DERİN ÖĞRENME KULLANILARAK İNGİLİZCE – TÜRKÇE ÇEVİRİLER İÇİN CÜMLE EŞLEME SİSTEMİ

Emir KIZILIRMAK
20 14 50 202

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Doktora Programı
Danışman: Dr. Öğr. Üyesi Erdal GÜVENOĞLU

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Eylül, 2023

DERİN ÖĞRENME KULLANILARAK İNGİLİZCE – TÜRKÇE ÇEVİRİLER İÇİN CÜMLE EŞLEME SİSTEMİ

Emir KIZILIRMAK
20 14 50 202
ORCID: 0000-0001-8518-5152

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim
Bilgisayar Mühendisliği Doktora Programı
Danışman: Dr. Öğr. Üyesi Erdal GÜVENOĞLU
İkinci Danışman: Doç. Dr. Volkan TUNALI

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Eylül, 2023



JÜRİ VE ENSTİTÜ ONAYI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli "Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge" ile bildirilen 6698 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

ETİK İLKE VE KURALLARA UYUM BEYANI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli "Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge" ile bildirilen 6698 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

TEŞEKKÜR

Doktora eğitimim boyunca bana desteğini ve zamanını her türlü problemimi çözmek için ayıran değerli danışman hocam Dr. Öğr. Üyesi Erdal GÜVENOĞLU'na ve ikinci danışman hocam Doç. Dr. Volkan TUNALI'ya sonsuz teşekkürlerimi sunarım.

Emir KIZILIRMAK

Eylül, 2023

ÖZET

DERİN ÖĞRENME KULLANILARAK İNGİLİZCE – TÜRKÇE ÇEVİRİLER İÇİN CÜMLE EŞLEME SİSTEMİ

Emir Kızılırmak
Doktora Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Doktora Programı
Danışman: Dr. Öğr. Üyesi Erdal Güvenoğlu
Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü, 2023

Doğal dil işleme, yapay zekânın ve dil biliminin gelişimiyle son yıllarda önem kazandı. Çeviri şirketlerinin yapmış olduğu, "Türkçe'den İngilizce'ye - İngilizce'den Türkçe'ye" birebir çevirileri veya web site çevirilerinde cümlelerin sırasının değişmesi veya rastgele sıralanması sonucunda, cümleleri tekrar düzenlemek için insan gücüne ihtiyaç duyulmaktadır. Doğal dil işleme alanında, cümle eşleştirme işlemiyle ilgili çeşitli algoritmalar vardır. Bu çalışmada, makine öğrenmesinin alt dalı olan derin öğrenme kullanılarak İngilizce - Türkçe çeviri metinler için efektif eşleştirme yöntemleri bulunmaya çalışıldı. Geliştirilen bir derin öğrenme yöntemiyle internetten çeşitli kaynaklardan veri kümeleri toplandıktan sonra belirli filtrelerden geçmesiyle birlikte eleme ve düzenleme işlemi yapıldı. Mevcut modellerin başarısı karşılaştırıldıktan sonra toplanan veri setinin eğitimiyle mevcut başarılı modellerden daha başarılı bir model elde edilmesiyle birlikte kosinüs benzerliği ile karşılaştırma yapıldı.

Anahtar Sözcükler: Derin Öğrenme, Doğal Dil İşleme, Cümle Eşleme Sistemi

ABSTRACT

SENTENCE ALIGNMENT FOR ENGLISH TO TURKISH TRANSLATIONS BY USING DEEP LEARNING SYSTEM

Emir Kızılırmak
PhD Thesis
Department of Computer Engineering
Computer Engineering Doctorate Program
Thesis Advisor: Assist. Prof. Erdal Güvenoğlu
Maltepe University Graduate School, 2023

Natural language processing has gained importance in recent years with the development of artificial intelligence and linguistics. In “Turkish to English - English to Turkish” one-to-one translations by translation companies or in website translations, the order of sentences is changed or randomized, requiring manpower to rearrange the sentences. In the field of natural language processing, there are various algorithms for sentence alignment. In this study, deep learning, a sub-branch of machine learning, is used to find effective matching methods for English-Turkish translated texts. With a deep learning method developed, datasets were collected from various sources from the internet and then sifted and organized through certain filters. After comparing the success of existing models, a more successful model was obtained by training the collected dataset and a comparison was made with cosine similarity.

Keywords: Deep Learning, Natural Language Processing, Sentence Alignment

İÇİNDEKİLER

JÜRİ VE ENSTİTÜ ONAYI ii
ETİK İLKE VE KURALLARA UYUM BEYANI iii
TEŞEKKÜR iv
ÖZET v
ABSTRACT vi
İÇİNDEKİLER vii
TABLOLAR LİSTESİ ix
ŞEKİLLER LİSTESİ x
KISALTMALAR xi
SEMBOLLER LİSTESİ xii
1. GİRİŞ 1
1.1 Problem 1
1.2 Amaç 2
1.3 Önem 2
1.4 Varsayımlar 3
1.5 Sınırlıklar 3
1.6 Tanımlar 3
2. LİTERATÜR ÇALIŞMASI 4
3. MATERYAL VE YÖNTEM 10
3.1 Doğal Dil İşleme 11
3.2 Doğal Dil İşlemenin Bileşenleri 12
3.2.1 Doğal dil üretimi 13
3.2.2 Doğal dil anlayışı 13

3.3 Doğal Dil İşleme Kullanım Alanları 13
3.4 Cümle Eşleme 14
3.5 Evrensel Cümle Kodlayıcı 16
3.6 BERT 17
3.7 Sentence-BERT 17
3.8 Derin Öğrenme 18
3.9 Kosinüs Benzerliği 19
3.10 Python Programlama 21
3.10.1 Numpy kütüphanesi 23
3.10.2 Scipy kütüphanesi 23
3.10.3 Datasets kütüphanesi 23
3.10.4 Sentence Transformers kütüphanesi 24
3.10.5 PyTorch kütüphanesi 24
3.11 Önerilen Yöntem 24
3.12 Birden Fazla Dile Sahip Modeller 25
3.13 Hazır Modellerin Kullanılması 25
3.14 Oluşturulacak Model İçerisinde Hazır Veri Seti Kullanılması 28
3.15 Oluşturulan Model İçerisindeki Veri Setinin Kullanılması 33
3.16 Modelin Geliştirilmesi 35
3.17 Cümle Eşleme 43
4. BULGULAR VE TARTIŞMA 47
5. SONUÇ ve ÖNERİLER 51
KAYNAKLAR 53
ÖZGEÇMİŞ 62

TABLOLAR LİSTESİ

Tablo 1. Hazır Modellerin Başarıları 28
Tablo 2. ISO 639 Dil Kodları..... 29
Tablo 3. Oluşturulacak Modelin Cümle ve Kelime Sayısı 33
Tablo 4. Sentence Transformers – Transformers Metodu 36
Tablo 5. Sentence Transformers – Pooling Metodu..... 37
Tablo 6. Sentence Transformers – Fit Metodu..... 42
Tablo 7. Cümle Sayısı ve Kelime Sayısı..... 44
Tablo 8. Kategori Başlığı ve Cümle Sayısı..... 49
Tablo 9. Doğru Eşleştirme Olmayan Kategori Başlığı ve Cümle Sayısı 50

ŞEKİLLER LİSTESİ

Şekil 1. Doğal Dil İşlemenin Yıllara Dağılımı 12
Şekil 2. Doğal Dil İşlemenin Bileşenleri 12
Şekil 3. Cümle Eşleştirme 15
Şekil 4. Evrensel Cümle Kodlayıcıya Genel Bakış 16
Şekil 5. SBERT Mimarisi 18
Şekil 6. Derin Öğrenme 19
Şekil 7. Olası Kosinüs Benzerliği Durumları 21
Şekil 8. Python Basit Mimarisi 22
Şekil 9. Python Kütüphane Yüklenmesi 22
Şekil 10. TED Konuşmalarının Kullanılması 29
Şekil 11. TED Konuşmalarının Dilinin İndekslerinin Tespit Edilmesi 30
Şekil 12. TED Konuşmalarının Tespit Edilen Örnek Cümleleri 31
Şekil 13. TED Konuşmalarının Çeviriler 31
Şekil 14. TED Konuşmalarının İngilizce ve Türkçe Örnekleri 32
Şekil 15. TED Konuşmalarının Kullanılması 32
Şekil 16. Eğitilecek Verinin Hazırlanması 34
Şekil 17. Modelin Geliştirilmesi 35
Şekil 18. Sentence Transformers – Öğretici ve Öğrenci Modeli 38
Şekil 19. Sentence Transformers 38
Şekil 20. Sentence Transformers – Paralel Veri Seti 39
Şekil 21. Sentence Transformers – Ayarlar 39
Şekil 22. Sentence Transformers – Veri Setinin Yüklenmesi 39
Şekil 23. Sentence Transformers – Veri Yükleyicisi 40
Şekil 24. Karesel Ortalama Hata Hesaplaması 41
Şekil 25. Sentence Transformers – Karasel Ortalama Hata Hesaplama Metodu 41
Şekil 26. Sentence Transformers – Fit 43
Şekil 27. Cümlelerin Vektör Karşılıklarının Bulunması 45
Şekil 28. Cümle Eşleştirme Başarısı 48

KISALTMALAR

AI	:	Yapay Zekâ	Artificial Intelligence
ALGOL	:	Algoritmik Dil	Algorithmic Language
AMP	:	Otomatik Karışık Hassasiyet	Automatic Mixed Precision
BERT	:	Transformatörlerden Çift Yönlü Kodlayıcı Gösterimleri	Bidirectional Encoder Representations from Transformers
CPU	:	Merkezi İşlem Birimi	Central Processing Unit
DDA	:	Doğal Dil Anlama	Natural Language Understanding
DDÜ	:	Doğal Dil Üretme	Natural Language Generation
DOA	:	Derin Ortalama Ağı	Deep Averaging Network
GPU	:	Grafik İşlemci Ünitesi	Graphics Processing Unit
GZIP	:	GNU Zip	GNU Zip
ISO	:	Uluslararası Standartlar Örgütü	International Organization for Standardization
KAB		Kelime Anlam Belirginleştirme	Word Sense Disambiguation
ML	:	Makine Öğrenmesi	Machine Learning
MSE	:	Karesel Ortalama Hata	Mean Squared Error
NLP	:	Doğal Dil İşleme	Natural Language Process
SBERT	:	Sentence-BERT	Sentence-BERT
TED	:	Teknoloji, Eğlence, Tasarım	Technology, Entertainment, Design

SEMBOLLER LİSTESİ

$\mathbf{A_i}$, $\mathbf{B_i}$	:	Vektör Bileşenleri
N	:	Öğretici Model
Ñ	:	Öğrenci Model
n	:	Vektör Bileşen Sayısı
$\mathbf{s_n}$	:	Kaynak Dosyadaki Dil Çevirileri
$\mathbf{t_n}$	:	Hedef Dosyadaki Dil Çevirileri

1. GİRİŞ

1.1 Problem

Dil, gönüllü olarak üretilen semboller aracılığıyla fikirleri, duyguları ve arzuları iletmek için tamamen insani ve içgüdüsel olmayan bir yöntemdir (Sapir, 1921). Dilbilim, dilin bilimsel veya sistematik olarak incelenmesi olarak tanımlanabilir. İnsan dillerinin kurallarını, sistemlerini ve ilkelerini bilimsel olarak incelemesi anlamında bir bilimdir (Hamawand, 2020). Doğal dil işleme, hem bir dizi teoriye hem de bir dizi teknolojiye dayanan metin analizine yönelik bilgisayarlı bir yaklaşımdır (Liddy, 2001). Doğal dil işlemenin bir araştırma alanı da cümle eşleştirme işlemidir. Cümle eşleştirme işlemi tanım olarak, iki farklı dilde birebir çevirisi olan çeviri şirketleri metinleri, akademik çalışmaları, haber sitelerinde yayınlanan makaleler, e-kitaplar, film alt yazıları gibi metinler üzerinde gerçekleşmektedir.

Birebir çevirisi olan fakat kaynak metin ile hedef metin arasında aynı sırada olmayan veya herhangi bir şekilde sıralaması karışmış cümleler arasında eşleme işlemi gerçekleştirilmektedir. İki farklı dil arasında sıralaması değişen cümleleri tekrar aynı sıralamaya getirmek için insan gücüne ihtiyaç duyulmaktadır. İnsan gücüyle cümle eşleştirme işlemi yapmak hem yorucu hem de çok zaman kaybettirici olmaktadır.

Cümle eşleştirme işlemi için literatürde yapılan çalışmalar olmasına rağmen İngilizce ve Türkçe cümleler arasında çok fazla kaynak bulunmamaktadır. Sebep olarak ise İngilizce'nin dil yapısı Türkçe'den çok farklı olmasından kaynaklanmaktadır. İngilizce analitik bir dil ve neredeyse hiçbir morfolojik özelliği bulunmamaktadır. Türkçe'de ise birden fazla eki bir araya getirilmesiyle birlikte her ekin de kendine özgü bir anlamı bulunmaktadır.

Cümle eşleştirme işlemi için literatürde yapılan çalışmalarda klasik olarak bilinen cümle eşleştirme yöntemleri ise uzunluk tabanlı eşleştirme, yer tabanlı eşleştirme ve sözlüğe dayalı eşleştirme olmak üzere üç farklı başlık altında incelenmektedir. İlk klasik cümle eşleştirme yöntemi olarak uzunluk tabanlı eşleştirme, bu yöntem ile birebir çevirisi yapılmış paralel iki metin arasında cümle uzunluklarını istatistiksel yöntemler ile

karşılaştırma yaptıktan sonra cümle eşleştirme işlemi yapmaktadır. İkinci klasik cümle eşleştirme yöntemi ise yer tabanlı eşleştirme, bu yöntem ile birebir çevirisi yapılmış paralel iki metin içerisindeki yeri ile ilgili istatistiksel işlemler yaptıktan sonra cümle eşleştirme işlemi yapmaktadır. Son klasik cümle eşleştirme yöntemi ise sözlüğe dayalı eşleştirme, bu yöntem ile paralel iki metin arasındaki kelimelerin sözlük ile karşılaştırma işlemi yapıldıktan sonra cümle eşleştirme işlemi yapmaktadır.

Gelişen teknolojiyle birlikte makine öğrenmesinin alt dalı olan derin öğrenme yöntemleri de gelişmektedir. Bu sebepten dolayı doğal dil işleme alanında birçok gelişme olmuştur. Bu gelişmelerin en başında BERT, RoBERTa ve orijinal dönüştürücüler kelime seviyesinde hesaplamaları gerçekleştirerek birçok amaca hizmet etmektedir.

BERT algoritmasıyla ilgili bir örnek vermek gerekirse "n" adet cümle içerisinde anlam açısından birbirlerine en yakın olan ilk "k" adet cümle çiftini hesaplamaktadır. Buradaki "n" adet cümle sayısı arttıkça BERT'in hesaplama konusunda yetersiz kalabilmektedir. Bu noktada SBERT yani Sentence-BERT ile daha başarılı sonuçlar elde edilebilmektedir. Çünkü BERT ağı değiştirilerek çok sayıda cümle için eğitilmesi gerçekleştirilmiştir (Gungor, 2006; Kızılırmak, 2020; Reimers ve Gurevych, 2019).

1.2 Amaç

Bu tez çalışmasında, birebir çevirisi olan fakat aynı sırada olmayan İngilizce – Türkçe cümleler arasında cümle eşleştirme işlemi problemine, insan gücüne ihtiyaç duyulmadan veya çok az bir insan gücüyle daha fazla başarılı sağlanması amaç olarak belirlenmiştir.

Teknolojinin gelişmesiyle birlikte birçok doğal dil işleme problemlerinin çözümlerinin önü açılmıştır. Çeşitli nedenler yüzünden sıralaması karışan birebir çevirisi olan cümleleri tekrar hizalamak ve en başta insan gücü kullanmadan veya insan gücünü en aza indirmesi amaçlanmıştır.

1.3 Önem

Cümle eşleştirme işlemi için İngilizce ve Türkçe dilleri kendi aralarında morfolojik olarak farklı yapılarda bulunmaktadır. Bu sebepten dolayı literatür içerisinde İngilizce ve Türkçe dilleri arasında literatürde çok fazla çalışma bulunmamaktadır. Bu çalışmayla birlikte

İngilizce ve Türkçe dilleri arasında farklı çalışmalara kaynak olarak kullanılmak üzere yeni geliştirilen bu modelin kullanılması için önem kazanmıştır.

1.4 Varsayımlar

Tekrar cümle eşleştirme işlemine ihtiyaç olmadan zaten olması gereken sırada olan cümleler bulunmaktadır. Bu cümleler için tekrar eşleştirme işlemi gerçekleştirilmektedir. Bu işlemle birlikte mevcutta doğru sırada olan bir cümleyi tekrar cümle eşleştirme işlemi yaptıktan sonra doğru olmayan bir eşleştirme yapılmasının yanı sıra bu işlem için süre harcanmaktadır.

1.5 Sınırlıklar

Cümle eşleştirme işlemini derin öğrenme yardımıyla gerçekleştirmek için bir dil modeline ihtiyaç bulunmaktadır. İlgili modelin eğitim süreci için birebir çevirisi yapılmış cümlelerin toplanması gerekmektedir. Birebir çevirisi yapılmış cümleleri toplamak için hazır veri seti ve oluşturulan veri setinin bir araya getirilmesi gerekmektedir. Veri seti oluşturulduktan sonra eğitim işlemi sonrasında ilgili model oluşturulmaktadır. Bu model oluşturulması için eğitilecek olan veri setindeki cümlelerin kategorileri, cümle sayısı ve cümle uzunlukları sınırlandırılmıştır.

Cümle eşleştirme sürecinde modelin eğitilmesi aşamasında cümle uzunluklarının belirlenmesi gerekmektedir. Belirlenen uzunluktan fazla ise eğitim sürecine dahil olmayacaktır.

1.6 Tanımlar

Kelime gösterimleri, kelimelerin uzay denkleminde vektör karşılıklarına denk gelmektedir. Örneğin, uzay denklemine göre kadın ve kralın vektör karşılıkları toplamının kelime karşılığı kraliçedir.

2. LİTERATÜR ÇALIŞMASI

Cümle eşleştirme işlemi için literatürde olan çalışmaları dilden bağımsız bir şekilde incelenmiştir. Bu inceleme sonucunda uzunluk tabanlı eştirme, yer tabanlı eşleştirme ve sözlüğe dayalı eşleştirme olmak üzere üç farklı ana başlık altında incelenmektedir. Fakat teknolojinin yıllar içerisinde gelişmesiyle birlikte günümüzde çeşitli derin öğrenme yöntemleri kullanılarak cümle eşleştirme işlemleri gerçekleştirilebilmektedir. İlgili çalışmaların yıllara göre sıralanmış bir şekilde yapılan çalışmaların sıralanması detaylarıyla birlikte ifade edilmektedir. Cümle eşleme işlemi için yapılan çalışmalar birçok çalışmayı da beraberinde getirmektedir. Literatür de ilk olarak cümle eşleşme süreçlerine 1988 yılında Kay ve Roscheisen tarafından yapılan bir çalışmayla birlikte literatürdeki ilk cümle eşleştirme işlemi özelliğini taşımaktadır.

1988 yılında Kay ve Roscheisen tarafından yapılan bir çalışmada cümle eşleme sırasında bir metindeki kelimenin diğer metindeki hangi kelimeye karşılık geldiği fikrine dayanır ve esas olarak dağılımların benzerliklerini karşılaştırılmaktadır. Cümle eşleştirme işleminin doğru olabilmesi için kelime seviyesinde eşleşme işlemi yapmıştır (Kay ve Roscheisen, 1993).

1991 yılında Peter F. Brown ve arkadaşları tarafından yapılan bir çalışmada cümle hizalama işlemi yaparken cümle içerisindeki kelimeleri dikkate alarak hizalama işlemini yapmıştır. Fakat bu tür algoritmalar yavaş ve son derece hassas hizalama sistemi olmasına rağmen çok büyük metin içeren veri setler için uygun olmayabilir. Geliştirilen algoritma kelimelerin sözlüksel ayrıntıları ile ilgilenmez sadece her cümledeki sözcük sayısı ile ilgilenmektedir (Brown, Lai ve Mercer, 1991a).

1993 yılında William A. Gale ve Kenneth W. Church tarafından yapılan bir çalışmada cümle hizalama işlemi bir dildeki cümlelerin diğer dilde karşılıklarını belirlemek anlamına gelmektedir. Bu geliştirilen algoritmada karakter uzunluklarının basit bir istatistiksel modeline dayalı olarak cümle hizalama işlemi gerçekleştirilecek bir algoritma geliştirmiştir. Algoritma uzun cümlelerin diğer dilde karşılığı olarak uzun cümlelere, kısa cümlelerin ise diğer dildeki kısa cümlelere karşılığı olduğuna göre hareket etmektedir. İki cümlenin karakter sayılarındaki cümle uyumuna göre olası bir eşleşme puanı

atanmaktadır. Bu olasılık puanı cümlelerin maksimum benzerlik uyumunu bulabilmek için dinamik bir şekilde kullanılmaktadır. İsviçre Birliği Bankası (UBS, Swiss Union Bank) tarafından İngilizce, Fransızca ve Almanca olarak yayınlanan üç dilli ekonomik raporlar üzerinde bu yöntem denendiğinde %4'ü hariç hepsi doğru bir şekilde hizalandığı gözlemlenmiştir. Hizalamaların en iyi puan alan %80'i seçildiğinde ise hata oranı %4'ten %0.7'ye düşmektedir. İngilizce - Fransızca metinlerin çevirisindeki hata oranları İngilizce - Almanca metinlerdeki hata oranına göre daha fazla olduğu gözlemlenmiştir (Gale ve Church, 1993a).

1993 yılında Stanley F. Chen tarafından yapılan bir çalışmada mevcut cümle eşleme algoritmaları kelime kimliklerini yok saymıştır ve sadece cümle uzunluklarını dikkate almıştır (Brown ve diğerleri, 1991a; Gale ve Church, 1993a).

Geliştirilen bu algoritma ile hizalama işlemi sırasında basit bir istatiksel kelimeden kelimeye çeviri modeli oluşturmuştur. Canadian Hansard verilerine göre yaklaşık %0.4 gibi bir hata oranı ile cümle eşleştirme işlemi gerçekleşmiştir. Bu sonuç bundan önceki çalışmaların sonuçlarına göre önemli bir gelişmedir ve algoritmanın dilden bağımsız olmasıdır (Chen, 1993).

1993 yılında Ido Dagan, Kenneth Church ve Willian Gale tarafından yapılan bir çalışmada cümle hizalama işlemi için char_align'ın (Gale ve Church, 1993b) çıktısıyla birlikte Brown el al.'ın Model ikisiyle (Brown, Lai ve Mercer, 1991b) birlikte yapılan geliştirmeler sonucunda word_align algoritmasıyla birlikte İngilizce - Fransızca cümleler hizalamıştır. Bu hizalanan cümleler Canadian Hansard tarafından 3.000.000 adet cümle temin etmiştir. Bu cümlelerin 1.000.000 cümlesi Hansard davaları ve geriye kalan 2.000.000 adet cümle ise Avrupa Ekonomi Topluluğundan gelen davaları içeren bir veri setiyle uyumlu hale getirmiştir. İlk önce 100 adet hizalanmış cümle çifti üzerinden eğitim önyükleme modeli oluşturmuş. Sonrasında ise veri setinin 20.000 cümlesi eğitilmiştir. Brown hizalama yöntemiyle hizalama işlemi yapmıştır. İki hizalama yönteminin hata oranlarını karşılaştırmak için 500'den fazla cümle incelemiştir. Brown algoritması ile hizalama yöntemi %0.6 iken, geliştirilen algoritma ile hata oranı %0.4 olarak gözlemlenmiştir (Dagan, Church ve Gale, 1993).

1994 yılında Dekai Wu tarafından yapılan bir çalışmada Gale and Church'un (1993) uzunluk tabanlı cümle eşleştirme sistemini İngilizce – Çince'ye cümleleri eşleştirmek için uygun bir şekilde değiştirildikten sonra ilgili cümlelerin alanına özgü sözlük ile çeviri desteğini dahil edilmesiyle birlikte geliştirilmiş olan istatiksel bir yöntemdir. Yaklaşık olarak 3.5 megabaytlık İngilizce ve Çince metinlerin hizalaması gerçekleşmiştir. Bire bir olarak sınıflandırılan 10.423 çift cümle içermektedir. Bu cümlelerin içerisinde 217 binden fazla İngilizce kelime yaklaşık olarak 1.269 megabayta karşılık gelen Çince metin yaklaşık 0.659 megabaytı temsil etmektedir. Hint ve Avrupa'daki paralel metinler ile tamamen farklı olmayan Hint ve Avrupa dışı diller arasında yüksek doğruluk sağlayan bir hizalama gösterimi gerçekleştirmiştir (Wu, 1994).

1994 yılında Pascale Fung ve Kathleen McKeown tarafından bir çalışmada Asya, Avrupa ve Hint dillerinde cümle eşleme işlemi için DK-vec diye geliştirilen algoritma ile kelimelerin kullanım sıklığı, kelimelerin cümledeki yeri ve yenilik bilgilerini kullanarak sözlük tabanlı cümle eşleştirme işlemi gerçekleştirmektedir (Fung ve McKeown, 1994).

1996 yılında Masahiko Haruno ve Takefumi Yamazaki tarafından yapılan bir çalışmada Japonca ve İngilizce gibi yapısal olarak birbirinden farklı dilleri hizalamak için istatistiksel olarak kelimeler ile karşılaştırmalarında sınırlama vardır. Bu tarz farklı dillerdeki metinleri hizalamak için önerilen yöntem cümleler içerisindeki kelimelerin karşılıklarını karşılaştırmaktır. Kelimeleri karşılaştırmak için iki dilli bir sözlük kullanmaktadır veya kelimeleri istatiksel yöntemler kullanılarak birbirleriyle karşılaştırılmaktadır. Geliştirilen yöntem olarak kademeli olarak birbirine karşılık gelen cümle çiftlerini bulmaktadır. Sonrasında iki cümlenin karşılıklarını değerlendirerek yeterli sayıda sözcük karşılığını kontrol etmektedir. Sonuç olarak yapısal diller çeşitli uzunluklarda ve türlerde metinler olabilmektedir. Geliştirilen bu yöntem ile Japonca ve İngilizce dilleri için standart yöntemlerden daha başarılı olmuştur (Haruno ve Yamazaki, 1996).

1999 yılında I. Dan Melamed tarafından yapılan bir çalışmada iki dilde çevirisi olan metinlerden faydalı bilgi çıkarmak için ilk önce metin haritalarına karşılık gelen kelimeleri ve metin segment sınırlarını bulup iki dil metin haritalarını çıkartmıştır. Bu makale metin haritalama algoritmalarının kalitesini ölçmek için üç ana görevi

bulunmaktadır. Bunlar sinyal oluşturma, gürültü filtreleme ve arama aşamalarıdır. Smooth Injective Map Recogniser (SIMR) algoritması ile bu üç ana görevi iyileştirmek amaçlamıştır. Fransızca - İngilizce ve Korece - İngilizce gibi diller üzerinde başarılı sonuçlar elde etmiştir. SIMR algoritmaları Geometrik segment hizalama (GSA) algoritması kullanılarak segment hizalama işlemini başarılı bir şekilde gerçekleştirmiştir. SIMR algoritması kullanılarak 200 megabayt üzerinde Fransızca - İngilizce metin haritaları üretmiştir. GSA ise bu metin haritalarını hizalama işlemine dönüştürmüştür. Hizalamalara ait metinler ve haritalar Dilsel Veri Konsorsiyumu (LDC) içerisinde mevcuttur (Melamed, 1999).

2002 yılında Robert C. Moore tarafından yapılan bir çalışmada cümle eşleme sistemlerinde önceki yaklaşımlarda cümle uzunluğu veya sözlüğe dayalı yöntemler bulunmaktadır. Bu yöntemlerden uzunluğa dayalı yöntemler başarılı ve sözlüğe dayalı yöntemler ise daha başarı olduğu gözlemlenmiştir. Fakat bu iki yöntem de yavaştır. Geliştirilen bu yöntem ile bu iki yaklaşım uyarlandıktan sonra birleştirilir ve kelimeleri ve cümleleri ayırmanın ötesinde dil ve cümle hakkında hiçbir bilgi gerektirmemektedir (Moore, 2002a).

2006 yılında Xiaoyi Ma tarafından yapılan bir çalışmada cümle hizalama yaklaşımlarının bazıları uzunluğa dayalı bazıları ise sözlük tabanlı yaklaşımlar ve bazıları ise bu iki yaklaşımı içermektedir. Champollion büyük ölçüde sözcüksel bilgilere bağlıdır, ancak cümle uzunluğu bilgilerini de kullanmaktadır. Bir çeviri sözlüğü gerekmektedir. Geçmiş deneylerde, çeviri sözlüğü daha da büyüdükçe Champollion'un performansının arttığını göstermektedir. Champollion ilk başta Çince - İngilizce paralel metinleri hizalamak için geliştirilmiştir. Sonrasında Arapça - İngilizce ve Hintçe - İngilizce gibi diller için geliştirilmiştir (Ma, 2006).

2006 yılında Ahmet Mustafa Güngör ve Tunga Güngör tarafından yapılan bir çalışmada cümle eşleştirme için geliştirilen yöntem cümle ve paragrafların konum bilgileri ve uzunlukları kullanılarak cümle eşleştirme gerçekleştirmektedir. Bu yöntem tamamen dilden bağımsız ve uygulanması kolaydır. Cümle eşleştirme başarısının paragraflar eşleşme aşamasındaki başarısı ile doğru orantılı olduğu gözlemlenmiştir (Gungor, 2006).

2007 yılında Dániel Varga ve arkadaşları tarafından yapılan bir çalışmada cümle eşleme işlemi için her cümle çiftinin benzerlik matrisi hesaplanmaktadır. En az 500 cümlelik komşuluk hesaplanır veya tüm cümleler için %10'undan yakını hesaplanmaktadır. (Gale ve Church, 1993b) tarafından geliştirilen yöntem gibi cümlelerin uzunluklarını karşılaştırmaktadır ve sözlük tabanlı benzerlikleri kontrol etmektedir (Halácsy, Kornai, Nagy, Németh ve Trón, 2007).

2010 yılında Rico Sennrich ve Martin Volk tarafından bir çalışmada cümle eşleştirme işlemi için ilk önce metnin başından sonuna kadar her bölüm için algoritma hesaplanır ve hedef metin ile kaynak metin arasındaki benzerlik puanı olarak "Blue" kullanılarak noktalar belirlenmektedir. Sonrasında bu bağlantı noktaları arasındaki cümleler Blue tabanlı yöntemler ya da (Gale ve Church, 1993b) uzunluk tabanlı algoritma ile hizalanmaktadır (Sennrich ve Volk, 2010).

2010 yılında Fabienne Braune ve Alexander Fraser tarafından yapılan bir çalışmada cümle eşleştirme için geliştirilen yöntem (Moore, 2002b) tarafından geliştirilen yöntem benzemektedir. Cümle uzunluğuna istatistiksel yöntemler kullanmaktadır (Braune ve Fraser, 2010).

2010 yılında Peng Li, ve arkadaşları tarafından yapılan bir çalışmada cümle hizalama işlemi için hem sözlük hem de uzunluk tabanlı gerçekleştirmektedir. Champollion Tool Kit (CTK) cümle hizalamak için sözlük çevirilerini kullanır fakat iki belgedeki cümle sayısı eşit değilse başarısı düşmektedir. Champollion algoritmasının temel özellikleri kullanılarak üzerinde yapılan geliştirmeler ile Fast Champollion yöntemi önermiştir. Bu yöntem CTK yönteminden dört veya beş kat daha hızlı ve uzun metinlerde 39,4 kat daha hızlı ve güçlüdür (Li, Sun ve Xue, 2010).

2019 yılında Brian Thompson ve Philipp Koehn tarafından yapılan bir çalışmada cümle hizalama işlemi cümle blokları skorlama işlemi yaparak gerçekleştirmektedir. Skorlama işlemi yapılırken çıkan vektör sonuçları cümle sayılarına bağlı değildir. Skorlamak için kosinüs benzerliğinden faydalanmaktadır (Thompson ve Koehn, 2019).

2020 yılında Katsuki Chousa ve arkadaşları tarafından yapılan bir çalışmada cümle eşleştirmek için SpanAlign diye bir yöntem önermektedir. Bu yöntem puanlama ve

optimizasyon olarak iki yöntem kullanmaktadır. Puanlama sistemi için SQuAd 2.0 soru yanıtlama sistemi gibi hedef metinde cevap olarak kaynak metnin çevirisindeki cevabı tahmin eder ve puanını hesaplamaktadır. İngilizce - Japonca cümle hizalamak için çift yönlü kullanılabilmektedir (Chousa, Nagata ve Nishino, 2020).

2020 yılında Emir Kızılırmak tarafından yapılan tarafından yapılan bir çalışmada cümle hizalama işlemi için geliştirilen yöntem ilk önce cümlede sözlükte bulunmayan kelimelerin tespit edilip istatistiksel olarak skorlanmasıyla birlikte diğer dildeki cümleler içerisindeki sözlükte bulunmayan kelimelerin istatistiksel yöntemler ile skorlanarak Levenstein algoritmasıyla birlikte değerlendirilmesiyle gerçekleştirmektedir. Eğer skorlar eşit çıkarsa o zaman sözlük yardımı ile tekrar skorlanıp karşılaştırma yapmaktadır (Kızılırmak, 2020).

2021 yılında Shengxuan Luo ve arkadaşları tarafından yapılan tarafından yapılan bir çalışmada cümle hizalama işlemi için iki dil kelime benzerlikleri ve cümle uzunluklarıyla birlikte diller arası kelime vektörlerinin eğitim işlemiyle gerçekleştirmektedir. Sonrasında cümleleri hizalamak için “Earth Mover’s Distance” algoritmasıyla gerçekleştirmektedir. Çince - İngilizce biyomedikal metinler üzerinde test etmiştir (Luo, Ying, Li ve Yu, 2022).

3. MATERYAL VE YÖNTEM

Son yıllarda bilgisayar ortamındaki verilerin artmasıyla ve bilgisayar donanımlarının gelişmesiyle birlikte bilgisayar bilimi problemlerinin çözümünde önem kazanmaktadır. Yapay zekâ ilk olarak 1950 yılında Alan M. Turing'in "Hesaplama Makineleri ve Zekâ" makalesi ile "Makineler düşünebilir mi?" sorusuyla birlikte düşünebilen bir makine olup olmayacağı yetisi bu soruyla birlikte başlamıştır. Sonrasında 1956 yılında bu soruya Dartmouth Kolejinde gerçekleşen bir konferansta McCarthy ve arkadaşları tarafından cevaplanmaya çalışılmıştır ve tarihte ilk kez "Yapay Zekâ" kelimesi ortaya çıkmıştır. Daha sonraki yıllarda Türkiye'de bu önemli soruyu ilk kez 1959 yılında Atatürk Üniversitesinde yapılan bir halk konferansında Ord. Prof. Cahit Arf tarafından 1958 - 1959 öğretim yılında dile getirilmiştir (Sari, 2021).

Yapay zekâ, insan beynini kendine referans alarak insan zekâsını taklit etmektedir. Yapay zekâ genellikle sınıflandırma, tahmin etme ve kümeleme gibi çıkarımlar yapmaktadır. Yapay sinir ağları, bulanık mantık, uzman sistemler ve genetik algoritmalar gibi başlıca yöntemleri bulunmaktadır (Demirhan, Kiliç ve Güler, 2010). Doğal dil işleme (DDİ) ise bilgisayar bilimi ve yapay zekânın önemli bir dalıdır. Doğal dil işleme insanlar arasında konuşulan doğal dilin ve bu dilin bilgisayarlar tarafından anlamlandırılma işlemidir. Aynı zamanda Yapay Zekâ ve Dil biliminin bir alt bilimidir. Doğal dil işleme alanı günümüzde arama motorları, istenmeyen e-posta tespiti, duygu analizi, diller arası çeviri, metin sınıflandırma, bilgi çıkarımı, varlık isim tanıma, sohbet botu ve cümle eşleştirme gibi alanlarda kullanılmaktadır (Balli, Guzel, Bostanci ve Mishra, 2022, Yıldırım ve Alkış, 2016).

Cümle eşleştirme işlemi birebir çevirisi yapılmış ve sıralaması bir şekilde değişmiş cümlelerin, kaynak metinler ile hedef metinler arasında gerçekleşmektedir. Birebir çevirisi olan fakat kaynak metin ile hedef metin arasında aynı sırada olmayan veya herhangi bir şekilde sırası karışmış cümleler arasında eşleştirme işlemi gerçekleştirilmektedir. İki farklı dil arasındaki cümleleri eşleştirmek için insan gücüne ihtiyaç duyulmaktadır. İnsan gücüyle cümle eşleştirme işlemi yapmak hem yorucu hem de çok zaman alıcı olmaktadır. Cümle eşleştirme işlemi için literatürde çeşitli algoritmalar

mevcut olmasına rağmen İngilizce ve Türkçe için çok fazla çalışma bulunmamaktadır. Bu sebepten dolayı İngilizce ve Türkçe cümleler arasında derin öğrenme kullanılarak farklı bir yöntem ile cümle eşleme sistemi geliştirilmesi amaçlanmaktadır.

3.1 Doğal Dil İşleme

Doğal dil işleme, analiz ederek, anlamaya çalışan veya bir dilin veya birden fazla dili üreten sistemi ifade etmektedir (Allen, 2003).

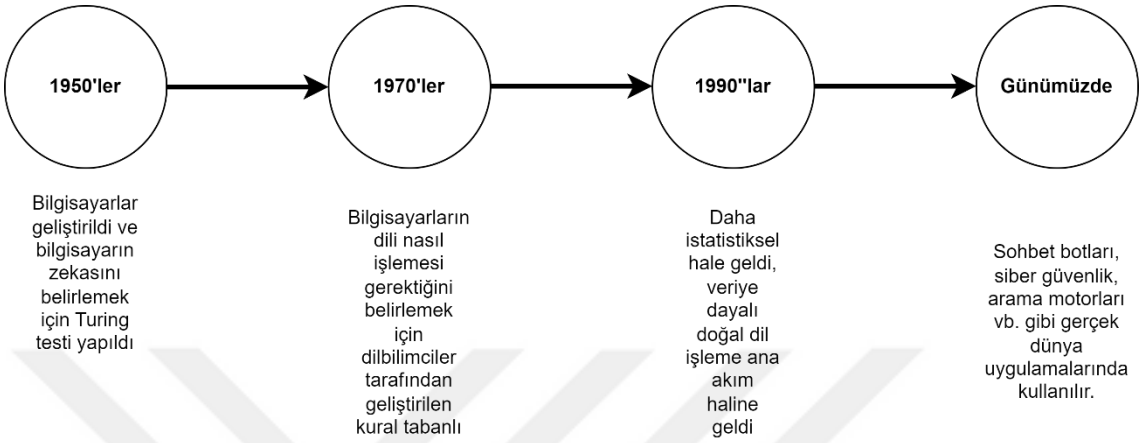
DDİ bilgisayarların insan dillerinde yazılmış, söylenmiş ifadeleri veya kelimeleri anlamasını sağlayan Yapay Zekâ ve Dilbilim dalıdır. Kullanıcıların işini kolaylaştırmak ve bilgisayar ile doğal arasında iletişim kurmak amacıyla ortaya çıkarak iki kısma ayrılmaktadır (Khurana, Koli, Khatter ve Singh, 2022).

1940'ların sonlarına doğru doğal dil işlemenin temelleri ilk önce makine çevirisi olarak başlanmıştır. Makine çevirisi ilk olarak Rusya ve İngilizce metin çevirileri üzerine ağırlıklı çalışmalar gerçekleştirmiştir (Léon, 2014).

1950'lerde ise yapay zekâ ve dilbilimin kesişimi olarak ortaya çıkmaktadır. Bu dönemde otomasyon ve olasılık modellerinin kullanımıyla sınırlı kalmıştır. Otomasyonların kullanımı Alan Turing'in algoritma hesaplamak için geliştirdiği modelle birlikte gerçekleşmiştir. Sonrasında bu dönemde gerçekleşen çalışmalarla birlikte Chomsky tarafından ilk olarak sonlu durum makineleri cümlenin gramerini tanımlamanın yolu olarak görülmüş ve bu yönde çalıştırmalar geliştirilerek bir model oluşturulmuştur. Chomsky tarafından oluşturulan bu modeller cebir ve küme teorisini kullanan ilk biçimsel dil kuramını kullanımına yol açılmıştır. Bu model ilk olarak Chomsky tarafından kullanılmasına rağmen 1958 yılında Backus ve Knoll tarafından ALGOL programlama dilinin temellerini bağlamdan bağımsız gramerler oluşturmaktadır. Bu geliştirilen modeller ise 1960'ların sonlarına doğru daha başarılı dilbilgisi dönüşümlerine yol açmıştır (Kumar, 2013).

SHRDLU, MIT Yapay Zekâ Laboratuvarı'ndan Terry Winograd tarafından 1968 ile 1970 yılları arasında yazılmış bir doğal dil anlama programıdır ve doğal dil işlemenin kırılma noktası olarak kabul edilmektedir (Tüysüz, 2003). 1970'li yıllardan günümüze kadar

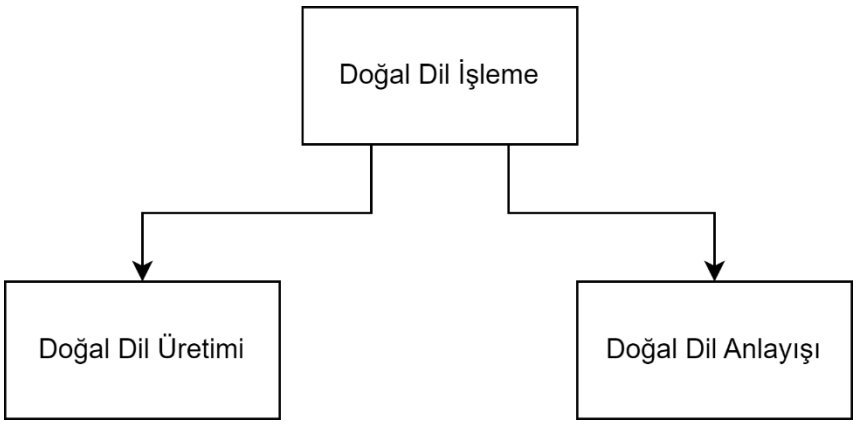
doğal dil işleme alanında çok fazla çalışmaya yer verilmektedir. Şekil 1'de Doğal dil işlemenin yıllar göre değişimine ait görsel verilmektedir.



Şekil 1. Doğal Dil İşlemenin Yıllara Dağılımı

3.2 Doğal Dil İşlemenin Bileşenleri

Doğal dil işleme insan ve bilgisayar arasındaki iletişimi sağlayan bir yapay zekânın alt dalıdır. Doğal dil işleme bileşenleri olarak metni anlama ve oluşturma görevlerini gerçekleştiren Doğal Dil Üretimi (DDÜ) ve Doğal Dil Anlayışı (DDA) olmak üzere iki bölüme ayrılmaktadır. Şekil 2'de doğal dil işlemenin bileşenlerine ait doğal dil üretiminin ve doğal dil anlayışının temel gösterimi verilmektedir.



Şekil 2. Doğal Dil İşlemenin Bileşenleri

3.2.1 Doğal dil üretimi

Doğal dil üretimi, hesaplamalı bilim ve yapay zekâ alanlarının bir alt alanı olarak konumlandırılmaktadır. Doğal dil üretimini, doğal dillerde anlaşılır metinler üreten uygulamalar veya bilgisayar sistemleri oluşturmayı ve oluşturulan bu belgelerin en belirgin örneği doğal dilde belgeler, raporlar, açıklamalar, özetler ve diğer metin türleri içeren bilim dalı olarak adlandırılmaktadır (Staykova, 2014).

Doğal dil üretimi, 1980'li yıllarda önem kazanmasına rağmen ilk olarak 1950'lerde makine çevirisinin küçük bir bölümü olarak ortaya çıkmıştır. 1960'lı yıllarda ise cümle içerisinde gramer kontrol etmek için doğal dil üreteçleri kullanılmıştır. 1970'lerde ise doğal dil üretimi mantık yürütüyor, metin kurallarını değerlendiriyor ve sofistike dilbilgisi modelleri olarak kullanmıştır. Ürettikleri metinlerin sayıca az olsa da literatürdeki en akıcı metinlere referans olarak gösterilmektedir (Dale, Moisl ve Somers, 2000).

3.2.2 Doğal dil anlayışı

Doğal dil anlayışı doğal dil işlemenin diğer bir alt kümesi olarak adlandırılmaktadır. Doğal dil işleme bir metni veya cümleyi analiz etmeye ve anlamaya çalışmaktadır. Doğal dil anlayışı ise insanların doğal olarak iletişim kurma biçimini taklit ederek otomatik bir şekilde metin oluşturmasına imkân sağlamaktadır.

Doğal dil anlayışı, makinelerin doğal dili anlamasını, kavramları, varlıkları, duyguları ve anahtar sözcükler gibi anlamlar çıkararak çeşitli analizler çıkartmasını sağlamaktadır. Günümüzde bunun en belirgin örneği bankaların müşteri hizmetleri uygulamasında, müşteriler tarafından sözlü veya yazılı bildirilen sorunları anlamak için kullanılmaktadır. Dilbilim, dilin anlamını, bağlamını ve çeşitli biçimlerini içeren bilim dalı olarak adlandırılmaktadır (Khurana ve diğerleri, 2022).

3.3 Doğal Dil İşleme Kullanım Alanları

Doğal dil işlemenin günümüzde çeşitli alanlarda kullanılmaktadır. Doğal dil işleme süreçlerinin kullanım alanlarının başlıca kategorileri aşağıdaki gibi gösterilmektedir,

- Makine çevirisi
- Cümle eşleştirme
- Spam filtreleme
- Metin özetleme
- Soru yanıtlama
- Duygu analizi
- Sohbet botları
- Dil tanımlama
- Varlık tanıma
- Sözlük anlam belirsizliği giderilmesi
- İntihal tespiti
- Ses bilimi

gibi çeşitli uygulamalarda kullanılmaktadır (Talan ve Aktürk, 2022).

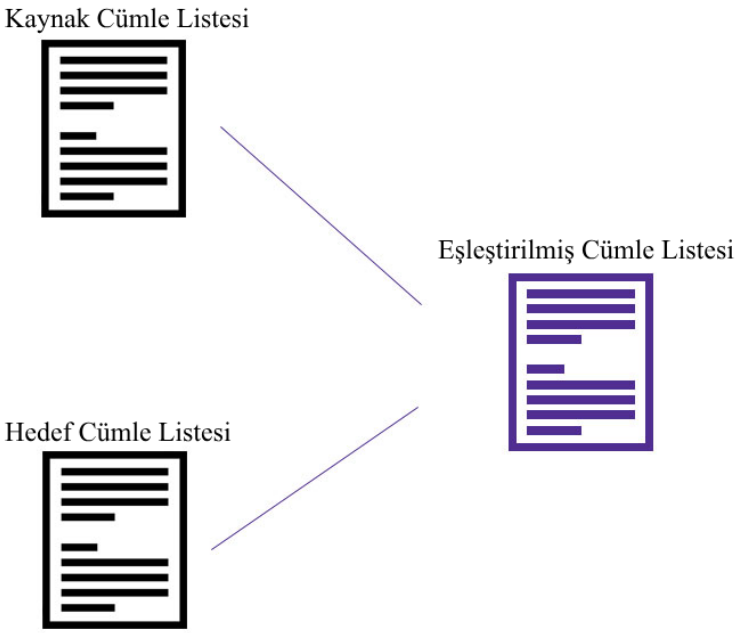
3.4 Cümle Eşleme

Cümle eşleme işlemi kaynak bir dildeki cümlenin diğer bir hedef dildeki cümlelerinin eşleşme sırasının herhangi bir şekilde sırasının değişmesiyle gerçekleşmektedir. Bu sıranın değişmesiyle birlikte tekrar eşleştirmek için cümle havuzundaki doğru cümleyle eşleşmesi sonucunda gerçekleşmektedir. Bir dilbilim cümlesi, soru, ifade, ünlem, istek, emir gibi çeşitli türleri bulundurmaktadır. Bu cümleler belirli bir gramer yapısına dayanan bir anlam içermektedir. Bu cümleler çeşitli dillerde olabilir ve bazen uzun bir cümlenin başka bir dildeki çevirisi birden fazla kısa cümleye denk gelebilmektedir. Bu yüzden bir cümlenin, başka bir dildeki çevirisi ile karşılaştırdığımızda cümlenin uzunluk sınırlarını belirlemek zordur. Doğal dil işleme alanında eşleştirme işlemini cümleler, paragraflar, bölümler, kısımlar gibi eşleştirme işlemi kelime veya alt kelime gruplarına göre eşleştirmesiyle yapılabilmektedir. Cümle eşleştirme işlemi için günümüzde uzunluk tabanlı cümle eşleştirme, yer tabanlı cümle eşleştirme ve sözlüğe dayalı cümle eşleştirme işlemleri standart olarak kullanılan yöntemlerdir. Uzunluk tabanlı cümle eşleştirme ve yer tabanlı cümle eşleştirme işlemlerinde istatistiksel yöntemler kullanılarak cümle eşleştirme işlemi gerçekleştirmektedir. Sözlük tabanlı cümle eşleştirme işlemi için iki

veya daha fazla dil için eşleştirme yapılacaksa aynı sayıda birbirinin çeviri kelimelerini kapsayan sözlüğe ihtiyaç bulunmaktadır (Kızılırmak, 2020).

Kelime anlam belirginleştirme (KAB), kullanıldığı bağlamda birden fazla anlamı olan bir kelime anlamının bilgisayarlı ve otomatik olarak belirlenmesi diye tanımlanabilmektedir. KAB, sözlük tabanlı cümle eşleştirmede önemlidir çünkü bir kelimenin birden fazla anlamı olabilmektedir. Örneğin, İngilizce dilinde "yüz" kelimesinin anlamı isim, sıfat ve fiil olarak kullanımına göre farklılık gösterebilmektedir. Verilen örnekten de görüleceği üzere, bir kelimenin anlamının doğru bir şekilde tespit edilmesi son derece başarılı cümle eşleştirme işlemi için önem taşıyan bir süreçtir (Tüysüz ve Güvenoğlu, 2014).

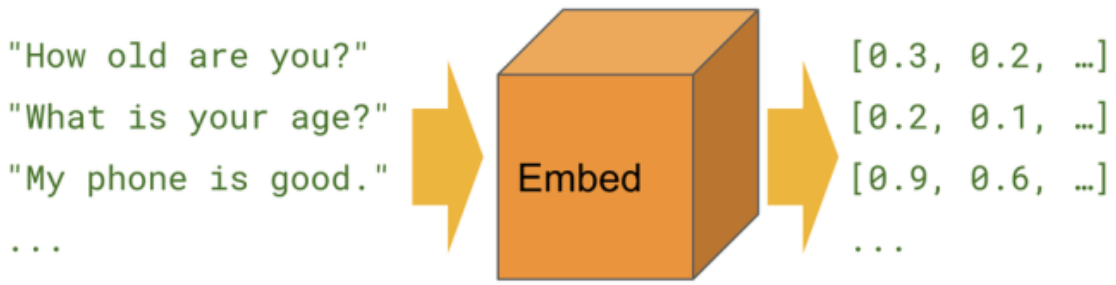
Bu klasik yöntemlerin yanı sıra derin öğrenme yöntemleriyle cümle eşleştirme işlemi gerçekleştirilebilmektedir. Cümle eşleştirme işlemi yapılırken bir adet cümlenin karşılığı olarak bir adet cümle veya n adet cümle eleştirmesi yapılabilmektedir. Bunun yanı sıra cümle eşleştirme işleminde cümle karşılığı da diğer dil çevirileri içerisinde bulunamayabilir (Ngo Ho, 2021). Şekil 3'te cümle eşleştirme işlemi kaynak ve hedef cümle listesi arasında eşleştirme işlemi yapıldıktan sonra eşleştirilmiş cümle listesi içerisinde temel gösterimi bulunmaktadır.



Şekil 3. Cümle Eşleştirme

3.5 Evrensel Cümle Kodlayıcı

Evrensel cümle kodlayıcı için araştırmacılar, metne bağlı kelime vektör karşılığını bir adım daha ileriye götürerek evrensel cümle kodlayıcısının temelde iki farklı varyasyonu bulunmaktadır. Bunlardan ilk varyasyon "Transformer" mimarisinde işlemler paralel olarak gerçekleştirildiği için eğitim için ayrılan süre azalmaktadır. İlk olarak "Encoder" katmanına gelen metni, sayısal bir vektör haline dönüştürebilmektir. Bu sayede kelimelerin yerleri de sayısal olarak saklanmaktadır. "Encoder" katmanı sayesinde cümleleri sıkıştırarak daha az yer kaplaması ve daha hızlı şekilde kullanılabilmesi amaçlanmıştır. Diğer varyasyon ise Derin Ortalama Ağından (DOA) faydalanarak gerçekleştirilebilmektedir. DOA modelinde ilk önce kelimeleri ve metinleri ikişer olarak ayırmayı gerçekleştiren bi-gram modeli için kelime vektörleri ve karşılaştırmanın ilk önce birlikte ortalaması hesaplanarak alınmaktadır. Ardından ileri beslemeli bir sinir ağından geçirilerek derin bir ortalama ağından yararlanılmaktadır. DOA modeli, "Transformer" modeline çok benzeyen bir şekilde eğitilmektedir. DOA modelinin en büyük avantajı ise hesaplama süresinin giriş dizesinin uzunluğu ile doğru orantılıdır. Bu sebepten dolayı DOA modeli kullanılarak metinlerin sınıflandırma görevinde iyi bir performans elde ettiğini göstermektedir (Cer ve diğerleri, 2018). Şekil 4'te evrensel cümle kodlayıcısına genel bir bakış gösterilmektedir.



Şekil 4. Evrensel Cümle Kodlayıcıya Genel Bakış (Martín Abadi ve diğerleri, 2015)

3.6 BERT

BERT algoritması, “Bidirectional Encoder Representations from Transformers” kelimelerinin ilk harflerinin kısaltması olarak adlandırılmaktadır. Google tarafından geliştirilen bu algoritmayla birlikte kelimeleri tek tek değerlendirmek yerine öncesinde veya sonrasında bulunan kelimeleri veya eş anlamlı ve benzer kelimelerle birlikte yorumlamasıyla gerçekleştirmektedir. Geliştirilen bu yöntem sayesinde daha doğru sonuçlar elde edilmesi beklenmektedir. Çünkü makinelerin günlük insanlar tarafından konuşulan dili anlaması ve yorumlaması zor olmuştur. BERT algoritmasıyla ilgili en belirgin örneklerinden bir tanesi Google üzerinde daha önceleri “2019 Brezilya’dan Amerika’ya seyahat edecek kişilerin vize alması gerekiyor.” cümlesi arama yapıldığı zaman haber sitesi sonuçlar kullanıcılara gösteriliyordu. BERT algoritmasının geliştirilmesiyle birlikte cümle içerisindeki edat, bağlaç gibi kelimeler önem kazanmıştır. Bu sayede Google arama motoruna yazılan cümle daha doğru yorumlanmaya başlamasıyla birlikte kullanıcılara artık elçilik ve konsolosluk sayfaları kullanıcılara gösterilmeye başlanmıştır (Pozzi, 2022).

3.7 Sentence-BERT

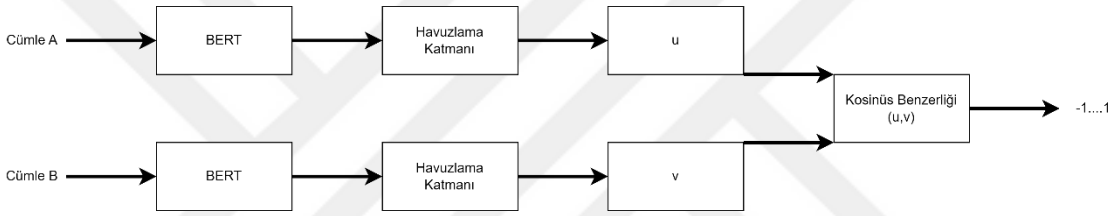
BERT algoritması içerisinde değişiklikler yapılmasıyla birlikte siyam sinir ağı ve yerel ikili örüntü kullanılarak Sentence-BERT (SBERT) algoritması ortaya çıkmıştır. Cümleleri ve kısa metinleri vektör haline çevirdikten sonra birçok alanda kullanılabilmektedir. Fakat birçok dilde eğitim verileri yaygın olmaması sebebiyle geliştirilen modellerin çoğu İngilizce dili için geliştirilmiştir (Reimers ve Gurevych, 2019).

Sentence-BERT’in çeşitli kullanım alanları mevcuttur. Bunların birkaç tanesine örnek vermek gerekirse,

- Cümlelerin vektör karşılıklarını hesaplama
- İki cümle arasındaki anlamsal benzerlikleri hesaplama

İki cümle arasındaki anlamsal arama ikiye ayrılmaktadır. Bunlardan biri simetrik arama ve diğeri asimetrik arama olarak adlandırılmaktadır. Simetrik arama olarak yapılan sorgu ve sorgunun çıktıları aynı uzunluk veya aynı miktarda içeriğe sahip olmaktadır. Asimetrik

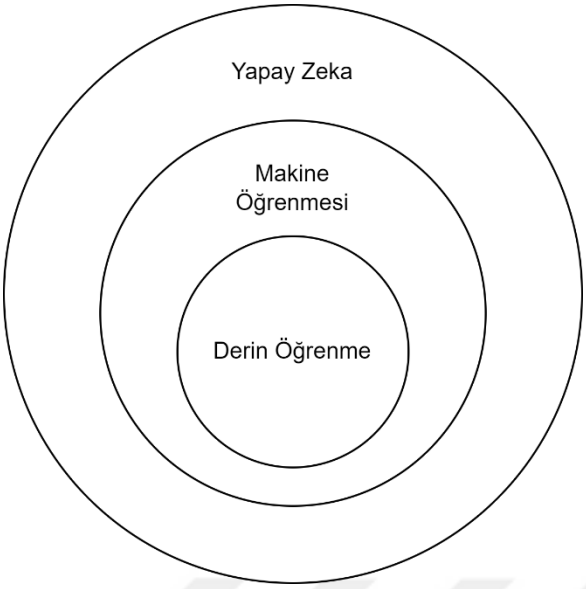
arama olarak yapılan sorgular genel olarak kısa olmaktadır ve sorgu bazı anahtar sözcükler ile yapılmaktadır. Yapılan bu sorguyla birlikte daha uzun ve daha fazla miktarda çıktılar üretmektedir. Bir diğer kullanım alanı ise verilen paragraf içerisinde semantik açıdan tekrar sıralayabilir ve büyük veya küçük metinler içerisinde cümleleri gruplamak için kullanılabilmektedir. Bir metin içerisinde cümlelerin anlamsal açıdan birbiriyle aynı veya birbirlerine benzerliklerini hesaplayabilmektedir ve bir dildeki cümlenin diğer bir dildeki karşılığını çevrimiçi bir şekilde bulmak için kullanılmaktadır. Birden fazla dil içerisindeki cümle karşılıklarıyla birlikte çeşitli işlemler yapılabilmektedir ve son olarak görüntüleri ve cümleleri aynı uzay vektörüne yerleştirmeye olanak sağlamaktadır. Şekil 5'te Sentence BERT'e ait genel bir mimari verilmektedir.



Şekil 5. SBERT Mimarisi

3.8 Derin Öğrenme

Derin öğrenme, yapay zekânın bir alt dalıdır ve yapay sinir ağlarına dayanan bir makine öğrenimi olarak adlandırılmaktadır. Yapay sinir ağları genel olarak insan beynini taklit etmektedir. Derin öğrenme; görüntü, metin veya ses dosyaları üzerinde çalışabilmektedir. Derin öğrenme büyük miktarda veri kümesi üzerinde çalışmaktadır. Büyük miktarda veriyi eğitmek uzun zaman almasına rağmen sorunu uçtan uca çözmektedir. Derin öğrenme denetimli öğrenme ve denetimsiz öğrenme olmak üzere iki alt başlıkta incelenmektedir. Denetimli öğrenme, verilen eğitim seti içerisinden öngörülemeyen veriler hakkında tahminler yapması beklenmektedir. Denetimsiz öğrenme ise önceden belirlenmiş bir veri etiketi olmadan bilinmeyen sonuçları bulması beklenmektedir (Chollet, 2021). Şekil 6'da Derin öğrenmenin yapay zekâ ve makine öğrenmesinin temel bir gösterimi bulunmaktadır.



Şekil 6. Derin Öğrenme

3.9 Kosinüs Benzerliği

Kosinüs benzerliği, doğal dil işleme alanında yaygın olarak kullanılan bir metriktir ve iki metin belgesi arasındaki benzerliği hesaplamakta kullanılmaktadır. Bu benzerliği hesaplamak için vektörlerin arasındaki kosinüs değeri hesaplandıktan sonra gerçekleşmektedir (Salton ve Buckley, 1988).

Kosinüs benzerliğini hesaplamak için iki metin, cümle, paragraf veya tüm belge arasındaki vektörü karşılıklarının bulunması sonrasında iki vektörün arasındaki açıyı ölçmektedir. Hesaplanan bu iki vektör arasındaki açı vektörlerinin iç çarpımının uzunluklarına bölünmesiyle bulunmaktadır. Sıfır olmayan iki vektör verildiğinde, aralarındaki kosinüs benzerliği aşağıdaki denklemlerde gösterilmektedir. Öklid nokta çarpımı denklem 3.1 içerisinde gösterilmektedir.

Öklid Nokta Çarpımı:

$$A \cdot B = \| A \| \cdot \| B \| \cdot \cos(\theta) \tag{3.1}$$

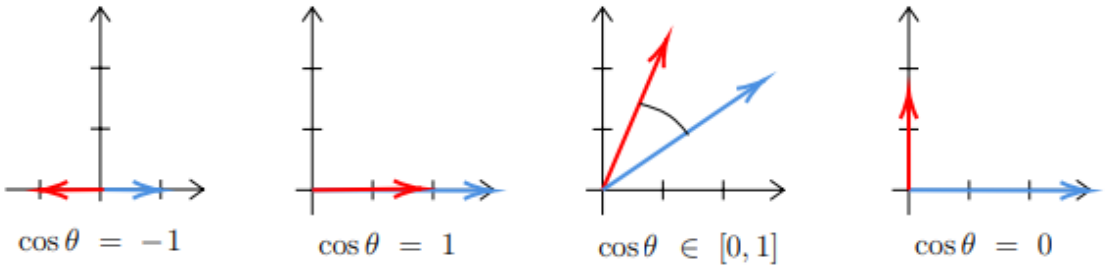
Sonrasında denklem 3.2'deki gibi gösterilmektedir.

$$\cos(\theta) = \frac{A \cdot B}{\| A \| \cdot \| B \|} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (3.2)$$

Buradaki A_i ve B_i, A ve B vektörlerinin bileşenleri olarak gösterilmektedir. Toplam sembolünde yer alan n ise vektörün bileşen sayısını göstermektedir. A ve B vektörü arasındaki benzerlik, aralarındaki açının kosinüsü olarak hesaplanmaktadır. Benzerlik bulunması denklem 3.3 üzerinde gösterilmektedir.

$$Benzerlik(A, B) = \cos(\theta)_{A.B} \quad (3.3)$$

Bir kosinüs benzerliği için ölçü değeri -1 ile 1 arasında değişmektedir. -1 olan vektörlerin tam olarak zıt olduğunu, 1 olan vektörlerin her ikisinin de aynı vektör olduğunu ve 0 vektörlerin ortogonal olduğunu göstermektedir. Ortogonal olması birbirleriyle veya vektörlerin kesim noktalarındaki teğetleri ile dik açı yapacak şekilde kesişmesidir. Olası değerlerin geriye kalanı benzerlik veya farklılığı olarak göstermektedir. Şekil 7'de olası kosinüs benzerlik durumları gösterilmektedir.

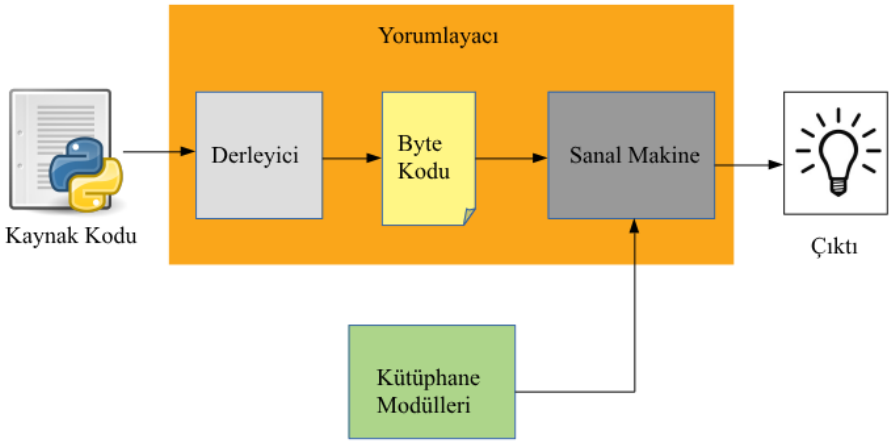


Şekil 7. Olası Kosinüs Benzerliği Durumları (Gómez Sánchez, 2021)

Kosinüs benzerliğinin karmaşıklığı O(n) olduğundan dolayı lineer bir karmaşıklığı bulunmaktadır. Elemanlarının çoğunun sıfır olduğu vektörlerde yalnızca sıfır olmayan boyutların dikkate alınması gerekmektedir. Bu sayede hesaplama hızı artmaktadır (Gómez Sánchez, 2021).

3.10 Python Programlama

Python programlama dilinin geliştirilmesi 1980'lerin sonuna dayanmaktadır. Kodun tasarımı, okunabilirliği, sözdizimi ve kod kavramları C++ veya Java gibi dillerden daha az kod ifade edilmesine olanak sağlamaktadır. Python programlama dilinin avantajları temiz söz dizimi, hataya tolerans etmesi, içerisinde hazır kullanılabilecek faydalı çok fazla modül bulunması, deneme yapma veya basit problemleri çözme konusunda kullanışlı olması ve normal dağılımı olmasına rağmen güçlü bir geliştirme ortamı sağlamaktadır (Тульчак, Марчук, Tulchak ve Marchuk, 2016). Şekil 8'de Python programlama dilinin basit bir şekilde mimarisi gösterilmiştir.



Şekil 8. Python Basit Mimarisi

Python programlama dilinde kod yazılması ve yazılan kodların okunabilirliği kolaydır. Ayrıca içerisinde birçok hazır kütüphane bulunması sebebiyle birçok açık kod kaynağına olanak sağlamaktadır. İhtiyaç olan paketleri yüklemek için "pip install KütüphaneAdı" şeklinde yüklenmesi gerekmektedir. Şekil 9'da örnek bir Python programlama dilinin kütüphane yüklenmesi gösterilmiştir.

```
pip install data
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting data
  Downloading data-0.4.tar.gz (7.0 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages (from data) (1.16.0)
Requirement already satisfied: decorator in c:\programdata\anaconda3\lib\site-packages (from data) (5.1.1)
Collecting funcsigs
  Downloading funcsigs-1.0.2-py2.py3-none-any.whl (17 kB)
Building wheels for collected packages: data
  Building wheel for data (setup.py): started
  Building wheel for data (setup.py): finished with status 'done'
  Created wheel for data: filename=data-0.4-py3-none-any.whl size=7227 sha256=e979b62fd266ca2234cc44cbe739d4fee918492ef79ffc504
037871c062abe5a
  Stored in directory: c:\users\emir\appdata\local\pip\cache\wheels\8a\0b\a3\37ca07d5a2838bba2e475e8090455e40b94631bd57a99a35f4
Successfully built data
Installing collected packages: funcsigs, data
Successfully installed data-0.4 funcsigs-1.0.2
Note: you may need to restart the kernel to use updated packages.
```

Şekil 9. Python Kütüphane Yüklenmesi

3.10.1 Numpy kütüphanesi

1990'lı yılların ortasında geliştirilen Numpy kütüphanesi, Python programlama dilinin bir kütüphanesidir. Bu kütüphane sayesinde matrislere, vektörlere ve yüksek boyutlu diziler üzerinde işlem yapmayı gerçekleştirmektedir. Esas amacı bilimsel hesaplamalar yapmaya olanak sağlamasıdır (Harris ve diğerleri, 2020).

Numpy kütüphanesinin iki temel amacı bulunmaktadır. İlk amacı N boyutlu bir dizi ve evrensel bir fonksiyon nesnesi üzerinde işlemler yapmaya olanak sağlamaktadır. N boyutlu diziyi tanımlayan iki temel özelliği bulunmaktadır. Temel özelliklerinden biri dizinin şekli diğeri ise dizinin hangi tür öğelerden oluşturduğu bilgisini içermektedir. Dizinin şekli ve dizinin boyutu boyunca ne kadar değişebileceği hakkında kullanıcılara bilgi sağlamaktadır. Bir diziyi tanımlayan diğer önemli özellik ise dizinin hangi tür öğelerden oluşacağının bilgisini içermektedir. Numpy kütüphanesi ile oluşturulan dizileri aynı tür veriyi içerdiği için her öğe aynı boyutta bellek bloğu kaplamaktadır ve dizideki her bellek bloğu aynı şekilde yorumlanmaktadır (Oliphant, 2015).

3.10.2 Scipy kütüphanesi

2000'li yılların başında geliştirilen Scipy kütüphanesi, Python programlama dilinin bir kütüphanesidir. Scipy kütüphane sayesinde bilimsel problemlerin modellenmesine ve çözülmesine olanak sağlamaktadır. Scipy kütüphanesi optimizasyon, cebir denklemleri, diferansiyel denklemler, özdeğer problemleri, entegrasyon ve birçok problemin çözülmesi için algoritmalar içermektedir. Aynı zamanda dizi veri yapıları ve hızlı sayısal rutinler sağlayan Numpy kütüphanesi üzerine inşa edilmiştir (Virtanen ve diğerleri, 2020).

3.10.3 Datasets kütüphanesi

Datasets kütüphanesi, Python programlama dilinin bir kütüphanesidir ve iki adet temel özelliğe bulunmaktadır. Bunlardan birincisi Datasets kütüphanesi içerisinde birçok amaçlı veri kümesi olmasıdır. Diğeri ise ön işlemeli veri seti içermesidir. Birçok amaçlı veri kümesi içerisinde görüntü işleme, ses verileri, 467 farklı dil içeren metinler bulunmaktadır. Bu veri setlerini tek bir komut satırı ile indirmeye ve önyüklemeye olanak sağlamaktadır. Önyükleme sonrasın da ise bu veri setlerini eğitmek veya değerlendirmek

için veri yüklemesine imkân vermektedir. Ayrıca önyüklemeli veri seti içerisinde kendi metin, görüntü ve ses dosyalarının eğitimine imkân sağlamaktadır.

3.10.4 Sentence Transformers kütüphanesi

Sentence Transformers kütüphanesi, Python programlama dilinin bir kütüphanesidir ve cümlelerin, paragrafların ve resimlerin vektörlerini hesaplamaya olanak sağlamaktadır. BERT, RoBERTa ve XLM-RoBERTa gibi dönüştürücü ağlar kullanarak çeşitli görevlere imkân tanımaktadır. Metinler ve benzer metinler ile kosinüs benzerliğini kullanılarak vektör uzayına yerleştirilmektedir.

Sentence Transformers cümle, metin ve görüntü vektör karşılıkları için Python programlama dili içerisinde kullanılmaktadır. Sentence Transformers kütüphanesi 100'den fazla dil için cümle, metin vektör karşılıklarını hesaplamak için kullanılmaktadır. Gerçekleşen hesaplama sonrasında bu vektör karşılıkları benzer bir anlama sahip cümleleri bulmak için kosinüs benzerliği ile karşılaştırılabilmektedir. Bu karşılaştırmalar sonucunda duygusal metin benzerliği, duygusal arama ve yeniden yazmak için kullanılabilmektedir (Reimers ve Gurevych, 2019).

3.10.5 PyTorch kütüphanesi

PyTorch kütüphanesi, Python programlama dilinin bir kütüphanesidir ve görüntü işleme, doğal dil işleme alanlarında açık kaynaklı makine öğrenmesi kütüphanesi olarak adlandırılmaktadır. PyTorch kütüphanesiyle birlikte derin öğrenme süreçlerinin geliştirilmesi ve uygulanması konusunda önemli ölçüde hız ve esneklik sağlamaktadır. PyTorch kütüphanesinin hızlı olmasının başlıca sebepleri GPU ve CPU kullanarak optimize edilmiş bir derin öğrenme kütüphanesidir. PyTorch kütüphanesiyle birlikte parçalara ayırma, indeksleme, matematiksel işlemler, doğrusal cebir ve indirgeme gibi bilimsel hesaplamalar konusunda optimize edildiği için ihtiyaçları hızlı ve tam anlamıyla karşılamak için kullanılmaktadır (Paszke ve diğerleri, 2019).

3.11 Önerilen Yöntem

İngilizce ve Türkçe cümleler arasında cümle eşleştirme işlemini gerçekleştirmek için ilk önce modelin eğitimiyle gerçekleştirilmektedir. Bu modelin eğitim süreci için birebir

çevirisi yapılmış İngilizce ve Türkçe cümlelere ihtiyaç bulunmaktadır. Cümle eşleme işlemini derin öğrenme yardımıyla eğitimini gerçekleştirmek için ilk önce Python programlama dilinin "Sentence Transformer" kütüphanesinin yüklenmesi gerekmektedir. "Sentence Transformer" kütüphanesinin yüklenmesinin ardından kütüphane içerisindeki hazır modellerden çok dilli olan eğitim modellerini sırasıyla yüklenmesi gerekmektedir. İngilizce ve Türkçe cümleler liste haline getirdikten sonra her bir dili ayrı değişkenlere atama işlemiyle gerçekleştirilmektedir. Atanan bu değişkenleri kullanabilmek için bu liste içerisindeki elemanları sayısallaştırmaya ihtiyaç bulunmaktadır. Sayısallaştırma işlemi için "Sentence Transformer" kütüphanesinin "Encode" metodunun yardımıyla birlikte İngilizce ve Türkçe cümlelerin teker teker vektör karşılıkları bulunmaktadır. Bulunan bu vektör karşılıklarını daha sonrasında kosinüs benzerliği yardımıyla birlikte iki vektör arasındaki benzerlikleri hesaplanmaktadır. Vektör benzerliklerinin hesaplanması sonucunda en yüksek skora sahip olunan cümle alınmaktadır. Alınan bu cümle sayesinde İngilizce ve Türkçe cümleler arasında birbirine benzer en yakın veya birebir karşılığı olan cümleyi bulma işlemi gerçekleştirilmektedir.

3.12 Birden Fazla Dile Sahip Modeller

Doğal dil işleme alanında çok sayıda veri seti ve model geliştirilmiştir. Fakat bu veri setleri ve modellerin dili çoğunlukla İngilizce diline odaklanmıştır. Birden fazla dilli cümle dönüştürücüleri kullanarak, ilgili cümleleri benzer uzay vektörlerinde konumlandırılmıştır (Reimers ve Gurevych, 2020).

3.13 Hazır Modellerin Kullanılması

1.000 adet Türkçe ve 1.000 adet İngilizce cümle arasında cümle eşleştirme işlemi için,

- all-mpnet-base-v2
- multi-qa-mpnet-base-dot-v1
- all-distilroberta-v1
- all-MiniLM-L12-v2
- multi-qa-distilbert-cos-v1
- all-MiniLM-L6-v2
- multi-qa-MiniLM-L6-cos-v1

- paraphrase-multilingual-mpnet-base-v2
- paraphrase-albert-small-v2
- paraphrase-multilingual-MiniLM-L12-v2
- paraphrase-MiniLM-L3-v2
- distiluse-base-multilingual-cased-v1
- distiluse-base-multilingual-cased-v2

Toplamda 13 modelin başarıları kendi aralarında test edilmiştir.

all-mpnet-base-v2 modeli incelendiğinde, bir milyardan fazla eğitim çiftinden oluşan, geniş ve çeşitli bir veri kümesi eğitilmiştir. Aynı zamanda 384 maksimum dizi uzunluğu, 768 boyutlu uzay vektörü ve 420 megabayt boyutunda bulunmaktadır.

"multi-qa-mpnet-base-dot-v1" modeli incelendiğinde, soru ve cevap çiftleri üzerine eğitilmiştir. Aynı zamanda 512 maksimum dizi uzunluğu, 768 boyutlu uzay vektörü ve 420 megabayt boyutunda bulunmaktadır.

"all-distilroberta-v1" modeli incelendiğinde, soru ve cevap çiftleri üzerine eğitilmiştir. Aynı zamanda 512 maksimum dizi uzunluğu, 768 boyutlu uzay vektörü ve 420 megabayt boyutunda bulunmaktadır.

"all-MiniLM-L12-v2" modeli incelendiğinde, bir milyardan fazla eğitim çiftinden oluşan, geniş ve çeşitli bir veri kümesi eğitilmiştir. Aynı zamanda 256 maksimum dizi uzunluğu, 384 boyutlu uzay vektörü ve 120 megabayt boyutunda bulunmaktadır.

"multi-qa-distilbert-cos-v1" modeli incelendiğinde, soru ve cevap çiftleri üzerine eğitilmiştir. Aynı zamanda 512 maksimum dizi uzunluğu, 768 boyutlu uzay vektörü ve 250 megabayt boyutunda bulunmaktadır.

"all-MiniLM-L6-v2" modeli incelendiğinde, bir milyardan fazla eğitim çiftinden oluşan, geniş ve çeşitli bir veri kümesi eğitilmiştir. Aynı zamanda 256 maksimum dizi uzunluğu, 384 boyutlu uzay vektörü ve 80 megabayt boyutunda bulunmaktadır.

Bu 13 modelin başarısı kendi aralarında test edildikten sonra en başarılı olan 4 adet model alınmıştır.

“multi-qa-MiniLM-L6-cos-v1” modeli incelendiğinde, soru ve cevap çiftleri üzerine eğitilmiştir. Aynı zamanda 512 maksimum dizi uzunluğu, 384 boyutlu uzay vektörü ve 80 megabayt boyutundadır.

“paraphrase-multilingual-mpnet-base-v2” modeli incelendiğinde, öğretici model olarak “paraphrase-mpnet-v4” ve öğrenci model olarak “xlm-roberta-base” kullanılmıştır. Aynı zamanda 128 maksimum dizi uzunluğu, 768 boyutlu uzay vektörü ve 970 megabayt boyutunda bulunmaktadır.

“paraphrase-albert-small-v2” modeli incelendiğinde, “nreimers/Albert-small-v2” modeli üzerine geliştirilmiştir. Aynı zamanda 256 maksimum dizi uzunluğu, 768 boyutlu uzay vektörü ve 43 megabayt boyutunda bulunmaktadır.

“paraphrase-multilingual-MiniLM-L12-v2” modeli incelendiğinde, öğretici model olarak “paraphrase-MiniLM-L12-v2” ve öğrenci model olarak “Microsoft/Multilingual-MiniLM-L12-H384” kullanılmıştır. Aynı zamanda 128 maksimum dizi uzunluğu, 384 boyutlu uzay vektörü ve 420 megabayt boyutunda bulunmaktadır.

“paraphrase-MiniLM-L3-v2” modeli incelendiğinde, “nreimers/MiniLM-L3-H384-uncased” modeli üzerine geliştirilmiştir. Aynı zamanda 128 maksimum dizi uzunluğu, 384 boyutlu uzay vektörü ve 61 megabayt boyutunda bulunmaktadır.

“distiluse-base-multilingual-cased-v1” modeli incelendiğinde, öğretici model olarak “mUSE” ve öğrenci model olarak “distilbert-base-multilingual” kullanılmıştır. Arapça, Çince, Hollandaca, İngilizce, Fransızca, Almanca, İtalyanca, Korece, Lehçe, Portekizce, Rusça, İspanyolca, Türkçe olmak üzere toplam 15 dil için geliştirilmiştir. Aynı zamanda 128 maksimum dizi uzunluğu, 512 boyutlu uzay vektörü ve 480 megabayt boyutunda bulunmaktadır.

“distiluse-base-multilingual-cased-v2” modeli incelendiğinde, öğretici model olarak “mUSE” ve öğrenci model olarak “distilbert-base-multilingual” kullanılmıştır. “distiluse-base-multilingual-cased-v1” modeli 15 dil desteklerken “distiluse-base-multilingual-cased-v2” modeli ise 50 dili desteklemektedir. Aynı zamanda 128 maksimum dizi uzunluğu, 512 boyutlu uzay vektörü ve 480 megabayt boyutunda bulunmaktadır (“Pretrained Models—Sentence-Transformers documentation”, 2022).

İlk olarak “paraphrase-multilingual-mpnet-base-v2” modeli kullanılarak test edilmiştir. Test edilen bu modelle birlikte 1.000 adet cümleden 967 tanesini doğru eşleştirme ve 1.000 adet cümleden 33 tanesini doğru olmayan eşleştirme yapmıştır ve toplamda %96.7 oranında başarılı elde edilmiştir.

İkinci sırada ise “paraphrase-multilingual-MiniLM-L12-v2” modeli kullanılarak test edilmiştir. Test edilen bu modelle birlikte 1.000 adet cümleden 966 tanesini doğru eşleştirme ve 1.000 adet cümleden 34 tanesini ise doğru olmayan eşleşme yaparak toplamda %96.6 oranında başarı elde edilmiştir.

Üçüncü sırada ise “distiluse-base-multilingual-cased-v2” modeli kullanılarak test edilmiştir. Test edilen bu modelle birlikte 1.000 adet cümleden 960 tanesini doğru eşleştirme ve 1.000 adet cümleden 40 tanesini ise doğru olmayan eşleşme yaparak toplamda %96 oranında başarı elde edilmiştir.

Dördüncüsü ve son model ise “distiluse-base-multilingual-cased-v1” modeliyle test edilmiştir. Test edilen bu modelle birlikte 1.000 adet cümleden 936 tanesi doğru eşleştirme ve 1.000 adet cümleden 64 tanesi doğru olmayan eşleşme yapmıştır. Bu sayede %93.6 oranında başarı elde edilmiştir. Hazır modellerle yapılan testlere ilişkin sonuçlar Tablo 1’de gösterilmektedir.

Tablo 1. Hazır Modellerin Başarıları

	paraphrase-multilingual-mpnet-base-v2	**paraphrase-multilingual-MiniLM-L12-v2**	**distiluse-base-multilingual-cased-v2**	**distiluse-base-multilingual-cased-v1**
Doğru	967	966	960	936
Yanlış	33	34	40	64

3.14 Oluşturulacak Model İçerisinde Hazır Veri Seti Kullanılması

Oluşturulacak olan modelin eğitilmesi için Türkçe ve İngilizce cümleler arasında birebir çevirisi yapılmış metinlere ihtiyaç bulunmaktadır. Bunun için Python programla dili içerisinde bulunan Datasets kütüphanesi kullanılmıştır. Bu kütüphane içerisinde çeşitli dillerde veri setleri bulunmaktadır. İtalyanca, Almanca, Fransızca dilleri başta olmak

üzere toplam 60 dilde çevirileri mevcuttur ve her dil içerisinde toplam 258.098 adet cümle bulunmaktadır. Bu veri seti toplam 791,01 megabayt veriden üretilmiştir ve veri seti 323,23 megabayt büyüklüğü boyutunda bulunmaktadır (Qi, Sachan, Felix, Padmanabhan ve Neubig, 2018).

Bu tez çalışmasında kullanılmak üzere Datasets kütüphanesi içerisindeki TED konuşmalarından birebir çevirisi yapılmış Türkçe ve İngilizce metinler kullanılmaktadır. Şekil 10'da TED konuşmalarının örnek kullanımı gösterilmektedir.

```
import datasets

data = datasets.load_dataset('ted_multi', split='train')

print(data)
print(data[0]['translations']['language'])
```

```
Found cached dataset ted_multi (C:/Users/emir/.cache/huggingface/datasets/ted_multi/plain_text/1.0.0/36fba834c6533853a24b6398207b3a1567455da505ceeed63bd94a5b7c6fd8b9)
```

```
Dataset({
    features: ['translations', 'talk_name'],
    num_rows: 258098
})
['ar', 'bg', 'de', 'el', 'en', 'es', 'eu', 'fa', 'fr', 'fr-ca', 'he', 'hr', 'hu', 'it', 'ja', 'ko', 'nb', 'nl', 'pl', 'pt', 'pt-br', 'ro', 'ru', 'sq', 'tr', 'vi', 'zh-cn', 'zh-tw']
```

Şekil 10. TED Konuşmalarının Kullanılması

TED konuşmalarını içeren bu veri seti içerisinde kullanmak istediğimiz dili ISO dil kodlarıyla birlikte kullanılmaktadır. ISO 639 dil kodları kendi içerisinde 1, 2, 3, 4, 5 ve 6 olarak bulunmaktadır. Bu ISO 639 kodlarının, kısa kodları ve kısa kodlarının açıklamalarına ait bilgi Tablo 2 içerisinde gösterilmiştir.

Tablo 2. ISO 639 Dil Kodları

Standart	ISO Açıklaması
ISO 639-1	Alfa-2 kodu
ISO 639-2	Alfa-3 kodu
ISO 639-3	Kapsamlı dil kapsamı için Alpha-3 kodu
ISO 639-4	Dil kodlaması için uygulama yönergeleri ve genel ilkeler
ISO 639-5	Dil aileleri ve grupları için Alfa-3 kodu
ISO 639-6	Dil varyantlarının kapsamlı kapsamı için Alfa-4 temsili

ISO-639 dil kodları dünya dillerine yönelik çeşitli referans türlerini kataloglamaya ve standartlaştırma fayda sağlamaktadır. Kataloglama, arşivleme ve makine işlemeye

yardımcı olarak dillere, dil gruplarına ve dil çeşitlerine referans olmayı kolaylaştırmayı amaçlamaktadır. Başka dillerdeki aynı kelime yazılışlarını ayırmak için kullanılmaktadır. Malzemelerin, ürünlerin, süreçlerin ve hizmetlerin tutarlı bir şekilde kullanılmasını sağlamaktadır (Morey, Post ve Friedman, 2013).

Ayrıca Datasets kütüphanesi içerisindeki özellikler bölümünde TED'in hangi konuşmasına ait olduğu bilgileri ve çeşitli dillerde çevirileri bulunmaktadır. Daha sonrasında veri setini eğitmek için belirli bir formata dönüştürmek gerekmektedir. Veri setinin eğitimi için kaynak cümleler ve bunların birebir çevirileri olan cümlelerden meydana gelmektedir. Veri setini oluşturmak için ilk önce kaynakta bulunan cümlelerin konumunu belirlemek ve sonrasında üzerinde iyileştirmek istediğimiz çeviriler belirlenmektedir. Şekil 11'de Datasets kütüphanesi içerisindeki TED konuşmalarının 60 dil arasındaki Türkçe ve İngilizce dillerinin dizinleri ekrana yazdırılması gösterilmektedir.

```
TRIndex = data[0]['translations']['language'].index('tr')

print ("Türkçe'nin Bulunduğu Index: ", TRIndex)

ENIndex = data[0]['translations']['language'].index('en')

print ("İngilizce'nin Bulunduğu Index :", ENIndex)
```

```
Türkçe'nin Bulunduğu Index:  24
İngilizce'nin Bulunduğu Index : 4
```

Şekil 11. TED Konuşmalarının Dilinin İndekslerinin Tespit Edilmesi

60 dil arasında Türkçe ve İngilizce dillerinin dizinleri tespit edilmiştir. Tespit edilen bu dizinlerle birlikte hem Türkçe hem de İngilizce cümlelerin dizinlerdeki ilk cümle elemanları örnek olması açısından ekrana yazdırılmaktadır. Şekil 12'de Datasets kütüphanesi içerisindeki TED konuşmalarının Türkçe ve İngilizce için ilk cümleleri gösterilmektedir.

```
TRsource = data[0]['translations']['translation'][TRIndex]

print(TRsource, "\n")

ENsource = data[0]['translations']['translation'][ENIndex]

print(ENsource)
```

```
Günümüzde mücadele ettiğimiz bütün eksiklikler arasında – öncelikle finansal ve ekonomik olanları düşünüyoruz – beni en çok ilg
ilendireni politik diyalog eksikliği – modern çatışmaları oldukları haliyle irdeleme becerimiz , varoluş nedeninin kaynağına gi
tmek ve kilit oyuncuları anlamak ve onlarla anlaşmak .

Amongst all the troubling deficits we struggle with today – we think of financial and economic primarily – the ones that concer
n me most is the deficit of political dialogue – our ability to address modern conflicts as they are , to go to the source of w
hat they &apos;re all about and to understand the key players and to deal with them .
```

Şekil 12. TED Konuşmalarının Tespit Edilen Örnek Cümleleri

Python programlama dilinin Datasets kütüphanesi içerisindeki TED konuşmalarına eğer dil kısıtlaması koymak istemiyorsak, İngilizce dilini kaynak dil olarak belirledikten sonra 60 dilde çevirisini elde etmek mümkündür. Şekil 13'te Python programlama dilinin Datasets kütüphanesi içerisindeki TED konuşmalarının dil sınırı olmadan İngilizce dilini kaynak olarak aldıktan sonra hedef olarak 60 dil için çevirisi mevcut olan veri setinin her dildeki ilk cümleleri gösterilmektedir.

```
pairs = []
for i, translation in enumerate(data[0]['translations']['translation']):
    if i != ENIndex:
        pairs.append((ENsource, translation))

print(pairs[0])
```

```
('Amongst all the troubling deficits we struggle with today – we think of financial and economic primarily – the ones that conc
ern me most is the deficit of political dialogue – our ability to address modern conflicts as they are , to go to the source of
what they &apos;re all about and to understand the key players and to deal with them .', 'من ضمن جميع المثبطات المقلقة التي نعاني منها اليوم نفكّ
ر في المقام الأول في الأمور المالية والاقتصادية وأكثر ما يهمني بشكل أكثر هو عجز الحوار السياسي – قدرتنا على فهم الصراعات الحديثة على ماهي عليه , بالذهاب الى اصلها الفعلي وعلى فهم
اللاعبين الرئيسيين وعلى التعامل معهم')
```

Şekil 13. TED Konuşmalarının Çeviriler

Bu tez çalışmasında kaynak olarak İngilizce dilini ve hedef olarak sadece Türkçe dili belirlenerek gerçekleştirilmiştir. Şekil 14'te Python programlama dilinin Datasets kütüphanesi içerisindeki TED konuşmalarının birebir çevirisi olarak İngilizce ve Türkçe dil karşılığı gösterilmektedir.

```
from sentence_transformers import InputExample
from tqdm.auto import tqdm

lang_list = ['tr']
train_samples = {f'en-{lang}': [] for lang in lang_list}

for row in tqdm(data):
    ENIndex = row['translations']['language'].index('en')
    ENsource = row['translations']['translation'][ENIndex].strip()
    for i, lang in enumerate(row['translations']['language']):
        if lang in lang_list:
            translation = row['translations']['translation'][i].strip()
            train_samples[f'en-{lang}'].append(
                ENsource+'\t'+translation
            )
```

```
100% 258098/258098 [01:25<00:00, 2910.29it/s]
```

```
print(ENsource+'\t'+translation)
```

```
( Applause )    ( Alkış )
```

Şekil 14. TED Konuşmalarının İngilizce ve Türkçe Örnekleri

Python programlama dilinin Datasets kütüphanesi içerisindeki TED konuşmalarının veri seti içerisindeki verilerin İngilizce cümleleri kaynak dosyası olarak verilmektedir. İngilizce cümlelerin birebir çevirisi olan Türkçe cümleler hedef olarak verilmektedir. İngilizce ve Türkçe cümleleri "Sekme ile Ayrılmış Değer" ile birleştirmek için "\t" operatörüyle birlikte birleştirme işlemi gerçekleştirilmiştir. Yine aynı şekilde çeşitli kaynaklardan oluşturmuş olduğumuz verilerden yani hazır olmayan bir veri seti oluşturulmuştur. Bu hazır olmayan veri seti içinde İngilizce cümleleri ve birebir çevirisi olan Türkçe cümleleri sekme ile ayrılmış değer ile birleştirme işlemiyle gerçekleştirilebilmektedir. Şekil 15'te Python programlama dilinin Datasets kütüphanesi içerisindeki TED konuşmalarının kullanılması gösterilmektedir.

```
import gzip
import os

if not os.path.exists('./data'):
    os.mkdir('./data')

for lang_pair in train_samples.keys():
    with gzip.open(f'./ted-train-{lang_pair}.tsv.gz', 'wt', encoding='utf-8') as f:
        f.write('\n'.join(train_samples[lang_pair]))
```

Şekil 15. TED Konuşmalarının Kullanılması

Sonrasında hazır veri seti kullanılarak oluşturulan veri setini eğitim için kaydetme işlemi gerçekleştirilmektedir. Bu işlemi yaparken "tsv.gz" formatında kaydedilmesi gerekmektedir çünkü cümle dönüştürücü üzerinde modelin eğitimi için bu formatta veri setine ihtiyaç bulunmaktadır.

3.15 Oluşturulan Model İçerisindeki Veri Setinin Kullanılması

Türkçe ve İngilizce cümleler arasında kaliteli birebir çevirisi yapılmış 10.323.997 adet İngilizce cümle ve 10.323.997 adet Türkçe cümle olmak üzere toplamda 20.647.992 adet cümle bulunmaktadır. Birebir çevirisi yapılmış kaliteli İngilizce ve Türkçe cümleler içerisinde Türkçe cümleler kendi içerisinde toplam 244.143.416 adet kelime bulunmaktadır. İngilizce cümleler kendi içerisinde toplam 269.310.676 adet kelime bulunmaktadır. Birebir çevirisi yapılmış İngilizce ve Türkçe cümlelerin kelime sayılarının toplamı ise 513.454.092 adet olarak hesaplanmıştır.

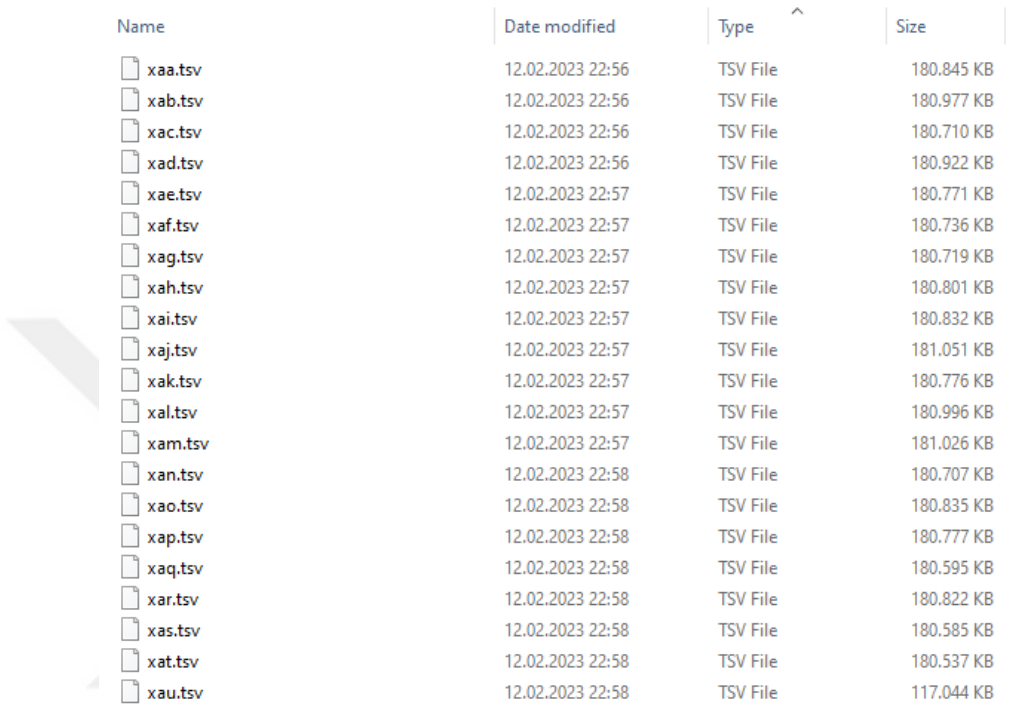
Birebir çevirisi yapılmış kaliteli İngilizce ve Türkçe cümleleri eğiterek yeni bir model oluşturulması sağlanmıştır. Oluşturulan bu model Türkçe – İngilizce cümleler için mevcut modellerin üzerinde bir başarı elde edilmesi amaçlanmıştır. Tablo 3 üzerinde İngilizce cümleler ve birebir çevirisi yapılmış Türkçe cümlelerle birlikte kelimelerin sayısı gösterilmiştir. İngilizce cümleler ve birebir çevirisi yapılmış Türkçe cümlelerle birlikte kelimelerin ve cümlelerin kendi aralarında toplam sayıları gösterilmektedir (Bañón ve diğerleri, 2022).

Tablo 3. Oluşturulacak Modelin Cümle ve Kelime Sayısı

	Cümle Sayısı	**Kelime Sayısı**
Türkçe	10.323.996	244.143.416
İngilizce	10.323.996	269.310.676
Toplam	20.647.992	513.454.092

Çeşitli kaynaklardan toplanan İngilizce ve birebir çevirisi yapılmış Türkçe cümleleri kendi içerisinde topladığımız zaman toplam 3.52 gigabayt Türkçe ve İngilizce cümleler içeren bir dosya elde edilmektedir. Türkçe ve İngilizce cümlelerden oluşturulan bu veri setini tek seferde okumak zor olacağı için toplamda 1.000.000 satır içeren dosyalara bölme işlemi süreci kolaylaştırmaktadır. Bölme işlemini yapmak için “split -l 1000000 dosya.tsv” komutundan faydalanarak her bir dosya içerisinde toplamda 1.000.000 satır

bulunan dosyalara bölmektedir. Şekil 16'da İngilizce ve birebir çevirisi yapılmış Türkçe cümlelerin her dosya içerisinde 1.000.000 satır bulunacak şekildeki dosyaların toplamda 21 dosyaya bölündüğünü ve bu dosyalar içerisindeki toplam cümlesi sayısı 20.647.992 elde ettiği gösterilmektedir.



Name	Date modified	Type	Size
xaa.tsv	12.02.2023 22:56	TSV File	180.845 KB
xab.tsv	12.02.2023 22:56	TSV File	180.977 KB
xac.tsv	12.02.2023 22:56	TSV File	180.710 KB
xad.tsv	12.02.2023 22:56	TSV File	180.922 KB
xae.tsv	12.02.2023 22:57	TSV File	180.771 KB
xaf.tsv	12.02.2023 22:57	TSV File	180.736 KB
xag.tsv	12.02.2023 22:57	TSV File	180.719 KB
xah.tsv	12.02.2023 22:57	TSV File	180.801 KB
xai.tsv	12.02.2023 22:57	TSV File	180.832 KB
xaj.tsv	12.02.2023 22:57	TSV File	181.051 KB
xak.tsv	12.02.2023 22:57	TSV File	180.776 KB
xal.tsv	12.02.2023 22:57	TSV File	180.996 KB
xam.tsv	12.02.2023 22:57	TSV File	181.026 KB
xan.tsv	12.02.2023 22:58	TSV File	180.707 KB
xao.tsv	12.02.2023 22:58	TSV File	180.835 KB
xap.tsv	12.02.2023 22:58	TSV File	180.777 KB
xaq.tsv	12.02.2023 22:58	TSV File	180.595 KB
xar.tsv	12.02.2023 22:58	TSV File	180.822 KB
xas.tsv	12.02.2023 22:58	TSV File	180.585 KB
xat.tsv	12.02.2023 22:58	TSV File	180.537 KB
xau.tsv	12.02.2023 22:58	TSV File	117.044 KB

Şekil 16. Eğitilecek Verinin Hazırlanması

Dosyalara ayırma işlemenin daha sonrasında Python programlama dilinin yardımıyla yazdığımız bir uygulama sayesinde tekrar ".tsv" formatına çevrilmektedir. ".tsv" formatına çevirme işlemiyle birlikte sekme ile ayrılmış değer formatı haline getirmektedir. Bu çevirme işlemini gerçekleştirirken her satır içerisinde "hem İngilizce hem de Türkçe karşılığı mevcut mu?" sorusunun karşılığını kontrol edilmektedir. Bu sayede veri seti içerisinde okuma yapılırken oluşabilecek karakter hataları gibi hataların önüne geçilmesi planlanmaktadır.

Birebir çevirisi yapılmış İngilizce ve Türkçe cümleleri sekme ile ayrılmış değer formatıyla ayırdıktan sonra kaydetme işlemi gerçekleştirilmektedir. Model eğitimi öncesindeki son aşamada ise "Gzip" sıkıştırması yapılmaktadır. Bu sıkıştırma işlemi ise eğitim setinin içerisinde bir sıkıştırma yaparak eğitim dosyalarını %90 oranı kadar

boyutlarında küçülme gerçekleştirmektedir. “Gzip” sıkıştırma işlemini gerçekleştirmek için Python programlama dilinin yardımıyla küçük ve basit bir uygulama yazarak bu işlemi her bir dosya için gerçekleştirilmektedir. Bu sıkıştırma uygulamasını yazabilmek için ilk önce “Gzip” kütüphanesi yüklenmektedir. Bu kütüphanenin yüklenmesiyle birlikte sonrasında “Compress” metodunun yardımıyla sıkıştırma işlemi gerçekleştirilmektedir. Böylece model eğitimi öncesinde ihtiyaç olan bütün adımları tamamlayarak bir sonraki adım olan eğitim sürecine başlamak için olan hazırlıklar gerçekleştirilmiştir.

3.16 Modelin Geliştirilmesi

İlgili modelin geliştirilmesi için birebir çevirisi yapılmış İngilizce ve Türkçe cümleler içerisinde bulunan ve daha öncesinde hazırlanmış olan hazır veri setiyle birlikte başka kaynaklardan toplanan verilerden oluşan veri setini cümle dönüştürücünün yorumlayabileceği bir formata getirilmesiyle birlikte model eğitimine başlanmaktadır. Şekil 17’de eğitilmesi için gerekli olan öğrenci modeli gösterilmektedir.

```
import os
from sentence_transformers import models, SentenceTransformer, ParallelSentencesDataset

xlmr = models.Transformer('xlm-roberta-base')
pooler = models.Pooling(
    xlmr.get_word_embedding_dimension(),
    pooling_mode_mean_tokens=True
)

student = SentenceTransformer(modules=[xlmr, pooler])

print(student)
```

```
Some weights of the model checkpoint at xlm-roberta-base were not used when initializing XLMRobertaModel: ['lm_head.dense.weigh
t', 'lm_head.bias', 'lm_head.layer_norm.bias', 'lm_head.layer_norm.weight', 'lm_head.dense.bias', 'lm_head.decoder.weight']
- This IS expected if you are initializing XLMRobertaModel from the checkpoint of a model trained on another task or with anoth
er architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing XLMRobertaModel from the checkpoint of a model that you expect to be exactly ide
ntical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).
SentenceTransformer(
  (0): Transformer({'max_seq_length': 512, 'do_lower_case': False}) with Transformer model: XLMRobertaModel
  (1): Pooling({'word_embedding_dimension': 768, 'pooling_mode_cls_token': False, 'pooling_mode_mean_tokens': True, 'pooling_mo
de_max_tokens': False, 'pooling_mode_mean_sqrt_len_tokens': False})
)
```

Şekil 17. Modelin Geliştirilmesi

Model eğitimi işlemi için ilk önce Python programlama dilinin “Sentence Transformers” kütüphanesini yükledikten ve ilgili tanımlamasını gerçekleştirdikten sonra “Transformer” metoduna “xlm-roberta-base” modeli parametre olarak verilmektedir. Bu model yerine BERTurk modeli tercih edilebilirdi. BERTurk modeli içerisinde 35 gigabayt veri

bulunmaktadır (Schweter, 2020). “xlm-roberta-base” modeli içerisinde 100’den fazla dil ve 2.5 terabayt eğitilmiş veri bulunmaktadır (Conneau ve diğerleri, 2019). Bu yüzden “xlm-roberta-base” modeli tercih edilmektedir. “Transformer” metodunun yardımıyla birlikte model eğitimi için ihtiyaç olan ilk katman oluşturulmaktadır. “Transformer” metodu içerisine parametre olarak model ismi veya model yolu, giriş değerleri, modele iletilen bağımsız değişkenler, önbellek dizini, cümleyi parçalarına ayırmak için gerekli değişkenler, modeli girişinin küçük harf yazılacağı ve cümleyi parçalarına ayırmak için kullanılacak uygulamanın dizini gibi parametreler bulunmaktadır. Tablo 4’te Python programlama dilinin “Sentence Transformers” kütüphanesinin “Transformer” metodunun içerisinde bulunan parametreler gösterilmektedir.

Tablo 4. Sentence Transformers – Transformers Metodu

Parametre Adı	Parametre Açıklaması
model_name_or_path	Hazır bir model veya diskteki bir model yüklenir.
max_seq_length	Belirtilen uzunluktan daha uzun olan tüm girdileri kısaltır.
model_args	Modele, anahtar ve değerleri modele iletilir.
cache_dir	Model dönüşümü için depolanan veya yüklenen önbellek dizini
tokenizer_args	Belirteç, anahtar ve değerleri modele iletilir.
do_lower_case	Bu değer “true” ise küçük harf yazar.
tokenizer_name_or_path	Belirtecin adı veya yolu eğer boş bırakılırsa “model_name_or_path” kullanılır.

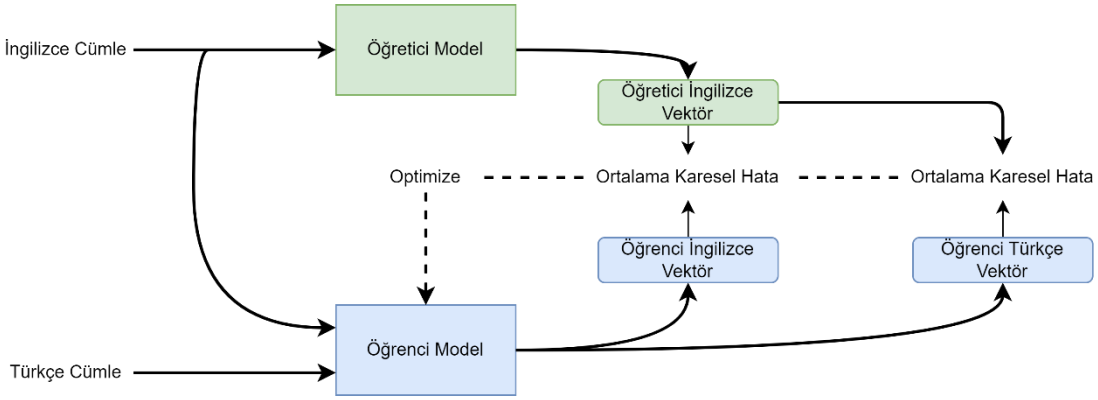
Transformer” metodu sonrasında Python programlama dilinin “Sentence Transformers” kütüphanesinin “Pooling” metodunun yardımıyla birlikte model eğitimi için ihtiyaç olan ikinci katman oluşturulmaktadır. İkinci katman sayesinde performans işlemi yani boyutları azaltarak ve gereksiz özellikleri daha az kullanarak önemli özelliklere odaklanmaktadır. Bu işlem sayesinde işlem gücü azalmaktadır. “Pooling” metodu içerisine gömülü kelimeler için boyut değerleri, performans ayarları, CLS belirteç değerleri bu değerler cümle düzeyinde sınıflandırmayı temsil etmektedir. Performans için kullanılacak maksimum belirteçler, ortalama performans belirteçleri ve ortalama performans karekök belirteçleri gibi değerleri içermektedir. Tablo 5’te Python

programlama dilinin “Sentence Transformers” kütüphanesinin “Pooling” metodunun içerdiği parametreler gösterilmektedir.

Tablo 5. Sentence Transformers – Pooling Metodu

Parametre Adı	Parametre Açıklaması
word_embedding_dimension	Kelime vektörlerinin boyutları
pooling_mode	Harf dizinin ortalama, maksimum ve ortak dil belirtimi
pooling_mode_cls_token	Metin gösterimlerinin ilk belirteci (ortak dil belirteci)
pooling_mode_max_tokens	Tüm belirteçler üzerinde her boyutta maksimum değeri
pooling_mode_mean_tokens	Tüm belirteçler üzerinde her boyutta ortalama değeri
pooling_mode_mean_sqrt_len_tokens	Tüm belirteçler üzerinde her boyutta ortalama değerinin verilen boyutta karekökü

Birebir çevirisi yapılmış Türkçe ve İngilizce cümlenin eş anlamları alındığı zaman çok dilli cümle dönüştürücüleri vektör karşılığı olarak her dil için ikisini de aynı vektör karşılığı gösterilmektedir. Türkçe ve İngilizce cümle çiftlerini içeren veri setimizi oluşturduktan sonra İngilizce cümleler hem öğretici model içerisine hem de öğrenci model içerisine beslenmektedir. Sonrasında diğer bir dil olan Türkçe cümleler öğrenci modeli içerisine beslenmektedir. Öğretici modeldeki vektör karşılığı ile öğrenci modeldeki vektör karşılıkları arasındaki karesel ortalama hata (MSE) kaybı hesaplanmaktadır. Öğrenci modeli ise bu MSE kaybı kullanılarak optimize edilmektedir (Chidambaram ve diğerleri, 2019). Şekil 18’de Python programlama dilinin “Sentence Transformers” kütüphanesinin öğretici ve öğrenci modellerinin basit bir şekilde mimarisi gösterilmektedir.



Şekil 18. Sentence Transformers – Öğretici ve Öğrenci Modeli

Tanımlamasını gerçekleştirmiş olduğumuz öğrenci model içerisine "Transformer" ve "Pooling" metotlarının değerleri parametre olarak tanımlanmaktadır ve sonrasında bu değişkeni yazdırdığımız zaman model hakkında çeşitli bilgiler vermektedir. Şekil 19'da Python programlama dilinin "Sentence Transformers" kütüphanesine ait çeşitli bilgiler gösterilmektedir.

```
teacher = SentenceTransformer('paraphrase-distilroberta-base-v2')

print(teacher)
```

```
SentenceTransformer(
  (0): Transformer({'max_seq_length': 256, 'do_lower_case': False}) with Transformer model: RobertaModel
  (1): Pooling({'word_embedding_dimension': 768, 'pooling_mode_cls_token': False, 'pooling_mode_mean_tokens': True, 'pooling_mo
de_max_tokens': False, 'pooling_mode_mean_sqrt_len_tokens': False})
)
```

Şekil 19. Sentence Transformers

Python programlama dilinin "Sentence Transformers" kütüphanesinin "ParallelSentencesDataset" metodunun yardımıyla birlikte birebir çevirisi yapılmış paralel cümleleri okumak için kullanılmaktadır. Bu paralel cümleleri okuyabilmek için cümlelerin arasında sekme ile ayrılmış değere sahip olması gerekmektedir. "Sentence Transformers" kütüphanesinin "ParallelSentencesDataset" metodunun ilk parametresi eğitilmesi gereken öğrenci modelidir. İkinci parametre ise veri kümesindeki ilk sütun içerisindeki cümle yerleştirmesini yapacak olan öğretici modelidir. Üçüncü parametre ise birebir çevirisi yapılmış paralel cümleleri kodlama yapmak için ayrılacak olan boyut

değerlerini içermektedir. ParallelSentencesDataset" metodunun en son parametre ise gömülü bellek kullanılacağı bilgisini içermektedir. Şekil 20'de Python programlama dilinin "Sentence Transformers" kütüphanesinin "ParallelSentencesDataset" metodunun içerdiği parametreler gösterilmektedir.

```
data = ParallelSentencesDataset(student_model=student, teacher_model=teacher, batch_size=32, use_embedding_cache=True)

print(data)
```

```
<sentence_transformers.datasets.ParallelSentencesDataset.ParallelSentencesDataset object at 0x0000029E7FEA7460>
```

Şekil 20. Sentence Transformers – Paralel Veri Seti

Bir sonraki aşamada ise verilecek olan dosya yolundaki okunacak maksimum cümle sayısıdır. Cümle içerisinde belirlenen karakter uzunluğundan fazla ise o örneği eğitime almayacak değerin atamasını gerçekleştirilmektedir. Bu noktada belirlenen sınırlara göre eğitim işlemi gerçekleştirilecektir. Şekil 21'de Python programlama dilinin "Sentence Transformers" kütüphanesinin ayarları gösterilmektedir.

```
max_sentences_per_language = 500000
train_max_sentence_length = 250
```

Şekil 21. Sentence Transformers – Ayarlar

İlgili klasördeki veri setini veya veri setinin içerisinde bulunan verilerinin yüklemesi gerçekleştirilmektedir. Şekil 22'de Python programlama dilinin "Sentence Transformers" kütüphanesinin veri yüklemesi gösterilmektedir.

```
train_files = [f for f in os.listdir('./data') if 'train' in f]
for f in train_files:
    print(f)
    data.load_data('./data/' + f, max_sentences=max_sentences_per_language,
                   max_sentence_length=train_max_sentence_length)
```

```
ted-train-en-tr.tsv.gz
```

Şekil 22. Sentence Transformers – Veri Setinin Yüklenmesi

Veri setini yüklendikten sonra Python programlama dilinin “PyTorch” kütüphanesinin veri yükleyicisine aktarılmaktadır. Şekil 23’te Python programlama dilinin “PyTorch” kütüphanesinin veri yükleme işlemi gösterilmektedir.

```
from torch.utils.data import DataLoader

loader = DataLoader(data, shuffle=True, batch_size=32)
```

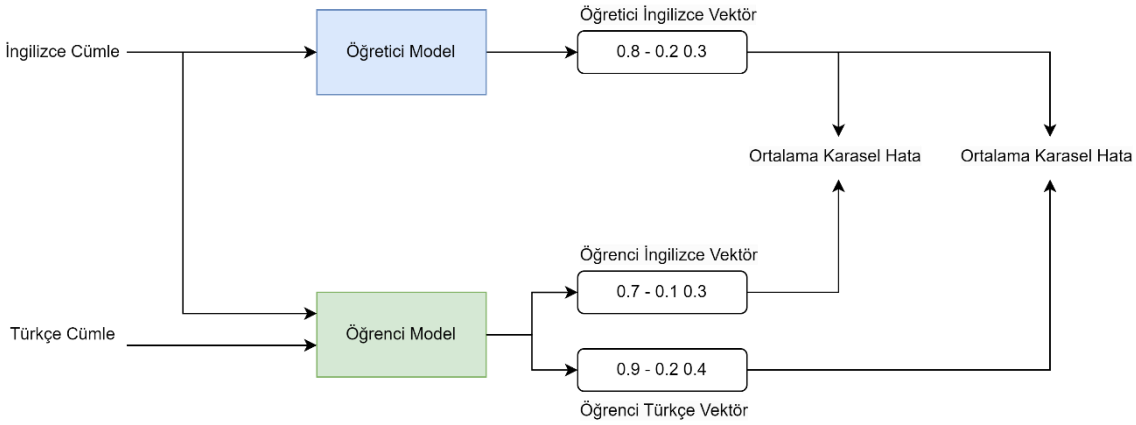
Şekil 23. Sentence Transformers – Veri Yükleyicisi

Hesaplanan cümle yerleştirme ile hedef cümle yerleştirme arasındaki ortalama hata karesi (MSE) kaybını hesaplamaktadır. MSE, kayıp fonksiyonu olarak adlandırılmaktadır ve bu fonksiyon bu tez çalışmasında kullanılan veri seti içerisindeki beklenen bazı değerleri en aza indirilerek regresyon görevi görmektedir. MSE kaybını hesaplamak için öğretici modeli N olarak adlandırılmaktadır ve bir veya birden fazla vektör karşılığı olan dil veri setini s olarak adlandırılmaktadır. Birebir çevirisi yapılmış dil dosyaları ((s_1, t_1),……., (s_n, t_n)) olarak adlandırılmaktadır. Kaynak dosyadaki dil çevirileri s_n olarak adlandırılmaktadır. Hedef dosyadaki dil çevirileri ise t_n olarak adlandırılmaktadır. Öğrenci modeldeki veri seti ise Ñ olarak adlandırılmaktadır. $\tilde{N}(s_i) \approx N(s_i)$ olarak adlandırılmaktadır. Verilen veri setinin sınırlığı ise Š olarak adlandırılmaktadır. MSE kaybını hesaplama formülü denklem 3.4’te gösterilmektedir.

$$MSE = \frac{1}{|\check{S}|}\sum_{i=\check{S}}^{n}[N(s_i) - \tilde{N}(s_i))^2 + (N(s_i) - \tilde{N}(t_i))^2] \quad (3.4)$$

Ñ ile N arasında büyüklüğü ve yapısı aynı olması gerekmektedir. Öğrenci modeli Ñ ile temsil edilmektedir ve öğretici model ise N olarak temsil edilmektedir. Eğitilmiş modeli ise Ñ ← N olarak gösterebilmektedir (Reimers ve Gurevych, 2020).

Şekil 24’te öğretici ve öğrenci modeller üzerindeki ortalama hata karesi hesaplamasının genel bir mimarisi gösterilmektedir.



Şekil 24. Karesel Ortalama Hata Hesaplaması

Geliştirilecek olan modelin eğitimine başlamadan önce Python programlama dilinin "Sentence Transformers" kütüphanesinin "MSELoss" metodu yardımıyla ortalama hata karesi hesaplaması gerçekleştirilmektedir. Şekil 25'te "Sentence Transformers" kütüphanesinin "MSELoss" metodunu kullanabilmek için "sentence_transformers" kütüphanesinin kullanılması ve çağrılma adımı gösterilmektedir.

```
from sentence_transformers import losses

loss = losses.MSELoss(model=student)
```

Şekil 25. Sentence Transformers – Karasel Ortalama Hata Hesaplama Metodu

Modelin eğitimine başlamak için gerekli olan Python programlama dilinin "Sentence Transformers" kütüphanesinin "Fit" metodunun parametreleri Tablo 6'da detaylı bir şekilde gösterilmektedir.

Tablo 6. Sentence Transformers – Fit Metodu

Parametre Adı	Parametre Açıklaması
train_objectives	Çok görevli öğrenme için birden fazla Veri Yükleyici ve Kayıp Fonksiyonu çiftlerinin değeri
evaluator	Eğitim sırasında geçici olarak tutulan geliştirme verileri üzerinde model performansını değerlendirir ve diske kaydedilen en iyi modeli belirlemek için kullanılmaktadır.
epochs	Eğitim işlemi için eğitim tur sayısı
steps_per_epoch	Eğitim işlemi için tur sayısı başına adım sayısı. Değer varsayılan olarak ayarlanırsa Veri Yükleyici boyutuna eşittir.
scheduler	Öğrenme ortamı zamanlayıcısı
warmup_steps	Öğrenme oranı 0'dan maksimum öğrenme oranına kadar artırılır. Bu kadar çok eğitim adımından sonra, öğrenme oranı doğrusal olarak sıfıra düşürülür.
optimizer_class	Optimize sınıfı
optimizer_params	Optimize parametreleri
weight_decay	Model parametreleri için ağırlık azaltması
evaluation_steps	Eğer 0'dan büyük ise her adım sayısından sonra değerlendiriciyi kullanarak modeli değerlendirmektedir.
output_path	Model ve değerlendirme dosyaları için depolama yolu
save_best_model	Değerlendiriciye göre en iyi model verilen adreste saklanmaktadır.
max_grad_norm	Gradyan normalizasyonu için kullanılmaktadır.
use_amp	Otomatik Karışık Hassasiyet (AMP) kullanılmaktadır.
callback	Her değerlendirmeden sonra çağrılan geri çağırma işlevi
show_progress_bar	İlerleme çubuğu çıktısı
checkpoint_path	Eğitim sırasında kontrol noktalarının kaydedileceği klasör
checkpoint_save_steps	Belirli bir adım sonra bir kontrol noktası olarak kaydetmektedir.
checkpoint_save_total_limit	Depolanacak toplam kontrol noktası sayısı

Geliştirilecek olan modelin eğitimine başlamak için Python programlama dilinin "Sentence Transformers" kütüphanesinin "Fit" metodunun yardımıyla ilgili model üzerinden eğitim işlemi gerçekleştirmektedir. Şekil 26'da "Sentence Transformers"

kütüphanesinin “Fit” metodunun nasıl çağrıldığını ve nasıl kullanıldığının süreçleri gösterilmektedir.

```
from sentence_transformers import evaluation
import numpy as np

epochs = 1
warmup_steps = int(len(loader) * epochs * 0.1)

student.fit(
    train_objectives=[(loader, loss)],
    epochs=epochs,
    warmup_steps=warmup_steps,
    output_path='./xlmr-ted',
    optimizer_params={'lr': 2e-5, 'eps': 1e-6},
    save_best_model=True,
    show_progress_bar=False
)
```

Şekil 26. Sentence Transformers – Fit

3.17 Cümle Eşleme

Bu tez çalışmasında derin öğrenme yönteminin yardımıyla İngilizce ve Türkçe dilleri arasında birebir çevirileri yapılmış fakat sıralarının herhangi bir şekilde değişmesi sonucu tekrar cümle eşleştirme adımını gerçekleştirmektedir. İlk önce kaynak veri seti için eğitilen model ile uzay denkleminde yer alan ilgili cümlenin vektör karşılığını bulunmaktadır. Sonrasında hedef veri seti için eğitilen model ile uzay denkleminde yer alan ilgili cümlenin vektör karşılığını bulunmaktadır. Vektör karşılığını bulduktan sonra kaynak ve hedef veri seti cümlelerinin uzay denkleminde bulunan vektör karşılıklarını kosinüs benzerlikleriyle hesaplanmaktadır. Gerçekleştirilen bu hesaplama sonrasında en yüksek skora sahip cümlelerin birebir çevirisi yapılmış doğru olan cümlelerin karşılığı olarak kabul edilmektedir.

Cümle eşleme işlemi için eğitmiş olduğumuz modeli test etmek için geliştirilen modelle birlikte 1.000 adet İngilizce cümle ve bu cümlelerin birebir çevirisi yapılmış ama sıralamaları rastgele değiştirilmiş olan 1.000 adet Türkçe cümlelerden oluşan test veri setinin eşleştirme işlemi ile gerçekleştirilmektedir. Tablo 7’de cümle eşleştirme işlemi için geliştirilen modeli test edebilmek için birebir çevirisi yapılmış İngilizce ve Türkçe test cümlelerinin ve kelimelerinin sayısıyla birlikte kelimelerin ve cümlelerin kendi aralarında toplam sayıları da gösterilmektedir.

Tablo 7. Cümle Sayısı ve Kelime Sayısı

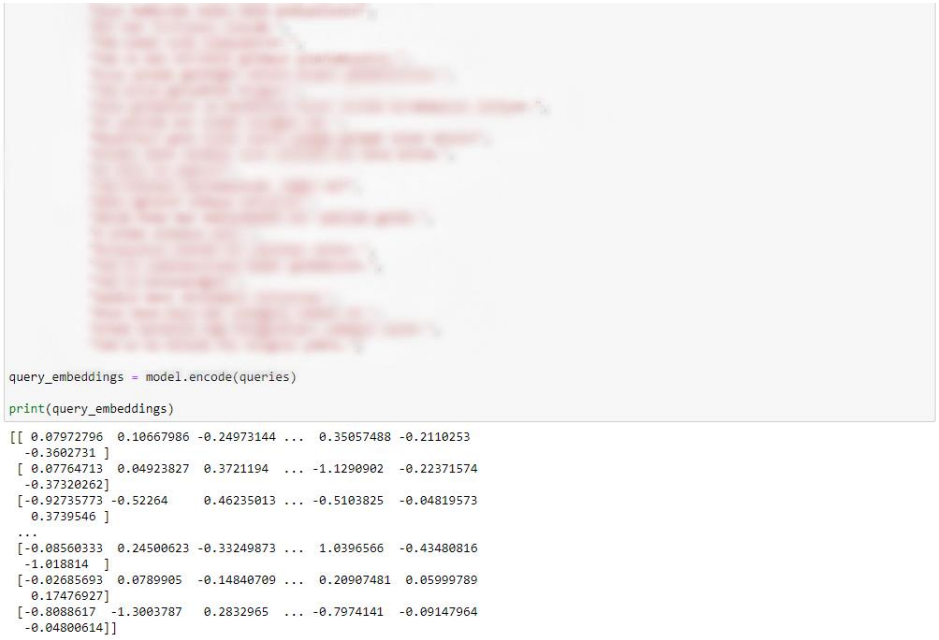
	Cümle Sayısı	**Kelime Sayısı**
Türkçe	1.000	4.793
İngilizce	1.000	6.558
Toplam	2.000	11.351

Test süreçlerini gerçekleştirmek için 1.000 adet İngilizce cümle veri setinin içerisinde toplam 4.793 adet İngilizce kelime bulunmaktadır. Test süreçlerini gerçekleştirmek için 1.000 adet Türkçe cümle veri setinin içerisinde ise toplam 6.558 adet Türkçe kelime bulunmaktadır. Türkçe ve İngilizce cümle veri setlerinin toplamı olan 2.000 adet cümle içerisinde hem Türkçe hem de İngilizce olmak üzere toplam 11.351 adet kelime bulunmaktadır.

Eğitim sürecini gerçekleştirdiğimiz modeli kullanarak cümle eşleştirme işlemi gerçekleştirilmektedir. Eğitmiş olduğumuz modeli kullanabilmek için "Sentence Transformers" kütüphanesinden faydalanılmaktadır. Bu kütüphane sayesinde daha önceden eğittiğimiz modelin yolu gösterilerek kütüphane içerisine bu model içerisine yüklenmektedir. Sonrasında ilk olarak İngilizce metinler içerisinden her bir cümleye ait vektör karşılığını hesaplama işlemi gerçekleştirilmektedir. Herhangi bir cümlenin veya kelimenin doğal dil işleme alanında yorumlanabilmesi için veya bilgisayar tarafından anlamlandırabilmesi için verinin sayısallaştırılmasına ihtiyaç duyulmaktadır. Vektör karşılıklarını bulmak için fastText, Word2Vec gibi hazır modellerde kullanılabilmektedir fakat bu modeller kullanıldığı zaman bir kelimenin birden fazla anlamı ortaya çıkabilmektedir. Bu sebepten dolayı bu modellerde vektör karşılığı hesaplandığı zaman tek bir anlam çıkarmaktadır. Bunun yanı sıra kelimenin konumunu dikkate almamaktadır. Bu modeller kelime seviyesinde işlem yapmaktadır. Cümle eşleştirme yapmak için cümle seviyesinde işlem yapmak daha önemlidir.

Bu sebepten dolayı her bir İngilizce ve Türkçe cümlenin vektör karşılığını bulmak aslında veriyi sayısallaştırdığımız anlamına gelmektedir. İngilizce cümleler içerisindeki her cümlenin vektör karşılığını bulduktan sonra birebir çevirisi yapılmış ama herhangi bir

şekilde sırası değiştirilmiş veya rastgele sıralanmış Türkçe cümleler içinde her bir cümlenin vektör karşılığını hesaplama işlemi gerçekleştirilmektedir. Şekil 27'de test edeceğimiz İngilizce veya Türkçe cümlelerin vektör karşılıkları hesaplandıktan sonraki vektör karşılık değerleri gösterilmektedir.



Şekil 27. Cümlelerin Vektör Karşılıklarının Bulunması

Test edeceğimiz İngilizce ve Türkçe cümlelerin vektör karşılıkları hesaplandıktan sonra İngilizce cümlelerin hesaplanmış olan vektör karşılıklarıyla ve Türkçe cümlelerin hesaplanmış olan vektör karşılıklarının arasında teker teker kosinüs benzerliklerinin hesaplanması işlemi gerçekleştirilmektedir. Bu işlem ilk olarak ilk İngilizce cümlenin hesaplanmış olan vektör karşılığını alır ve herhangi bir şekilde sırası değiştirilmiş veya rastgele sıralanmış Türkçe cümlelerin hesaplanmış vektör karşılıklarını teker teker karşılaştırmaktadır.

İlk İngilizce cümlenin hesaplanmış vektör karşılığı için 1.000 adet Türkçe cümlenin hesaplanmış vektör karşılığı karşılaştırılmaktadır. Sonrasında bir sonraki İngilizce cümlenin hesaplanmış vektör karşılığına geçmektedir ve yine bu işlem 1.000 adet Türkçe cümlenin hesaplanmış vektör karşılığı için karşılaştırılmaktadır. Bu işlem 1.000 adet İngilizce cümlenin hesaplanmış vektör karşılığı listesi boyunca devam etmektedir. Bu

işlemler test verisinin Türkçe ve İngilizce sayısının kendi arasında çarpımıyla ne kadar bir karşılaştırma yapacağı hesaplanmaktadır. 1.000 adet İngilizce ve 1.000 adet Türkçe cümle olduğu için bu işlemler 1.000 × 1.000 olmak üzere toplamda 1.000.000 adımda gerçekleştirilecektir. İngilizce ve Türkçe vektörler arasında karşılaştırılan bu sonuçlar 1 sayısına yaklaştıkça en yakın veya doğru bir eşleştirme işlemi gerçekleştirilmektedir. Hesaplanan bu değer 1 sayısından 0 sayısına doğru yaklaştıkça ise en ilgisiz veya alakasız cümle olduğu anlamına gelmektedir.

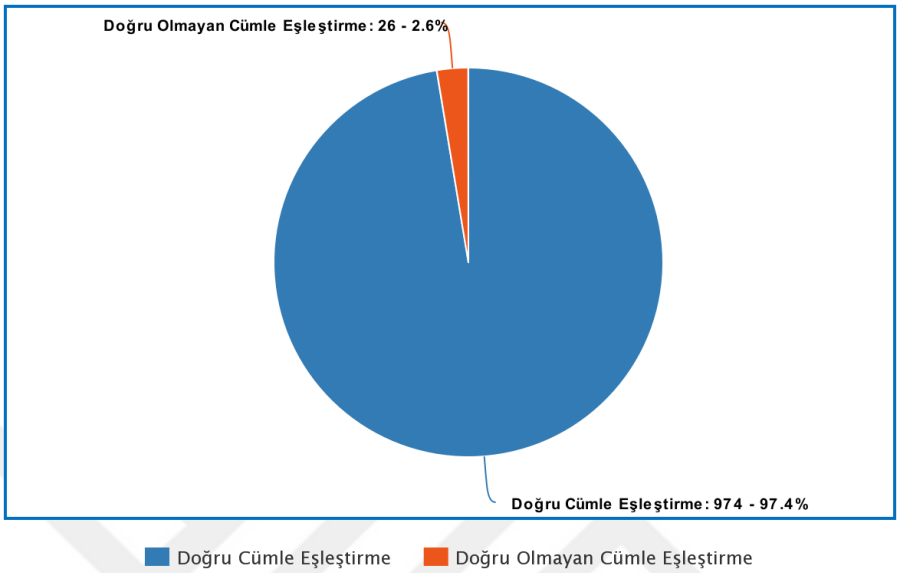
4. BULGULAR VE TARTIŞMA

Bu tez çalışmasında önerilen yöntem derin öğrenme yöntemle birlikte birebir çevirisi yapılmış olan İngilizce ve Türkçe cümlelerin eşleştirme işlemi gerçekleştirilmektedir. Hali hazırda kullanılan mevcut modellerin yanı sıra geliştirilen modelin mevcut modellerden daha başarılı olması hedeflenmektedir. İlgili modeli oluşturmak için hazır veri setlerinin yanı sıra internet üzerinden çeşitli kaynaklardan İngilizce ve Türkçe çevirisi yapılmış kaliteli cümleleri bir araya toplanmaktadır. Toplanan cümleler belirli filtrelerden ve aşamalardan geçirilmektedir. Mevcut modellerden daha başarılı olması için çeşitli deneme süreçleri gerçekleştirilmiştir. Birbirlerinden farklı veri setleri oluşturulduktan sonra eğitim işlemini gerçekleştirilmektedir. Sonuç olarak toplamda beş farklı model eğitilmiştir. Sonrasında oluşturulmuş olan beş farklı modelin başarıları kendi aralarında test edilmiştir. Her veri setinin hazırlanması ve eğitilmesi için geçen süre en az bir hafta en fazla ise iki hafta içerisinde gerçekleştirilmektedir. Eğitim süreçleri Intel Core™ I5-10400T CPU @ 2.00 GHz işlemci, 16 GB bellek ve 500 GB SSD sabit diski donanımlarına sahip Microsoft Windows 10 işletim sistemi üzerinde gerçekleştirilmiştir. Bu süre de veri setinin hazırlanması ve eğitim süreci gibi süreçler gerçekleştirilmiştir.

En son eğitim işlemini gerçekleştirilmiş olduğumuz veri seti içerisinde 10.000.000'dan fazla birebir çevirisi yapılmış İngilizce ve Türkçe kaliteli cümleler bulunmaktadır. Cümle eşleştirme işlemindeki başarısını test etmek amacıyla, oluşturulan 1.000 adet İngilizce cümle ile birebir çevirisi yapılmış ancak farklı sıralamalarda olan 1.000 adet Türkçe cümle içermektedir. Toplamda 2.000 cümle içeren İngilizce ve Türkçe veri kümesi oluşturulduktan sonra cümle eşleştirme işlemi gerçekleştirilerek başarı oranları gözlemlenmiştir.

En son eğitmiş olduğumuz modelle birlikte cümle eşleştirme işlemi gerçekleştirildiğinde diğer modellerden daha fazla başarılı cümle eşleştirme yaptığı gözlemlenmiştir. Cümle eşleştirme işlemi için diğer modellerde ve geliştirilen bu model içerisinde kullanılan test veri seti içerisinde geliştirilen modelin başarısı test edilmiştir. Test süreçleri 1.000 adet cümleden 974 tanesini doğru cümle eşleştirme yaparak gerçekleştirmiştir ve 1.000 adet cümleden geriye kalan 26 tanesini ise doğru olmayan eşleştirme yaparak

gerçekleştirmiştir. Cümle eşleştirme işlemi için geliştirilen modelle birlikte %97.4 oranında başarılı elde edilmiştir. Şekil 28'de cümle eşleştirme işlemine ait doğru cümle eşleştirme ve doğru olmayan cümle eşleştirme bilgileri verilmektedir.



Şekil 28. Cümle Eşleştirme Başarısı

Geliştirilen bu model sayesinde diğer hazır modellerin mevcutta eğitilmiş modellerden %0.72 oranında daha fazla doğru cümle eşleştirme işlemi yaparak mevcut modelden daha başarılı bir model elde edilmektedir.

1.000 adet Türkçe ve 1.000 adet İngilizce cümle içeren test veri seti içerisinde kullandığımız verileri 10 kategori başlığı içerisinde kategorize edilmiştir. En popüler 10 kategoriyi araştırdığımızda Teknoloji, Seyahat, Sağlık, Spor, Yiyecek ve İçecek, Moda, Ev ve Bahçe, Evcil Hayvanlar, İş Dünyası, Kültür ve Sanat olmak üzere karşımıza çıkmaktadır. Cümle eşleştirme işlemi için oluşturulan 1.000 adet İngilizce cümleyi kendi içerisinde 10 kategori başlığı altında değerlendirilmiştir. Aşağıdaki Tablo 8 içerisinde değerlendirilen 1.000 adet cümlenin kategori başlıkları ve cümle sayılarına ait detaylar verilmektedir.

Tablo 8. Kategori Başlığı ve Cümle Sayısı

Kategori Başlığı	Cümle Sayısı
Teknoloji	230
Seyahat	171
Sağlık	141
Spor	96
Yiyecek ve İçecek	74
Moda	72
Ev ve Bahçe	70
Evcil Hayvanlar	59
İş Dünyası	48
Kültür ve Sanat	39
Toplam Cümle Sayısı	1.000

Test verileri içerisindeki bu cümleleri 10 popüler kategori altında incelediğimizde ilk sırada teknoloji kategorisi yer almaktadır. 1.000 cümlenin içerisinde toplam 230 adet cümle teknoloji kategorisinde, 1.000 cümlenin içerisinde toplam 171 tanesi seyahat kategorisinde, 1.000 cümlenin içerisinde toplam 141 tanesi sağlık kategorisinde, 1.000 cümlenin içerisinde toplam 96 tanesi spor kategorisinde, 1.000 cümlenin içerisinde toplam 74 tanesi yiyecek ve içecek kategorisinde, 1.000 cümlenin içerisinde toplam 72 tanesi moda kategorisinde, 1.000 cümlenin içerisinde toplam 70 tanesi ev ve bahçe kategorisinde, 1.000 cümlenin içerisinde toplam 59 tanesi evcil hayvan kategorisinde, 1.000 cümlenin içerisinde toplam 48 tanesi iş dünyası kategorisinde, 1.000 cümlenin içerisinde toplam 39 tanesi kültür ve sanat kategorisinde yer almaktadır. Aşağıdaki Tablo 9 içerisinde 26 tane doğru olmayan eşleştirme gerçekleştirmeyen cümlenin kategori başlıkları ve cümle sayılarına ait detaylar verilmektedir.

Tablo 9. Doğru Eşleştirme Olmayan Kategori Başlığı ve Cümle Sayısı

Kategori Başlığı	Cümle Sayısı
Teknoloji	5
Seyahat	5
Sağlık	3
Spor	4
Yiyecek ve İçecek	4
Moda	2
Ev ve Bahçe	0
Evcil Hayvanlar	2
İş Dünyası	0
Kültür ve Sanat	1
Toplam Cümle Sayısı	26

5. SONUÇ ve ÖNERİLER

Gerçekleştirilen tez çalışmasında doğal dil işleme alanında derin öğrenme yöntemi kullanılmıştır. İngilizce cümleler ve bu cümlelerin birebir çevirileri yapılmış Türkçe cümlelerin sırasının değişmesiyle birlikte yeniden cümleleri eşleştirme işlemi için ihtiyaç duyulmaktadır. Sonrasında cümle eşleştirme işlemi için olan insan gücü ihtiyacını ortadan kaldırmak veya en aza indirmek için cümle eşleştirme işlemi gerçekleştirilmiştir. İngilizce ve Türkçe cümleler arasında çok fazla akademik çalışma bulunmamaktadır. Bunun sebebi Türkçe'nin morfolojik olarak dil yapısının İngilizce'den farklı olmasından kaynaklanmaktadır.

Oluşturulan farklı veri setleriyle birlikte model eğitim süreçlerinin çeşitli denemeleri gerçekleştirilmiştir. Hazır veri setiyle birlikte oluşturulan veri seti içerisindeki Türkçe ve İngilizce cümleleri derin öğrenme yöntemi yardımıyla eğiterek yeni bir model oluşturduktan sonra ilgili cümlelerin vektör karşılıkları hesaplanmıştır. Hesaplanan vektör karşılıklarının kosinüs benzerliği yardımıyla cümle eşleştirme işleminin süreçleri gerçekleştirilmiştir. Cümle eşleştirme işlemini 1.000 adet İngilizce ve 1.000 adet Türkçe cümle arasında test edilmiştir. Cümle eşleştirme işlemi için mevcut en başarılı dört model arasından en başarılı olanı kullanılarak 1.000 adet cümlenin 967 tanesini doğru cümle eşleştirme gerçekleştirmiştir ve 1.000 adet cümleden geriye kalan 33 tanesini doğru olmayan eşleştirme yaparak gerçekleştirmiştir. Mevcut model için en başarılı olan cümle eşleştirme oranı %96.7 olarak hesaplanmıştır. Yeni modeli geliştirmek için çeşitli kaynaklardan kaliteli cümleler toplanmıştır. Toplanan bu cümleleri belirli filtrelerden ve aşamalardan geçirdikten sonra yeni modeli oluşturmak için eğitim işlemini gerçekleştirilmiştir. Oluşturulan bu modelle birlikte cümle eşleştirme işlemi için yapılan testler sonucunda 1.000 adet cümle içerisinden 974 adet doğru cümle eşleştirmesi ve 1.000 adet cümle içerisinden 26 adet doğru olmayan cümle eşleştirmesi gerçekleştirilmiştir. Geliştirilen bu modelle yapılan testler sonucunda başarılı olan cümle eşleştirme oranı %97.4 olarak hesaplanmıştır. Geliştirilen bu model ile birlikte daha öncesinden eğitilmiş olan mevcut diğer modellere göre kendi içerisinde 1.000 adet İngilizce ve 1.000 adet Türkçe cümle içeren test verisi üzerinde %0.72 oranında başarıya

sahiptir. Bu başarı mevcut modeller arasındaki en başarılı olan “paraphrase-multilingual-mpnet-base-v2” modeline göre daha fazla başarı elde edilmiştir.

Sonuç olarak bu çalışmayla birlikte İngilizce - Türkçe cümle eşleştirme işlemi için 10 milyondan fazla kaliteli veriyi topladıktan sonra belirli filtrelerden ve aşamalardan geçirilmektedir. Günlerce süren eğitim işlemi gerçekleştirilmiştir. Daha sonrasında mevcut modeller ile geliştirilen bu model kendi aralarındaki başarıları test edilmiştir. Yapılan testler sonucunda mevcut modellerden daha başarılı bir model elde edildiği gözlemlenmektedir. Bu geliştirilen yeni model sayesinde veya bu geliştirilen yeni modeli daha fazla geliştirdikten sonra intihal uygulamaları gibi farklı dillerdeki intihalleri yakalamak için kullanabilmenin önü açılmaktadır. Çünkü farklı dillerdeki intihalleri yakalamak zordur ancak cümle eşleştirme süreci için geliştirilen bu model sonrasında daha büyük bir veri seti içerisindeki birebir çevirisi yapılmış olan İngilizce ve Türkçe cümleleri eğitilerek farklı dillerdeki intihalleri tespit edebilmek için çok dilli intihal uygulamaları geliştirmenin önü açılmaktadır.

KAYNAKLAR

Allen, J. F. (2003). Natural language processing. *Encyclopedia of Computer Science* içinde (ss. 1218-1222). GBR: John Wiley and Sons Ltd.

Balli, C., Guzel, M. S., Bostanci, E. ve Mishra, A. (2022). Sentimental Analysis of Twitter Users from Turkish Content with Natural Language Processing. *Computational Intelligence and Neuroscience, 2022*, 2455160. doi:10.1155/2022/2455160

Bañón, M., Esplà-Gomis, M., Forcada, M. L., García-Romero, C., Kuzman, T., Ljubešić, N., … Zaragoza, J. (2022). Turkish-English parallel corpus MaCoCu-tr-en 1.0. 02 Haziran 2023'te http://hdl.handle.net/11356/1520 adresinden erişildi.

Braune, F. ve Fraser, A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* içinde , COLING '10 (ss. 81-89). USA: Association for Computational Linguistics.

Brown, P. F., Lai, J. C. ve Mercer, R. L. (1991a). Aligning Sentences in Parallel Corpora. *29th Annual Meeting of the Association for Computational Linguistics* içinde (ss. 169-176). ACL 1991, sunulmuş bildiri, Berkeley, California, USA: Association for Computational Linguistics. doi:10.3115/981344.981366

Brown, P. F., Lai, J. C. ve Mercer, R. L. (1991b). Aligning sentences in parallel corpora. *Proceedings of the 29th annual meeting on Association for Computational Linguistics* içinde , ACL '91 (ss. 169-176). USA: Association for Computational Linguistics. doi:10.3115/981344.981366

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., … Kurzweil, R. (2018, 12 Nisan). Universal Sentence Encoder. arXiv. doi:10.48550/arXiv.1803.11175

Chen, S. F. (1993). Aligning Sentences in Bilingual Corpora Using Lexical Information. *31st Annual Meeting of the Association for Computational Linguistics* içinde (ss. 9-16). ACL 1993, sunulmuş bildiri, Columbus, Ohio, USA: Association for Computational Linguistics. doi:10.3115/981574.981576

Chidambaram, M., Yang, Y., Cer, D., Yuan, S., Sung, Y.-H., Strope, B. ve Kurzweil, R. (2019). Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model.

Chollet, F. (2021). *Deep Learning with Python, Second Edition*. Simon and Schuster.

Chousa, K., Nagata, M. ve Nishino, M. (2020). SpanAlign: Sentence Alignment Method based on Cross-Language Span Prediction and ILP. *Proceedings of the 28th International Conference on Computational Linguistics* içinde (ss. 4750-4761). COLING 2020, sunulmuş bildiri, Barcelona, Spain (Online): International Committee on Computational Linguistics. doi:10.18653/v1/2020.coling-main.418

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., … Stoyanov, V. (2019). Unsupervised Cross-lingual Representation Learning at Scale. *CoRR, abs/1911.02116*. 20 Haziran 2023'te http://arxiv.org/abs/1911.02116 adresinden erişildi.

Dagan, I., Church, K. ve Gale, W. (1993). Robust Bilingual Word Alignment for Machine Aided Translation. *Very Large Corpora: Academic and Industrial Perspectives* içinde 15 Haziran 2023'te https://aclanthology.org/W93-0301 adresinden erişildi.

Dale, R., Moisl, H. ve Somers, H. (Ed.). (2000). *Handbook of Natural Language Processing*. Marcel Dekker.

Demirhan, A., Kiliç, Y. A. ve Güler, İ. (2010). Tıpta Yapay Zeka Uygulamaları, 11.

Fung, P. ve McKeown, K. (1994). Aligning Noisy Parallel Corpora Across Language Groups: Word Pair Feature Matching by Dynamic Time Warping. *Proceedings of the First Conference of the Association for Machine Translation in the Americas* içinde . AMTA 1994, sunulmuş bildiri, Columbia, Maryland, USA. 11 Haziran 2023'te https://aclanthology.org/1994.amta-1.11 adresinden erişildi.

Gale, W. A. ve Church, K. W. (1993a). A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, *19*(1), 75-102.

Gale, W. A. ve Church, K. W. (1993b). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, *19*(1), 75-102.

Gómez Sánchez, J. (2021, 25 Ocak). Analysis and comparison of text similarity measures. 12 Haziran 2023'te https://upcommons.upc.edu/handle/2117/343180 adresinden erişildi.

Gungor, T. (2006). Compiling a Turkish-English Bilingual Corpus and Developing an Algorithm for Sentence Alignment. *Third International Bulgarian-Turkish Conference on Computer Science*, 291-296.

Halácsy, P., Kornai, A., Nagy, V., Németh, L. ve Trón, V. (2007). Parallel corpora for medium density languages. *Recent Advances in Natural Language Processing IV* içinde (ss. 247-258). doi:10.1075/cilt.292.32var

Hamawand, Z. (2020). *Modern Schools of Linguistic Thought: A Crash Course*. Springer.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357-362. doi:10.1038/s41586-020-2649-2

Haruno, M. ve Yamazaki, T. (1996). High-performance bilingual text alignment using statistical and dictionary information. *Proceedings of the 34th annual meeting on*

Association for Computational Linguistics içinde , ACL ’96 (ss. 131-138). USA: Association for Computational Linguistics. doi:10.3115/981863.981881

Kay, M. ve Roscheisen, M. (1993). Text-Translation Alignment. *Computational Linguistics*, *19*(1), 121-142.

Khurana, D., Koli, A., Khatter, K. ve Singh, S. (2022). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, 1-32. doi:10.1007/s11042-022-13428-4

Kızılırmak, E. (2020). *İngilizce-Türkçe çeviri metinlerinde levenshtein uzaklığı ile desteklenmiş çapa tabanlı cümle eşleme*. (Yayın no: 625078). [Yüksek Lisans Tezi, Maltepe Üniversitesi]. https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp

Kumar, E. (2013). *Natural Language Processing*. I. K. International Pvt Ltd.

Léon, J. (2014). Early Machine Translation. A. Beckmann, E. Csuhaj-Varjú ve K. Meer (Ed.), *Language, Life, Limits* içinde , Lecture Notes in Computer Science (ss. 275-282). Cham: Springer International Publishing. doi:10.1007/978-3-319-08019-2_28

Li, P., Sun, M. ve Xue, P. (2010). Fast-Champollion: A fast and robust sentence alignment algorithm. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* içinde , COLING ’10 (ss. 710-718). USA: Association for Computational Linguistics.

Liddy, E. (2001). Natural Language Processing. *School of Information Studies—Faculty Scholarship*. 20 Haziran 2023'te https://surface.syr.edu/istpub/63 adresinden erişildi.

Luo, S., Ying, H., Li, J. ve Yu, S. (2022, 7 Şubat). Sentence Alignment with Parallel Documents Facilitates Biomedical Machine Translation. arXiv. doi:10.48550/arXiv.2104.08588

Ma, X. (2006). Champollion: A Robust Parallel Text Sentence Aligner. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)* içinde . LREC 2006, sunulmuş bildiri, Genoa, Italy: European Language Resources Association (ELRA). 20 Haziran 2023'te http://www.lrec-conf.org/proceedings/lrec2006/pdf/746_pdf.pdf adresinden erişildi.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, … Xiaoqiang Zheng. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 05 Haziran 2023'te https://www.tensorflow.org/ adresinden erişildi.

Melamed, I. D. (1999). Bitext maps and alignment via pattern recognition. *Computational Linguistics*, *25*(1), 107-130.

Moore, R. C. (2002a). Fast and Accurate Sentence Alignment of Bilingual Corpora. *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users* içinde , AMTA '02 (ss. 135-144). Berlin, Heidelberg: Springer-Verlag.

Moore, R. C. (2002b). Fast and Accurate Sentence Alignment of Bilingual Corpora. *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users* içinde , AMTA '02 (ss. 135-144). Berlin, Heidelberg: Springer-Verlag.

Morey, S., Post, M. W. ve Friedman, V. A. (2013, 1 Ocak). The language codes of ISO 639: A premature, ultimately unobtainable, and possibly damaging

standardization. La Trobe University, University of New England, University of Chicago. 20 Haziran 2023'te http://hdl.handle.net/2123/9838 adresinden erişildi.

Ngo Ho, A. K. (2021, Şubat). *Generative Probabilistic Alignment Models for Words and Subwords: A Systematic Exploration of the Limits and Potentials of Neural Parametrizations*. (Theses). 17 Haziran 2023'te https://theses.hal.science/tel-03210116 adresinden erişildi.

Oliphant, T. E. (2015). *Guide to NumPy* (2nd bs.). North Charleston, SC, USA: CreateSpace Independent Publishing Platform.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox ve R. Garnett (Ed.), *Advances in Neural Information Processing Systems 32* içinde (ss. 8024-8035). Curran Associates, Inc. 10 Haziran 2023'te http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf adresinden erişildi.

Pozzi, L. (2022). *Keyed Alike: Towards Versatile Domain-Specific Keyword Extraction with BERT*. (Master Thesis). 10 Haziran 2023'te https://studenttheses.uu.nl/handle/20.500.12932/43073 adresinden erişildi.

Pretrained Models—Sentence-Transformers documentation. (2022, 7 Eylül). 23 Mayıs 2023 tarihinde https://www.sbert.net/docs/pretrained_models.html adresinden erişildi.

Qi, Y., Sachan, D., Felix, M., Padmanabhan, S. ve Neubig, G. (2018). When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation? *Proceedings of the 2018 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers) içinde (ss. 529-535). New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-2084

Reimers, N. ve Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *CoRR, abs/1908.10084*. 02 Haziran 2023'te http://arxiv.org/abs/1908.10084 adresinden erişildi.

Reimers, N. ve Gurevych, I. (2020). Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. *arXiv preprint arXiv:2004.09813*. 10 Haziran 2023'te http://arxiv.org/abs/2004.09813 adresinden erişildi.

Salton, G. ve Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513-523. doi:https://doi.org/10.1016/0306-4573(88)90021-0

Sapir, E. (1921). *Language, An Introduction to the Study of Speech*. *Language*. New York, NY: New York: Brace.

Sari, F. (2021). Cahit Arf'ın "Makine Düşünebilir mi ve Nasıl Düşünebilir?" Adlı Makalesi Üzerine Bir Çalışma. *TRT Akademi*. doi:10.37679/trta.962940

Schweter, S. (2020, Nisan). BERTurk—BERT models for Turkish. Zenodo. doi:10.5281/zenodo.3770924

Sennrich, R. ve Volk, M. (2010). MT-based Sentence Alignment for OCR-generated Parallel Texts. *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas: Research Papers* içinde . AMTA 2010, sunulmuş bildiri, Denver, Colorado, USA: Association for Machine Translation in the Americas. 05 Haziran 2023'te https://aclanthology.org/2010.amta-papers.14 adresinden erişildi.

Staykova, K. (2014). Natural Language Generation and Semantic Technologies. *Cybernetics and Information Technologies, 14*(2), 3-23. doi:doi:10.2478/cait-2014-0015

Talan, T. ve Aktürk, C. (2022). *Bilgisayar Bilimlerinde Teorik Ve Uygulamalı Araştırmalar*. İstanbul: Efe Akademi Yayınları.

Thompson, B. ve Koehn, P. (2019). Vecalign: Improved Sentence Alignment in Linear Time and Space. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* içinde (ss. 1342-1348). EMNLP-IJCNLP 2019, sunulmuş bildiri, Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1136

Tüysüz, M. A. A. (2003). *Doğal Dillerin Anlambilimsel (Semantik) İncelemesi (Yayın no: 136453)* [Yüksek Lisans Tezi, Trakya Üniversitesi]. http://dspace.trakya.edu.tr/xmlui/handle/trakya/379 adresinden erişildi.

Tüysüz, M. A. A. ve Güvenoğlu, E. (2014). Türkçe için Karşılaştırmalı bir Kelime Anlamı Belirginleştirme Uygulaması. *Akademik Bilişim*, *14*.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … van Mulbregt, P. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature methods, 17*(3), 261-272. doi:10.1038/s41592-019-0686-2

Wu, D. (1994). Aligning a parallel English-Chinese corpus statistically with lexical criteria. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* içinde , ACL '94 (ss. 80-87). USA: Association for Computational Linguistics. doi:10.3115/981732.981744

Yıldırım, D. ve Alkış, Ü. (2016). Süleyman Demirel Üniversitesi Sağlık Bilimleri Dergisi , Temporomandibular Eklem Bozukluklarının Değerlendirilmesinde Kullanılan Görüntüleme Yöntemleri Imaging Methods Used In The Evaluation Of Temporomandibular Joint Disorders, 7(2), 51-57.

Тульчак, Л. В., Марчук, А. О., Tulchak, L. V. ve Marchuk, A. O. (2016). History of Python. 05 Haziran 2023'te https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10471/461.pdf?sequence=3 adresinden erişildi.

ÖZGEÇMİŞ

Emir KIZILIRMAK

Eğitim

Derece	Yıl	Üniversite, Enstitü, Anabilim Dalı
Yüksek Lisans	2022	Ahmet Yesevi Üniversitesi, Lisansüstü Enstitüsü Siber Güvenlik Tezsiz Yüksek Lisans Programı
Yüksek Lisans	2020	Maltepe Üniversitesi, Lisansüstü Enstitüsü Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Lisans	2015	Maltepe Üniversitesi, Mühendislik Fakültesi Bilgisayar Mühendisliği (İngilizce)

İş/İstihdam (Varsa)

Yıl	Görev
2021 -	Kıdemli Yazılım Yöneticisi - Burgan Bank
2021 - 2021	Kıdemli Yazılım Uzmanı - Pudo
2018 - 2021	Kıdemli Yazılım Uzmanı & DBA - Fiba Perakende
2017 - 2018	Yazılım Uzmanı - Ceva Lojistik
2016 - 2017	Yazılım Danışmanı – Intellica