anlaşılmaktadır.

2.7.6. Deney 3 Sonucunda Elde Edilen Bulgular

Yapılan deneyde aynı habere ait iki farklı metin arasındaki benzerlik hesaplanmaktadır. Haberlerin aynı konudan bahsetmesi nedeniyle benzerliğin yüksek olması beklenmektedir. Çalışma sonrasında elde edilen bulguları incelemek için, metinlerin ilk ve son hali Tablo 2.11’de gösterilmektedir.

Tablo 2.11. Deney 3 sonucunda metinlerin ve kelimelerin değişimi

metin1	İnovaLig 2019 finalleri İstanbul'da yapıldı. Bu yıl ikincisi düzenlenen organizasyonu Oylum Talu yönetti. Kentlerde görülen temel sorunlara yönelik çeşitli projeler görücüye çıktı. Yarışmacılar, projelerini siyasilerin de bulunduğu jüri heyeti karşısında sundular. Oylamada rakiplerine karşı büyük başarı gösteren İlyas Koter, ödülünü misafir jüri üyesi Estonya Başkanından aldı.
yeni metin1	İnovaLig 2019 finalleri İstanbul'da gerçekleştirildi . Bu sene ikincisi gerçekleştirilen organizasyonu Oylum Talu yönetti. Şehirlerde görülen temel sıkıntılara dönük çeşitli projeler görücüye çıktı. Yarışmacılar, projelerini yöneticilerin de bulunduğu jüri heyeti karşısında sundular. Oylamada rakiplerine karşı ciddi üstünlük gösteren İlyas Koter, madalyayı konuk jüri üyesi Estonya Başkanından aldı.
metin2	İnovaLig organizasyonunun 2019 yılı finalleri İstanbul'da gerçekleştirildi. Bu sene ikincisi gerçekleştirilen organizasyonun moderatörü Oylum Talu oldu. Şehirlerde hayatı olumsuz etkileyen sıkıntılara dönük çeşitli projeler görücüye çıktı. Girişimciler, proje sunumlarını yöneticilerin de bulunduğu jüri heyeti karşısında yaptılar. Oylamada ciddi üstünlük ile öne çıkan İlyas Koter, madalyayı konuk jüri üyesi Estonya Başkanından aldı.

Yapılan deney öncesinde metin1 ile metin2'ye ait yapısal benzerlik oranı 0.54 olarak bulunmuştur. Bu oran aynı haberden benzer şekilde bahseden iki metin için yeterli değildir. Örnek metinler incelendiğinde anlamların daha yakın olduğu sonucuna varmak mümkündür. Nitekim çalışmaların sonucunda önerilen yöntemin bulduğu yeni benzerlik oranı 0.75 olmuştur. Bu oran hem yapısal hem de anlamsal benzerlik oranlarının hibritlenmesi ile bulunan güncel sonucu göstermektedir.

$$Kos_{(m1,m2)} = BO1 = 0.544$$

$$Kos_{(ym1,m2)} = BO2 = 0.758$$

Deneyden elde edilen bulgulara göre, pivot cümlelere ait benzerlik oranlarındaki değişim Tablo 2.12'de gösterilmektedir.

Tablo 2.12. Deney 3 sonucunda benzerlik oranlarının cümle bazlı değişimi

metin1	İlk Benzerlik Oranı (BO1)	Son Benzerlik Oranı (BO2)	Değişim (%)
cümle 1	0.72	0.86	19
cümle 2	0.47	0.70	49
cümle 3	0.42	0.73	74
cümle 4	0.52	0.63	21
cümle 5	0.53	0.80	51
total	0.54	0.75	39

Elde edilen buğulara göre, iki metin arasındaki benzerlik oranı %39 artacak şekilde hatadan kurtarılmıştır. Yapısal ölçümler ile saptanamayan anlam benzerlikleri total benzerliği 0.54'ten 0.75'e çıkarmıştır. Bu yükselmeye asıl sebep ise, cümlelerin tekil halde güncellenmesidir. Pivot cümleler halinde metin1'e ait her cümle ayrı ayrı önerilen yöntem ile incelenmiş ve güncellenmiştir. Bu işlemler sonucunda anlamsal yakınlığı saptanan kelimeler bütüncül halde Tablo 2.13'te gösterilmektedir.

Tablo 2.13. Deney 3 sonucunda algoritma tarafından saptanan yakın kelimeler

Kelimenin metin1'deki Hali	Yöntemin Önerdiği Alternatif
yapıldı	gerçekleştirildi
sene	yıl
düzenlenen	gerçekleştirilen
şehirlerde	kentlerde
sorunlara	sıkıntılara
yönelik	dönük
siyasilerin	yöneticilerin
büyük	ciddi
başarı	üstünlük
ödülünü	madalyayı
misafir	konuk

Değişen kelimeler incelendiğinde, önerilen yöntemin konudan bağımsız genel Türkçe veriler kullanımında da tutarlılığını koruduğu görülmektedir. Eş anlamlı ve yakın anlamlı

kelime tespitleri ile saptanamayan benzerlikler ortaya çıkarılmıştır. Bu yaklaşım ile Türkçe tüm içeriklerin kıyaslanması mümkündür. Fakat kelimelerin konulara göre alacağı terim anlamlarını yakalamak ve kısaltmaları algılamak, çalışılan derlemin genişliği ile orantılı şekilde mümkün olacaktır.



3. SONUÇLAR VE TARTIŞMA

Yapılan çalışmada, Türkçe metin benzerliklerinin tespitine yönelik ölçüm hatalarını minimuma indirmek üzerine odaklanılmıştır. Bu hataların sebepleri incelenmiş ve mevcut ölçüm yöntemlerinin eksiklikleri saptanmıştır. Yapılan literatür taraması ile birlikte mevcut problemlerin çözümü için yeni bir yöntem önerilmiştir. Önerilen yöntemin çözebileceği mevcut sorunlar maddeler halinde aşağıdaki gibi gösterilmektedir.

- Yapısal benzerlik ölçümü yöntemlerinin, yazım hatalarına karşı çözümsüz kalması,
- Cümle içerisinde terim kullanımı ile gelen anlam benzerliklerinin saptanamaması,
- Cümle içerisinde aynı kelimenin yabancı bir dil ile ifade edilmesinde oluşan benzerliklerin saptanamaması,
- Kısaltmaların kullanımı ile cümlede değişmeyen anlam benzerliğinin saptanamaması,
- Bir kelimenin yerine eş anlamlı veya yakın anlamlı farklı bir kelime kullanılmasıyla oluşan ölçümlerin hatalı olması. Sözlük kullanımıyla çözülen bu sorun için ilgili alana ait sözlük bulunamaması problemi,
- Dilin yapısına göre kelimeye gelen eklerin anlamı değiştirmesiyle oluşan ölçüm hatalarının saptanamaması.

Önerilen yöntem ile maddeler halinde sıralanan problemleri çözmek için çalışmalar yapılmıştır. Yapısal benzerlik ölçümlerinin metin benzerliğinde yetersiz kalması göz önünde bulundurularak, yapısal ve anlamsal benzerlik yöntemlerinin hibrit kullanımı tercih edilmiştir. Bu çerçevede yapısal benzerlik için Kosinüs Benzerliği Algoritması tercih edilmiştir. Anlamsal benzerlik tespiti yapmak için de Derin Öğrenme temelli kelime temsili yöntemi olan Word2Vec yöntemi kullanılmıştır. Bu iki yöntemin ortak kullanımı ile metinlerin hem yapısal hem de anlamsal olarak kıyaslanması mümkün hale gelmiştir.

Önerilen yöntemi test edebilmek için Türkçe metinlerden oluşan bir derlem hazırlanmıştır. Oluşturulan derlem içerisindeki metinlerin tamamı Bilgi Güvenliğine ait bilimsel yayınlardan oluşmaktadır. Word2Vec modelinin daha tutarlı sonuçlar verebilmesi için öncelikli olarak tek alana bağlı eğitilmesi hedeflenmiştir. Böylelikle eğitildiği alan üzerinde anlamsal ilişkileri ve teknik terimleri yakalayabilme özelliği öne çıkmaktadır.

Yapılan deneylerde, önerilen yöntemin hedeflenen mevcut sorunlar üzerine başarılı çözümler ürettiği gözlemlenmiştir. Elde edilen bulgulara göre, yapısal benzerlik ölçümü sonuçlarının %79’unda saptanamayan ek benzerlikler tespit edilmiş ve sonuçlar güncellenmiştir. Güncellenen ölçüm sonuçlarındaki en önemli faktör, eş anlamlı veya yakın anlamlı kelimelerin saptanarak benzerlik oranına dahil edilmesidir. Bu sonuçlara ilaveten Bilgi Güvenliği alanında kullanılan terimler tespit edilmiş ve bu terimlerin Türkçe muadillerinin kullanımı ile atlanan benzerlikler açığa çıkarılmıştır. Konudan bağımsız ikinci bir derlem ile yapılan çalışmalarda, önceki sonuçlara yakın başarıda sonuçlar alınmıştır. Tek farklılık; derlem içeriğinden ötürü kelimelerin sözlük anlamlarına yönelik çıkarım daha yüksek iken terim anlamlı çıkarımlar daha düşük oranda görülmektedir. Önerilen yöntemin hem konu temelli hem de konudan bağımsız genel Türkçe içeriklerde kullanılabilirliği deneyler ile sunulmuştur.

Yapılan deneyler ile önerilen yöntemin intihal tespiti, arama önerisi, metinlerin sınıflandırılması gibi alanlarda kullanımı mümkün görülmektedir. Ayrıca alana özgü sözlük çıkarımı sorununa alternatif bir çözüm sunulmaktadır. Terimler sözlüğü ve eş anlamlılar sözlüğü kullanımına alternatif olarak anlamsal yakınlık vektörleri önerilmektedir.

4. ÖNERİLER

Yapılan çalışmada daha büyük derlemler kullanılması ile olası iyileştirmeler sağlanacağı öngörülmektedir. Daha fazla veri ile eğitilen modelin önerileri gerek kelimeler gerekse kelimelerin aldığı eklerin kattığı anlamlar açısından daha tutarlı olacaktır. Ayrıca alana odaklı eğitim setini bir de genel dil özelliklerini kapsayan derlem ile desteklemek, çalışılan dil için daha tutarlı sonuçlar almak açısından avantaj sağlayacaktır.

Önerilen yöntemin ilk adımında kullanılan Kosinüs Benzerliği Algoritması yazım benzerliklerini (sesteş kelimeler vb.) algılayamamaktadır. Alternatif yaklaşımlar ile yazım benzerliklerini algılayan bir çalışma yöntemin başarısını arttıracaktır.



KAYNAKLAR

- [1] **Adalı, E.**, 2016. Doğal dil işleme, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, **5**, 1-19.
- [2] **Gali, N., Mariescu. R. and Franti. P.**, 2016. Similarity measures for title matching, *International Conference on Pattern Recognition*, Cancun, Mexico, 4-8 December, 1549-1554.
- [3] **Hamming, R. W.**, 1950. Error detecting and error correcting codes, *The Bell System Journal*, **2**, 147-160.
- [4] **Levensthein. V. I.**, 1965. Binary codes capable of correcting deletions, insertions and reversals, *Soviet Physics Doklady*, **8**, 707-710.
- [5] **Żorski, W., Drogosiewicz, B.**, 2018. Real-Time Comparable Phrases Searching Via the Levenshtein Distance with the Use of CUDA Technology. *KKIO Software Engineering Conference*, Gothenburg, Sweden, May-June 2018, 135-150.
- [6] **Jaro M. A.**, 1989. Advances in record linkage methodology as applied to the 1985 census of Tampa Florida, *Journal of the American Statistical Society*, **84**, 414 - 420.
- [7] **Winkler, W. E.**, 1990. String comparator metrics and enhanced decision rules in the Fellegi-Sunter Model of Record Linkage, proceedings of the section on survey research methods, *American Statistical Association*, **16**, 354–359.
- [8] **Shashavali, D., Vishwjeet, V., Kumar, R., Mathur, G., Nihal, N., Mukherjee, S., and Venkanagouda Patil, S.**, 2019. Sentence similarity techniques for short vs variable length text using word embeddings. *Computación y Sistemas*, **23**, 999-1004.
- [9] **Jaccard, P.**, 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull Soc Vaudoise Sci Nat*, **37**, 547-579.
- [10] **Dice, L. R.**, 1945. Measures of the amount of ecologic association between species, *Ecology*, **26** , 297-302.

- [11] **Chaudhary, L., and Singh, B.,** 2019. Community Detection Using Maximizing Modularity and Similarity Measures in social networks, *Smart Systems and IoT: Innovations in Computing: Proceeding of SSIC 2019*, Rajasthan, India, January 2019, 197-206.
- [12] **Kim, D., Seo, D., Cho, S., Kang, P.,** 2019. Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec, *Information Sciences*, **477**, 15-29.
- [13] **Qaiser, S., Ali, R.,** 2018. Text mining: use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, **181**, 25-29.
- [14] **Doan, A. Halevy, A. and Ives, Z.,** 2012. Principles of data integration, Elsevier, Waltham.
- [15] **Monge, E.A and Elkan, C.P.,** 1997. An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records, *Proceeding 2nd ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery, USA*, 23-29.
- [16] **Landauer, T. K., & Dumais, S. T.** 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, **104**, 211-240
- [17] **Miller, G. A.** 1995. WordNet: a lexical database for English, *Communications of the ACM*, **38**, 39-41.
- [18] **Wu, Z., & Palmer, M.,** 1994. Verbs semantics and lexical selection, *Proceedings of the 32nd annual meeting on Association for Computational Linguistics Association for Computational Linguistics*. New Mexico, USA, June 1994, 133-138.
- [19] **Wattiheluw, F. H., Sarno, R.** 2018. Developing Word Sense Disambiguation Corpuses Using Word2vec and Wu Palmer for Disambiguation, *2018 International Seminar on Application for Technology of Information and Communication*, Semarang, Indonesia, September 2018, 244-248.
- [20] **Wan, S., Angryk, R. A.** 2007. Measuring semantic similarity using wordnet-based context vectors, *2007 IEEE International Conference on Systems, Man and Cybernetics*, Montreal, Canada, October 2007, 908-913).

- [21] **Resnik, P.** 1995. Using information content to evaluate semantic similarity in a taxonomy, *Proceedings of IJCAI-95*, Montreal, Canada, August 1997, 448-453.
- [22] **Jiang, J. J., & Conrath, D. W.** 1997. Semantic similarity based on corpus statistics and lexical taxonomy, *Proceedings of 10th Research in Computational Linguistics*, Tapei, Taiwan, August 1997, 19-33.
- [23] **LeCun, Y., Bengio, Y., & Hinton, G.,** 2015. Deep learning. *Nature*, **521**, 436-444.
- [24] **Levy, O., & Goldberg, Y.,** 2014. Dependency-based word embeddings. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, USA, June 2014, 302-308.
- [25] **Mikolov, T., Chen, K., Corrado, G., & Dean, J.** 2013. Efficient estimation of word representations in vector space, *International Conference on Learning Representation*, Arizona, USA, June 2013.
- [26] **Nawang Sari, R. P., Kusumaningrum, R., & Wibowo, A.,** 2019. Word2Vec for Indonesian Sentiment Analysis towards Hotel Reviews: An Evaluation Study, *Procedia Computer Science*, **157**, 360-366.
- [27] **Pennington, J., Socher, R., & Manning, C.,** 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Katar, October 2014, 1532-1543.
- [28] **Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T.** 2017. Enriching word vectors with subword information, *Transactions of the Association for Computational Linguistics*, **5**, 135-146.
- [29] **Yücesoy, B.,** 2007. Türkçe dokümanlar için anlamsal benzerlik hesaplama yöntemi, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [30] **Soğancıoğlu, G.,** 2016. A semantic sentence similarity estimation approach for the biomedical domain, *Yüksek Lisans Tezi*, Boğaziçi Üniversitesi, İstanbul.
- [31] **Amasyalı, M. F., Beken, A.,** 2009. Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi ve Metin Sınıflandırmada Kullanılması, 17. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Kocaeli Üniversitesi, Kocaeli, 9-11 Nisan, 1-4.
- [32] **Kairaldeem, R. A.,** 2014. Calculation of textual similarity using semantic relatedness function, *Yüksek Lisans Tezi*, Çankaya Üniversitesi, Ankara.

- [33] **Başak, S.**, 2009. Türkçe Dokümanların Benzerliği,
<http://www.selcukbasak.com/download/TurkceDokumanBenzerligi.pdf>,
 12.10.2019
- [34] **Kaya, K. M.**, 2015. Plagiarism detection supported by web mining for Turkish textual documents, *Doktora Tezi*, Çukurova Üniversitesi, Adana.
- [35] **Korenčić, D., Ristov, S., Šnajder, J.** 2018. Document-based topic coherence measures for news media text, *Expert Systems with Applications*, **114**, 357-373.
- [36] **Dursun, B., Sonmez, A. C.**, 2008. Türkçe metin benzerlik hesaplamasi için yeni bir yöntem, 16th *Signal Processing, Communication and Applications Conference*, Aydın, April 20-22, 1-4.
- [37] **Nisar, M. S., Waqar, M., & Khan, N.** 2018. Evaluation of string based ontology alignment, *International Journal of Computer Applications*, **975**, 88-87.
- [38] **Álvarez-Carmona, M. A., Franco-Salvador, M., Villatoro-Tello, E., Montes-y-Gómez, M., Rosso, P., Villaseñor-Pineda, L.**, 2018. Semantically-informed distance and similarity measures for paraphrase plagiarism identification, *Journal of Intelligent & Fuzzy Systems*, **34**, 2983-2990.
- [39] **Mihalcea, R., Corley, C., Strapparava, C.**, 2006. Corpus-based and knowledge-based measures of text semantic similarity, *Proceedings of the American Association for Artificial Intelligence*, Boston, USA, July 2006, 775-780.
- [40] **Kiourtis, A., Mavrogiorgou, A., Nifakos, S., Kyriazis, D.**, 2019. A string similarity evaluation for healthcare ontologies alignment to HL7 FHIR resources, In *Intelligent Computing-Proceedings of the Computing Conference*, London, United Kingdom, July 2019, 956-970
- [41] **Mitkov, R.**, 2016. Computational Phraseology light: automatic translation of multiword expressions without translation resources, *Yearbook of Phraseology*, **7**, 149-166.
- [42] **Sun, Y., Ma, L., Wang, S.**, 2015. A comparative evaluation of string similarity metrics for ontology alignment, *Journal of Information & Computational Science*, **12**, 957-964.
- [43] **Sravanthi, P., Srinivasu, B.**, 2017. Semantic similarity between sentences, *International Research Journal of Engineering and Technology (IRJET)*, **4**, 156-161.

- [44] **Yucesoy, B., Oguducu, S. G.**, 2007. Comparison of semantic and single term similarity measures for clustering Turkish documents, *6th International Conference on Machine Learning and Applications (ICMLA)*, Cincinnati, USA, December 2007, 393-398.
- [45] **Erdoğan, A. E.**, 2018. Sosyal medya profilleri arasında benzerlik tespiti ve gösterimi, *Yüksek Lisans Tezi*, TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- [46] **Ozer, Z., Ozer, I., Findik, O.**, 2018. Diacritic restoration of Turkish tweets with word2vec, *Engineering Science and Technology, an International Journal*, **21**, 1120-1127.
- [47] **Acı, Ç. İ., ÇIRAK, A.**, 2019. Türkçe haber metinlerinin konvolüsyonel sinir ağları ve Word2Vec kullanılarak sınıflandırılması, *Bilişim Teknolojileri Dergisi*, **12**, 219-228.
- [48] **Öztürk, M. B., Can, B.**, 2016. Clustering word roots syntactically, *24th Signal Processing and Communication Application Conference (SIU)*, Zonguldak, 16-19 Mayıs, 1461-1464.
- [49] **Ayata, D., Saraçlar, M., Özgür, A.**, 2017. Turkish tweet sentiment analysis with word embedding and machine learning, *25th Signal Processing and Communications Applications Conference (SIU)*, Antalya, 15-18 Mayıs, 1-4.
- [50] **Şahin, G.**, 2017. Turkish document classification based on Word2Vec and SVM classifier. *25th Signal Processing and Communications Applications Conference (SIU)*, Antalya, 15-18 Mayıs, 1-4.
- [51] **Bilgin, M., Köktaş, H.**, 2019. Sentiment analysis with term weighting and word vectors, *International Arab Journal of Information Technology*, **16**, 953-959.
- [52] **Karasoy, O., Ballı, S.**, 2017. Classification Turkish SMS with deep learning tool Word2Vec. *International Conference on Computer Science and Engineering (UBMK)*, Antalya, 5-8 Ekim, 294-297.
- [53] **Arabacı, M. A., Esen, E., Atar, M. S., Yılmaz, E., Kaltahoğlu, B.**, 2018. Detecting similar sentences using word embedding, *26th Signal Processing and Communications Applications Conference (SIU)*, İzmir, Mayıs 2018, s. 1-4.

- [54] **Shahzad, K., Kanwal, S., Malik, K., Aslam, F., Ali, M.,** 2019. A word-embedding-based approach for accurate identification of corresponding activities, *Computers & Electrical Engineering*, **78**, 218-229.
- [55] **Hu, K., Luo, Q., Qi, K., Yang, S., Mao, J., Fu, X., Zhu, Q.,** 2019. Understanding the topic evolution of scientific literatures like an evolving city: Using Google Word2Vec model and spatial autocorrelation analysis, *Information Processing & Management*, **56**, 1185-1203.
- [56] **Dillenberger, J.,** 2019. Evaluation of Model and Hyperparameter Choices in word2vec, *Bachelor's Thesis*, Institute for Web Sciences and Technology, Koblenz, Almany.
- [57] **Orkphol, K., & Yang, W.** 2019. Word Sense Disambiguation Using Cosine Similarity Collaborates with Word2vec and WordNet, *Future Internet*, **11**, 114.
- [58] **Lauren, P., Qu, G., Yang, J., Watta, P., Huang, G. B., & Lendasse, A.,** 2018. Generating word embeddings from an extreme learning machine for sentiment analysis and sequence labeling tasks, *Cognitive Computation*, **10**, 625-638.
- [59] **Zhu, Y., Yan, E., & Wang, F.,** 2017. Semantic relatedness and similarity of biomedical terms: examining the effects of recency, size, and section of biomedical publications on the performance of word2vec, *BMC medical informatics and decision making*, **17**, 95.
- [60] **Chen, Z., He, Z., Liu, X., & Bian, J.** 2018. Evaluating semantic relations in neural word embeddings with biomedical and general domain knowledge bases, *BMC medical informatics and decision making*, **18**, 65.
- [61] **Yan F, Fan Q, Lu M.,** 2018. Improving semantic similarity retrieval with word embeddings. *Concurrency Computat Pract Exper.*, **30**, 1-6.
- [62] **Sitikhu, P., Pahi, K., Thapa, P., Shakya, S.,** 2019. A Comparison of Semantic Similarity Methods for Maximum Human Interpretability, *arXiv preprint arXiv:1910.09129*, 1-4.
- [63] **Raineri, S., Debras, C.,** 2019. Corpora and representativeness corpora and representativeness: Where to go from now?, *hal-univ-paris10.archives-ouvertes.fr*, **19**, 1-10.

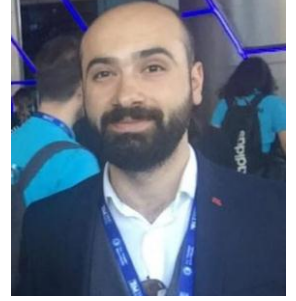
- [64] **Roy, A., Cruz, R. M., Sabourin, R., Cavalcanti, G. D.,** 2018. A study on combining dynamic selection and data preprocessing for imbalance learning, *Neurocomputing*, **286**, 179-192.
- [65] **Aro, T. O., Dada, F., Balogun, A. O., Oluwasogo, S. A.,** 2019. Stop words removal on textual data classification, *International Journal of Information Processing and Communication*, **7**, 1-9.
- [66] **Saif, H., Fernández, M., He, Y., Alani, H.,** 2014. On stopwords, filtering and data sparsity for sentiment analysis of twitter, *Proceeding of 9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland, May 2014, 810-817.
- [67] **Luo, C., Zhan, J., Xue, X., Wang, L., Ren, R., Yang, Q.** 2018. Cosine normalization: Using cosine similarity instead of dot product in neural networks, *International Conference on Artificial Neural Networks*, Rhodes, Greece, October 2018, 382-391.
- [68] **Xiang, L., Yang, X., Zhang, J., & Wang, W.,** 2019. A word-frequency-preserving steganographic method based on synonym substitution, *International Journal of Computational Science and Engineering*, **19**, 132-139.
- [69] **Wu, L., Yen, I. E., Xu, K., Xu, F., Balakrishnan, A., Chen, P. Y., Witbrock, M. J.,** 2018. Word mover's embedding: from Word2Vec to document embedding, *arXiv preprint arXiv:1811.01713*, 1-15.
- [70] **Masum, E., Samet, R.,** 2018. Mobil BOTNET ile DDoS Saldırısı, *International Journal of InformaticsTechnologies*, **11**, 111-121.
- [71] Haber metinleri derlemi,
<https://github.com/ahmetax/derlemtr/tree/master/buyukveri>, 1.11.2019

ÖZGEÇMİŞ

İrfan AYGÜN

Doğum Yeri - Tarihi : ELAZIĞ - 06.11.1990

Uyruk : T.C.



ÖĞRENİM DURUMU

Derece	Üniversite	Bölüm	Yıllar
Lisans	Gazi Üniversitesi	Bilgisayar Mühendisliği	2013
Y. Lisans	Fırat Üniversitesi	Bilgisayar Mühendisliği	Halen

İLETİŞİM BİLGİLERİ

Adres	Manisa Celal Bayar Üniversitesi – Hasan Ferdi Turgutlu Teknoloji Fakültesi
e-posta	irfan.aygun@cbu.edu.tr

ÇALIŞTIĞI KURUMLAR

Kurum Adı	Görev Unvanı	Yıllar
Manisa Celal Bayar Üniversitesi	Araştırma Görevlisi	2018 - Halen
Aygün Yazılım Teknoloji ve Arge Ltd. Şti.	Kurucu / Genel Müdür	2015 / 2018

ÖZGÜN YAYINLARI

Ulusal hakemli dergilerde yayımlanan makaleler

1	Mansur Alp Toçoğlu, Azer Çelikten, İrfan Aygün , Adil Alpkocak. (2019) Türkçe Metinlerde Duygu Analizi İçin Farklı Makine Öğrenmesi Yöntemlerinin Karşılaştırılması, DEUFMD, 21(63), 719-725.
---	--