

**İNGİLİZCE–TÜRKÇE ÇEVİRİ METİNLERDE LEVENSHTTEIN  
UZAKLIĞI İLE DESTEKLENMİŞ ÇAPA TABANLI CÜMLE  
EŞLEME**

Emir Kızılırmak  
171402209

**YÜKSEK LİSANS TEZİ**

Bilgisayar Mühendisliği Anabilim Dalı  
Bilgisayar Mühendisliği Yüksek Lisans Programı  
Danışman: Dr. Öğr. Üyesi Mehmet Ali Aksoy Tüysüz

İstanbul  
T.C. Maltepe Üniversitesi  
Lisansüstü Eğitim Enstitüsü  
Ocak, 2020

**İNGİLİZCE–TÜRKÇE ÇEVİRİ METİNLERDE LEVENSHTAIN  
UZAKLIĞI İLE DESTEKLENMİŞ ÇAPA TABANLI CÜMLE  
EŞLEME**

Emir Kızılırmak

17 14 02 209

Orcid: 0000-0001-8518-5152

**YÜKSEK LİSANS TEZİ**

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Yüksek Lisans Programı

Danışman: Dr. Öğr. Üyesi Mehmet Ali Aksoy Tüysüz

İstanbul

T.C. Maltepe Üniversitesi

Lisansüstü Eğitim Enstitüsü

Ocak, 2020



## JÜRİ VE ENSTİTÜ ONAYI

EMİR KIZILIRMAK'ın "İngilizce-Türkçe Metinlerde Levenshtein Uzaklığı İle Desteklenmiş Çapa Tabanlı Cümle Eşleme" başlıklı tezi 24.01.2020 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Maltepe Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği" nin ilgili maddeleri uyarınca Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans/~~Doktora~~ tezi oy birliğiyle/~~oy çokluğuyla~~, başarılı/~~başarısız~~ olarak kabul edilmiştir.

	Unvanı, Adı ve Soyadı	İmza
Üye (Tez Danışmanı)	Dr. Öğr. Üyesi Mehmet Ali Aksoy TÜYSÜZ	
Üye	Dr. Öğr. Üyesi Selim BAYRAKLI	
Üye	Dr. Öğr. Üyesi Volkan TUNALI	



Prof. Dr. Belma AKŞİT  
Enstitü Müdürü V.



**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ  
ETİK İLKE VE KURALLARA  
UYUM BEYANI**

Doküman No	FR-178
İlk Yayın Tarihi	01.03.2018
Revizyon Tarihi	23.01.2020
Revizyon No	01
Sayfa	1

**05/02/2020**

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bulguların sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilmeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; çalışmamın Maltepe Üniversitesinde kullanılan “bilimsel intihal tespit programı” ile tarandığını ve öngörülen standartları karşıladığını beyan ederim.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

Emir KIZILIRMAK

## TEŐEKKÜR

Bu tez alıřmamı gerekleřtirmemde, Bilgisayar Mühendisliđi lisans eđitimim ve Tezli Bilgisayar Mühendisliđi yüksek lisans eđitimim süresi boyunca kattıđı birbirinden deđerli bilgilerin yanısıra 1 yıl boyunca yardımlarımı esirgemeyen, paylařtıđı her bilginin hayatıma ayrı bir anlam kattıđı saygıdeđer deđerli danıřman hocam; Dr. Öğr. Üyesi Mehmet Ali Aksoy Tüysüz'e tez alıřmam boyunca bir an bile olsun desteđini esirgemeyen aileme ve Hande Beliz Dađhan'a sonsuz teőekkürlerimi sunarım.

Emir Kızılırmak

Ocak 2020

## ÖZ

# İNGİLİZCE–TÜRKÇE ÇEVİRİ METİNLERDE LEVENSHTAIN UZAKLIĞI İLE DESTEKLENMİŞ ÇAPA TABANLI CÜMLE EŞLEME

Emir Kızılırmak  
Yüksek Lisans Tezi  
Bilgisayar Mühendisliği Anabilim Dalı  
Bilgisayar Mühendisliği Yüksek Lisans Programı  
Danışman: Dr. Öğr. Üyesi Mehmet Ali Aksoy Tüysüz  
Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü, 2020

Doğal dil işleme, yapay zekânın ve dil biliminin gelişimiyle son yıllarda önem kazanmıştır. Çeviri şirketlerinin yapmış olduğu, “Türkçe’den İngilizce’ye - İngilizce ’den Türkçe’ye” birebir çevirileri cümlelerin sırasının değişmesi veya rastgele sıralanması sonucunda, cümleleri tekrar düzenlemek için insan gücüne ihtiyaç duyulmaktadır. Doğal dil işleme alanında, cümle hizalama algoritmaları ile ilgili çok sayıda algoritma vardır. Bu tez aşamasında sözlüğe ve cümlelerin kelime konum hizalama tekniklerine dayanarak yeni ve etkili bir yöntem önerdik. Bu yöntem sayesinde insan gücünü azaltmak, cümleleri hızlı ve doğru biçimde hizalamak mümkün olacaktır. Bu tez çalışması cümle hizalama işlemini, sözlüğe dayalı ve Levenshtein Mesafe algoritmasını kullanarak gerçekleştirmektedir. Yapılan algoritma sonrasında bilimsel, teknoloji gibi metinlerde sözlükte bulunmayan ve metin içerisinde kendini tekrarlamayan kelimelerin sayısı arttıkça cümle hizalama başarı oranının arttığı sözlükte bulunan ve kendini tekrarlayan kelime sayısının kitap, roman, hikâye gibi metinlerde ise cümle hizalama başarı oranının düştüğü gözlemlenmiştir.

**Anahtar Sözcükler:** Doğal Dil İşleme; Cümle Eşleme; Çapa

## ABSTRACT

# A LEVENSHTEIN DISTINCE AUGMENTED ANCHOR-BASED SENTENCE ALIGNMENT FOR ENGLISH-TURKISH TRANSLATED TEXT

Emir Kızılırmak

Master Thesis

Department of Computer Engineering

Computer Engineering Programme

Advisor: Asst. Prof. Mehmet Ali Aksoy Tüysüz

Maltepe University Graduate School, 2020

The natural language process has gained importance in recent years with the development of artificial intelligence and linguistics. As a result of the changing or random ordering of sentences made by the translation companies, from Turkish to English & English to Turkish” manpower is needed to rearrange the sentences. In the field of natural language processing, there are many algorithms related to sentence alignment algorithms. In this thesis, we propose a new and effective method based on the vocabulary and word position alignment techniques of sentences. With this method, it will be possible to reduce the manpower and to align sentences quickly and accurately. This thesis is based on dictionary and “Levenshtein Distance” algorithm. After the algorithm, it is observed that sentence alignment success rate decreases with the number of words that are not found in the dictionary such as scientific and technology and the number of words that do not repeat itself in the text increases, and the number of words that repeat itself in books, novels and stories decreases.

**Keywords:** Natural Language Process; Sentence Allignment; Anchor

# İÇİNDEKİLER

JÜRİ VE ENSTİTÜ ONAYI.....	ii
ETİK İLKE VE KURALLARA UYUM BEYANI.....	iii
TEŞEKKÜR.....	iv
ÖZ.....	v
ABSTRACT.....	vi
İÇİNDEKİLER.....	vii
TABLolar LİSTESİ.....	viii
ŞEKİLLER LİSTESİ.....	ix
KISALTMALAR.....	x
ÖZGEÇMİŞ.....	xi
BÖLÜM 1. GİRİŞ.....	1
1.1 Doğal Dil İşleme (Natural Language Processing).....	1
1.2 Doğal Dil İşleme Kullanım Alanları.....	2
1.3 Doğal Dil İşleme Paketleri [3].....	3
1.4 Cümle Eşleme.....	4
1.5 Bilgisayar Destekli Çeviri.....	5
1.6 Literatür Araştırması.....	6
BÖLÜM 2. YÖNTEM.....	8
2.1 Veriler.....	8
2.2 Uygulanan Yöntem.....	8
BÖLÜM 3. BULGULAR VE YORUMLAR.....	14
3.1 C# Programlama Dili Hakkında.....	14
3.1.1 Web Uygulamaları (ASP.NET).....	15
3.1.2 Konsol Uygulamaları.....	16
3.1.3 Mobil Uygulamalar.....	17
3.1.4 Web Servis Uygulamaları.....	18
3.1.5 DLL Uygulamalar.....	19
3.1.6 Windows Form Uygulamaları.....	19
3.1.7 Oyun Tasarlama.....	19
BÖLÜM 4. SONUÇ.....	20
4.1 Gelecekte Yapılabilecek Çalışmalar.....	22
KAYNAKÇA.....	23



## TABLolar LİSTESİ

Tablo 1 - Hukuk içerikli metinlerdeki cümle ve kelime sayıları .....	21
Tablo 2 - Hukuk içerikli metinlerdeki cümle eşleme işleminden sonra ki cümle sayısı	21
Tablo 3 - Teknoloji içerikli metinlerdeki cümle ve kelime sayıları .....	21
Tablo 4 - Teknoloji içerikli metinlerdeki cümle eşleme işleminden sonra ki cümle sayısı .....	21



## ŞEKİLLER LİSTESİ

Şekil 1 - İlk İngilizce cümle.....	8
Şekil 2 - İngilizce cümlenin kelimelere ayrıştırılması .....	9
Şekil 3 - Sözlükte bulunmayan kelimeler .....	10
Şekil 4 - Zemberek kütüphanesinin yüklenmesi.....	11
Şekil 5 - Sözlükte bulunmayan kelimelerin geçtiği Türkçe cümleler.....	12
Şekil 6 - Türkçe cümlelerin skorlanması .....	12
Şekil 7 - Levenstein örneği .....	13
Şekil 8 - Dünyada C# kullanımı .....	14
Şekil 9 - Web Uygulamaları (ASP.NET).....	16
Şekil 10 - Konsol Uygulamaları .....	17
Şekil 11 - Mobil Uygulamalar .....	18
Şekil 12 - Web Servis Uygulamaları .....	18
Şekil 13 - Windows Form Uygulamaları .....	19

## KISALTMALAR

<b>AI</b>	: Yapay Zeka (AI)
<b>CAT</b>	: Bilgisayar Destekli Çeviri ( Computer Assisted Translation)
<b>CMD</b>	: Komut Satırı (Command)
<b>C#</b>	: C Sharp
<b>IEEE</b>	: Elektrik ve Elektronik Mühendisleri Enstitüsü (The Institute of Electrical and Electronics Engineer)
<b>NLP</b>	: Doğal Dil İşleme (Natural Language Processing)
<b>SOAP</b>	: Simple Object Access Protocol – Basit Nesne Erişim Protokol

# ÖZGEÇMİŞ

Emir Kızıllırmak

Bilgisayar Mühendisliği Anabilim Dalı

## Eğitim

<i>Derece Yıl</i>	<i>Okul/Bölüm</i>
Y.Ls.	2020 Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Ls.	2015 Maltepe Üniversitesi, Mühendislik Fakültesi Bilgisayar Mühendisliği (İngilizce)
Lise	2005 İstek Özel Uluğbey Lisesi

## İş/İstihdam

<i>Yıl</i>	<i>Görev</i>
2018 -	Fiba Perakende Grubu – Yazılım Uzmanı ve DBA
2017- 08	Ceva Lojistik – Yazılım Uzmanı
2016- 07	Intellica Yazılım Danışmanlığı – Yazılım Danışmanı

## Kişisel Bilgiler

Doğum yeri ve yılı : İstanbul, 1988 Cinsiyet: Erkek  
Yabancı diller : İngilizce (çok iyi); Almanca(çok az)  
E-posta : emirkizilirmak@gmail.com

# BÖLÜM 1. GİRİŞ

## 1.1 Doğal Dil İşleme (Natural Language Processing)

Doğal dil işleme (Natural Language Processing - NLP), genel anlamıyla yapay zekâ ile dil biliminin bir araya gelmesinden oluşan Bilgisayar Bilim dalıdır. Doğal dil işleme, insanların kullandıkları dili bilgisayar ortamına aktararak işleme işlemidir. Doğal dil işleme ilk olarak 1948 yılında Londra’da bulunan Birkbeck College’da geliştirilen bir sözlük sistemiydi. Sonrasında 1949 yılında Warren Weaver tarafından İkinci Dünya Savaşı sırasında kod kırmak için kullanılmıştır. İkinci Dünya savaşı sırasında Weaver, insanların çeşitli diller konuşsa da tek bir dilde yazılan belgenin içerisindeki kodu bir kez kırıldığında, bu belgeyi başka dilde çıkarmak mümkün olacaktır. Bu fikri İkinci Dünya savaşında Amerikalılar, Almanca’yı İngilizce’ye çevirmek uğraştılar sonrasında Rusça’dan İngilizce’ye ve Fransızca’ya çevirmek için kullanılmıştır. Sovyetler Birliği, İngilizce’den veya Fransızca’dan Rusça’ya çeviriler başladı 1950 yılında Alan Turing, şuan Turing testi olarak bilinen “Makineler düşünebilir mi ?” fikriyle “Bilgi İşlem Makineleri ve Zekâ” makalesi yayınlanmıştır.

1960’lı yılların ortalarında Otomatik Dil İşleme Komitesi (ALPAC) bir rapor hazırladı ve bu raporda, “Genel bilimsel metnin makine çevirisi olmadığı ve hiçbirinin acil durumda olmadığı” sonucuna varıldı. Bu rapor sonrasında Makine çevirisi için yatırımlar durduruldu ve Doğal dil işleme alanındaki çalışmalar durduruldu.

1970’li yıllarda ise konuşma verilerini bilgisayar tarafında anlaşılabilir ve yapılandırmış olan “kavramsal ontolojiler” yazmaya başlandı ve bu alanda birçok sohbet botu yazıldı.

1980’lere kadar geliştirilmiş olan doğal dil işleme sistemleri karmaşık elle yazılmış kural setlerine dayanıyordu. Fakat 1980’li yıllarda birçok makine öğrenme algoritmalarının geliştirilmesiyle doğal dil işleme alanında yeniçağ açılmıştır. Dönüşümsel dilbilgisinin azalmasından kaynaklıydı. Doğal dil işleme ile mevcut durumlar kullanılarak benzer şekilde yeni kural setleri oluşturacak hale gelmiştir.

1990'lı yıllarda, IBM araştırma firmasının çalışmaları sonucunda makine çevirisi meydana geldi. Öncelikle makine çevirisini, Kanada parlamentosu, Avrupa birliği ve çeviri şirketleri yararlanmaya başladı. Makine çevirisinin başarısı kullanılan derleme bağıydı.

2000'li yıllardan günümüze doğal dil işleme çok fazla denetimsiz ve yarı denetimli öğrenme algoritmaları geliştirilmiştir. 2010'lu yıllarda temsili öğrenme ve derin sinir ağı tarzı makine öğrenme sistemleri, doğal dil işleme için çok daha gelişmiş ve başarılı sonuçlar elde etmeye yarar sağlamıştır.

Bilgi teknolojilerinin gelişmesiyle doğal dil işleme daha pratik ve kolay hale gelmiştir. İnsan dilindeki belirsizlik konusunda doğal dil işleme yetersiz kalmaktadır. Örneğin, borsa haberlerinin doğal dil işleme ile analiz edilip sonrasındaki borsa piyasasının iyi veya kötü gideceği tahmini yapılamamaktadır. [1]. [2].

## 1.2 Doğal Dil İşleme Kullanım Alanları

- Soru Cevaplama (Question Answering, OA), Bilgisayara sorulan bir soruya cevap verme yazılımıdır. Örneğin, bazı web sitelerinde bulunan sohbet botlarında kullanılan yöntemlerdir.
- Bilginin Çıkartılması (Information Extraction, IE), Bir metni analiz ederek içerisinden bilgi çıkartması. Örneğin, bir konferans davetiyesi metni içerisinden konferansın ne zaman ve nerede gerçekleşeceği gibi bilgilerin analiz edilmesidir.
- Duygu Analizi (Sentement Analysis, SA), Bir cümlenin olumlu veya olumsuz olduğunun analiz edilmesidir. Örneğin, A markası arabanın fiyatı çok uygun (olumlu) olduğunu fakat frenlerinin kötü (olumsuz) olduğunun analiz edilmesidir.
- Makine Diline Çevirisi (Machine Translation, MT), Bir dilden diğer dile doğal dil işleme işlemiyle çevrilmesidir. Bunun en bilinen örneği ise Google Translate 'tir.
- Sözcük Anlamı Açıklaştırma (Word Sense Disambiguation, WSD), Bir kelimenin sözlük anlamı olduğunu yoksa sözlükte bulunmayan bir kelime olmasının analizidir. Örneğin, Tatil turizminin en popüler yerlerinden birisi Bodrum'dur. Burada ki "Bodrum" kelimesi, evin bodrumu mu yoksa Türkiye'de bulunan" Bodrum " ilçesi mi olduğunun analiz edilmesidir.

- Özetleme (Summarization), Metin veya paragraf içerisinde, konu ile ilgili özet çıkarmaya fayda sağlamaktadır. Mesela, bir haber metni içerisinde, haberini özetini çıkartmak.

### 1.3 Doğal Dil İşleme Paketleri [3]

- Dil Tespiti (Language Detection), Kelimenin veya metnin hangi dilde olduğunu belirlemek için kullanılmaktadır.
- Cümle Tespiti (Sentence Detector), Metin içerisindeki bir cümleleri tespit ederek, başlangıç ve bitişini bulmaktadır.
- Varlık İsim Tanıma (Named Entity Recognition), 3 temel kategoride incelenmektedir. Bunlardan ilki Enamex, kişi, yer ve organizasyon gibi ifadeleri bulmak için kullanılır. İkinci olarak TIMEX, gün ve tarih gibi zaman ile ilgili ifadeleri bulmak için kullanılır. Sonuncusu ise NUMEX, para ve yüzde ifadeleri bulmak için kullanılır.
- Kelime Bölütleyici (Tokenizer), Cümlelerin veya Metnin kelimelere ayrıştırılma işlemi için kullanılır.
- Metin Normelleştirme (Normalization), Yazım hatalarını denetleyip kelime önerme ve hatalı metinleri düzeltmek için kullanılır.
- Doküman Sınıflandırma (Document Categorizer), Naive Bayes algoritması kullanılarak belge sınıflandırıcı model oluşturulması için kullanılır.
- Metin Parçası Etiketleme (Part of Speech Tagger), Bir cümle içerisindeki kelime türlerini belirlemek için kullanılır.
- Bölümleme (Chunker), Bir cümle içerisindeki kelime gruplarını belirlemek için kullanılır.
- Kök Bulmak (Steaming), Kelimenin kök halini bulmak için kullanılır.
- Sözlüğe Dayılı Kök Bulmak (Lemmatization), Kelimenin sözlükteki 'doğru' halini tespit etmek için kullanılır.

## 1.4 Cümle Eşleme

Cümle hizalama işlemi bir doğal dil işleme araştırma alanıdır. Tanım olarak, iki farklı dilde birebir çevirisi olan çeviri şirketleri metinleri, akademik çalışmaları, haber sitelerinde yayınlanan makaleler, e-kitaplar üzerinde gerçekleşmektedir. Günümüzde cümle hizalama işlemi için gerçekleştirmek için çeşitli yöntemler ve algoritmalar mevcuttur. Bunları üç farklı başlık altında inceleyebiliriz.

1. Uzunluk tabanlı hizalama, Metnin anlam bilimi sayılmaz, istatistiksel yöntemleri kullanırlar.
  2. Yer tabanlı hizalama, Uzunluk tabanlı hizalamaya benzemektedir. İki hizalama yönteminde de istatistiksel yöntemleri kullanmaktadır.
  3. Sözlüğe dayalı hizalama, İki derlem içerisindeki kelimelerin sözlük ile karşılaştırmasını yapmaktadır.
- Uzunluk tabanlı cümle hizalama algoritmalarına örnek olarak **Gale and Church (1993)** algoritması ve **Brown et. (1991)** algoritması bulunmaktadır. Bu algoritmaların ikisinde cümle uzunluklarını kelime sayısı ile karşılaştırma işlemini yapmaktadır. ), **wu (1994)**
  - Yer tabanlı hizalama algoritmalarına örnek olarak **Chuch (1993), Fung ve McKeown (1994)**
  - Sözlüğe dayalı hizalama algoritmalarında, **Kay and Riischeisen (1993), Chen (1993), Haruno and Yamazaki (1996)**

Cümle hizalama işlemi, bir cümlelerin bir cümleye çevrildiği (paralel derlem içerisinde) çevirilerde, cümle hizalama işleminin sonucunda 1-0, 1-1 veya 1-N ilişki olarak sonuç vermektedir. Hizalama işleminin sonucunda 1-1 ilişki bulması en iyi ihtimal anlamına gelmektedir. 1-0 ilişki bulma, cümlelerin diğer derlemde denk gelen cümleyi bulamamasıdır. 1-N ilişki ise diğer derlemde eşleşebilecek 1'den fazla cümle bulması anlamına gelmektedir.



## 1.5 Bilgisayar Destekli Çeviri

CAT (Computer Assisted Translation) veya Türkçe'deki ifade şekliyle Bilgisayar Destekli Çeviri (BDÇ) araçları, çeviri yaparken çevirmene kolaylık sağlaması amacıyla kullanılan bilgisayar yazılımlarıdır. Bilgisayar Destekli Çevirisi insan gücüne ihtiyaç duyulmadan veya çok az insan gücüyle gerçekleşen makine çevirisidir. Makine çevirisi, II. Dünya savaşından sonra 1950'lerin başına dayanmaktadır. 1954 yılında Georgetown üniversitesi ve IBM'in ortaklığıyla "Georgetown Deneyi" ile 60'dan fazla Rusça cümlelerin İngilizce'ye çevirisini içeriyordu. Bu deneye makine çevirisi araştırmalarına öncülük etmiştir. Sonrasında çok sayıda araştırma yapılmıştır.

1964 yılının Nisan ayında, John R. Pierce ile kurulan "ALPAC" (Automatic Language Processing Advisory Committee - Otomatik Dil İşleme Danışma Komitesi) 1966 yılında yayınlanmış raporunda makine çevirisinde hesaplamalı dilbilimdeki temel araştırmaya duyulan ihtiyacı anlattığı için çok önem kazanmıştır. Fakat bu rapor sayesinde ABD hükümetinin makine çevirisi ile ilgili yatırım azaltmasına sebep oldu. 1980'lerin sonunda bilgisayar gücünün artmasıyla maliyetler azaldı ve makine çevirisinin tekrar önem kazanmaya başladı. 1990 yıllarında başında ise bilgisayar destekli çevirileri önem kazanmaya başlamıştır. Günümüzde bilgisayar destekli çeviriyi, arama motorları ve Office programları kullanmaktadır. Örneğin, arama motorlarına "bugün hava çok nasıl" diye yazdığınız zaman "bugün hava nasıl" bunu mu demek istediniz diye kullanıcıya bilgi vermektedir. Bu bilginin arkasında bilgisayar destekli çeviri bulunmaktadır. Office programlarında yazım denetimi yaparken de bilgisayar destekli çeviri bulunmaktadır. Günümüzde çeviri şirketlerinin en çok kullandığı bilgisayar destekli çeviri programları,

1. SDL Trados
2. Wordfast
3. MemoQ
4. Logoport
5. SDL Studio
6. SDLX
7. Dejavu
8. Idiom

Bilgisayar destekli çevirinin tercüme sektörüne sağladığı katkılardan biri çok hızlı sonuçlara ulaşabilmek, insan gücünden daha hızlı sonuçlara ulaşmayı sağlar. Aslında bu araçlar size çeviri yapmıyor sadece kılavuz olmaktadır. Kolay terminoloji yöntemi sağlamaktadır. Terminoloji yönetiminde kendi oluşturmuş olduğunuz sözlüğü yazılımlara dâhil ettiğiniz zaman çok daha kaliteli sonuçlara sahip olmaktadır. Proje yönetimi sağlamaktadır, tercümeleri bu uygulama üzerinden paylaşabilir veya alabilirsiniz. Kalitesini kontrol edebilir çalışmalar, bilgisayar destekli çeviri programları çeviri yapmanın yanı sıra dilbilgisi kontrolünde gerçekleştirmektedirler. Html, pdf, psd, doc gibi dosya türleri üzerinde çalışma imkânı sunmaktadırlar.

## 1.6 Literatür Araştırması

Yapılan literatür taramasında cümle eşleme işlemi için,

- 1- **Compiling a Turkish-English Bilingual Corpus and Developing an Algorithm for Sentence Alignment** (Şerafettin Taşçı, A. Mustafa Güngör, Tunga Güngör) [4]

Konum tabanlı ve uzunluk temelli cümle hizalama algoritması kullanılarak ilk önce iki metin LingPipe ayırıcı kullanılarak metinler paragraf ve cümlelere bölünmüştür.

- İlk aşamada, kaynak ve hedef metinlerdeki paragraflar hizalanır.
- İkinci aşama, Her paragraf çifti için, paragraflardaki cümleler sıralanır.

Tüm paragrafla çiftleri arasında bir skor hesaplanır ve skorun belirlenen limit değerden en düşük olan minimum skor olan çift eşleşir. Daha sonrasında her iki metinde hizalananlar ve hizalanacak paragraflar iki bölüme ayrılır ve algoritma tekrar tekrar çalıştırılır. Örneğin, bir hizalama algoritma bilimsel bir metne uygulandığında başarılı sonuçlar verebilir, ancak başarısı bir roman veya felsefe metni üzerine uygulandığında çarpıcı şekilde düşebilir.

- 2- **Turkish-English Sentence Alignment** (Ahmet Mustafa Güngör, Şerafettin Taşçı) [5]

Uzunluk temelli cümle hizalama algoritmasını kullanarak paragraf hizalama işlemi yapılmaktadır.

- Paragraf arasında puanlama yapılırken,
- Tam metin uzunlukları
- Cümle uzunluğu
- Üst cümlelerin bağlam uzunlukları
- Nether bağlamında cümle uzunlukları
- Tam metin uzunluk oranı
- Üst bağlam uzunluğu oranı)
- Nether bağlamı uzunluk oranı
- Cümle uzunluğu oranı
- Ağırlık katsayısı

Yukarıdaki parametreler kullanılarak puan, hesaplanır ve eşğin altındaysa, en küçük çifti seçerek, tamamlanacak. Hizalamanın ardından paragrafların üst ve net kısımları bütün metin olarak kabul edilir ve yukarıdakiler için aynı prosedür uygulanır. Eğer hiçbiri metindeki çiftler arasındaki puanlar eşikten daha küçüktür, ardından tüm metinler hizalanır.

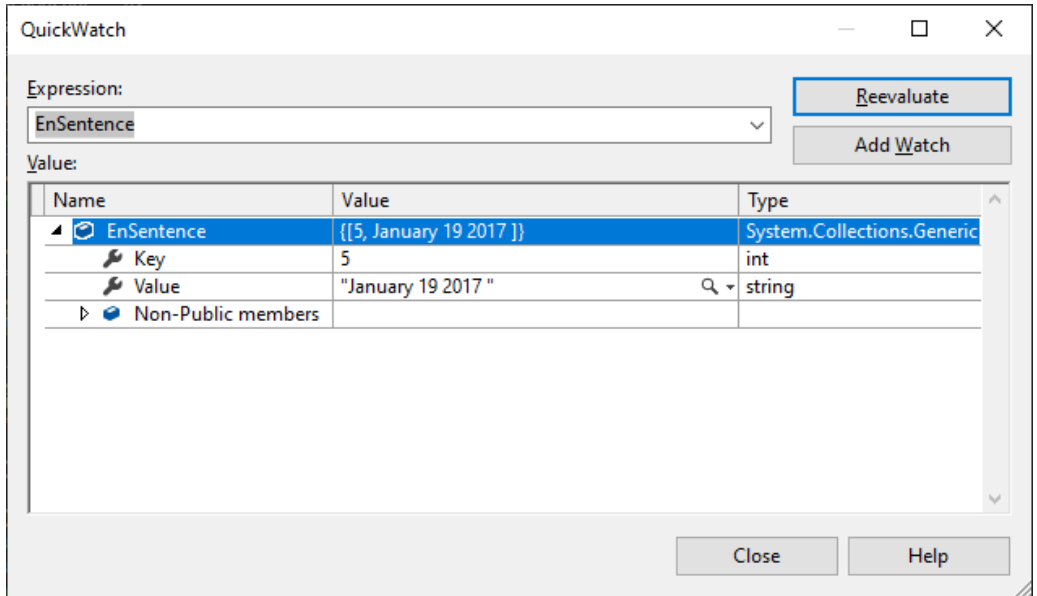
## BÖLÜM 2. YÖNTEM

### 2.1 Veriler

Bu tez çalışmamda iki farklı veri seti kullanılmıştır. İlk veri seti, çeviri firmasından birebir Türkçe – İngilizce çevrilmiş karışık içerikli metinler elde edilmiştir. Toplamda, 398 adet Türkçe – İngilizce birebir çevirisi yapılmış Hukuk ile ilgili cümleler ağırlıklı karışık metinler üzerinde tez çalışması yapılmıştır. İkinci veri seti ise teknoloji alanında toplamda 267 adet İngilizce - Türkçe birebir çevirisi yapılmış cümleler üzerinde çalışılmıştır.

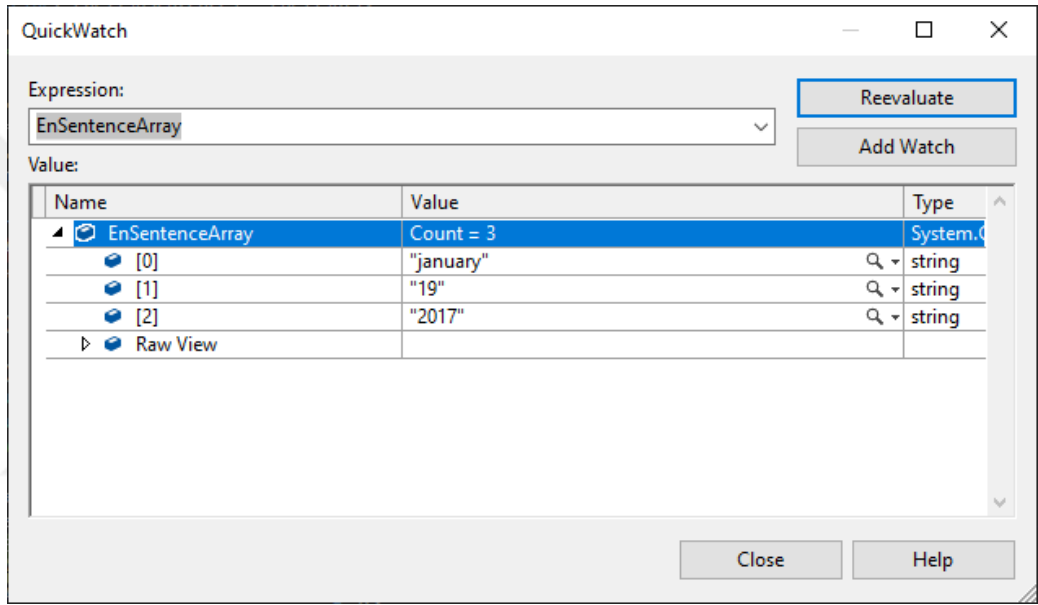
### 2.2 Uygulanan Yöntem

Bu tez çalışmamda C# programlama dili kullanılmıştır. C# kodunu derlemek için Microsoft Visual Studio programı yardımı ile derlenmiştir. İlk önce elde edilen birebir çevirisi yapılmış İngilizce - Türkçe metin veri seti ile bu tezimde cümle hizalama algoritması için geliştirdiğim yöntem, ilk önce İngilizce ve Türkçe veri seti uygulamaya kullanıcı tarafından belirtilen dizinden yüklenir. İlk yüklenmiş olan, İngilizce veri seti içerisinden ilk cümleyi alır.



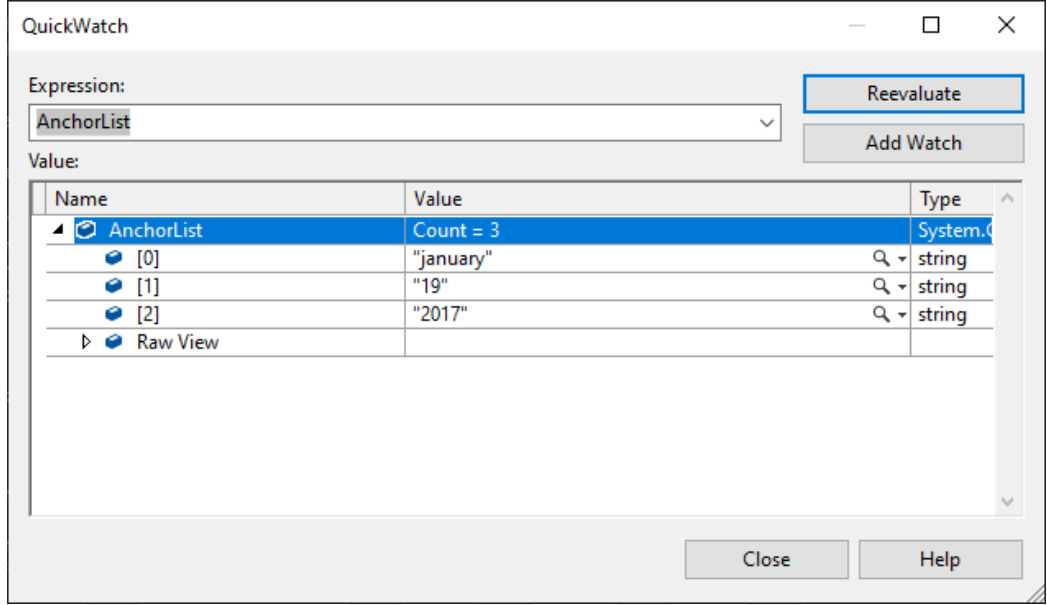
Şekil 1 - İlk İngilizce cümle

Cümleyi tokenize işlemini yaparak, kelimelerine ayırır ve kelimelerin noktalama işaretlerini ortadan kaldırmış olur ve sonrasında C# algoritma dilinde bulunan ToLower() metodu yardımıyla büyük harf olan kelimeleri küçük harflerine dönüştürür. Listede bulunan Etkisiz kelimeler (Stop words) listeden atılır. Etkisiz kelimelerin atılma amacı her yazıda geçtikleri için cümle hizalama işleminde pozitif bir katkı sağlamamaktadır. Etkisiz kelimeler atıldıktan sonra liste yeniden oluşturulur ve kelime kökü bulmak işlemi olan Lemmatization işlemi yapılır. Eğer kelimenin kökü bulunursa listedeki kelime, kökü ile yer değiştirilir. Eğer bulunamazsa kelime olduğu gibi bırakılır.



Şekil 2 - İngilizce cümlelerin kelimelere ayrıştırılması

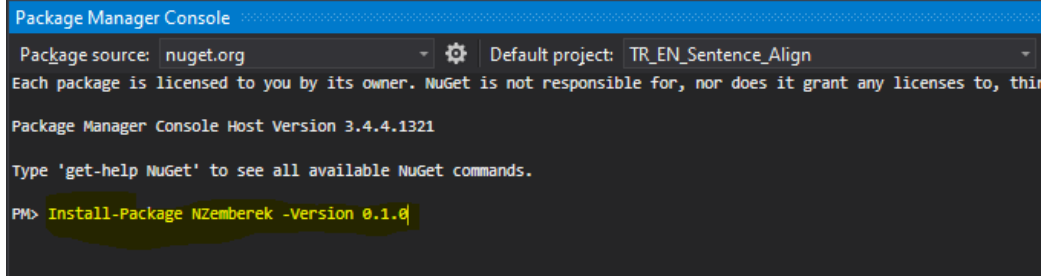
Bir sonraki aşamada daha öncesinde hazırlanmış olan İngilizce – Türkçe sözlüğün anahtar – değer listesi programa yüklenir. İngilizce kelimelerin bulunduğu liste içerisinde bulunan kelimenin sözlükte bir karşılığı olduğunu veya olmadığını kontrolünü yapmaya başlar. Kontrolü yaparken öncesinde sözlükteki kelimeyi ToLower() metodu yardımıyla büyük harf olan kelimeleri küçük harflerine dönüştürür. Eğer sözlük içerisinde karşılığını bulamaz ise bu kelime sözlükte olmayan kelime (Anchor) olarak işaretlenmektedir. İlk İngilizce cümle listesinin içerisindeki bütün kelimeleri kontrol eder ve sözlükte olmayan kelimeleri İngilizce anchor listesine ekler. Bu bölümde, araştırmanın kimlik bilgilerinden olan problem, amaç, önem, varsayımlar, sınırlıklar ve tanımlara yer verilmiştir.



Şekil 3 - Sözlükte bulunmayan kelimeler

Türkçe metin veri seti içerisindeki içerisinden ilk cümleyi alır ve o cümleyi tokenize işlemini yaparak, kelimelerine ayırarak. Kelimelerin noktalama işaretlerini ortadan kaldırmış olur ve sonrasında C# algoritma dilinde bulunan ToLower() metodu yardımıyla büyük harf olan kelimeleri küçük harflerine dönüştürür. Listede bulunan Etkisiz kelimeler (Stop words) listeden atılır. Sonrasında liste yeniden oluşturulur ve kelime kökü bulmak işlemi olan Lemmatization işlemi Zemberek kütüphanesi yardımıyla yapılır. Zemberek kütüphanesi Zemberek kütüphanesinin Microsoft Visual Studio'ya eklenmesi için Visual Stüdyo'nun üst menüsünde bulunan Araçlar (Tools) sekmesinden "NUGET Paket Yöneticisi (NUGET Package Manager)" bölümüne girilir. Paket Yöneticisi (Package Manager) bölümüne basılır , sonrasında açılan, paket Yöneticisi Konsolu (Package Manager Console) alanına "Install-Package

NZemberek -Version 0.1.0” komutu çalıştırılır ve Zemberek kütüphane projemize otomatik olarak dahil edilir.

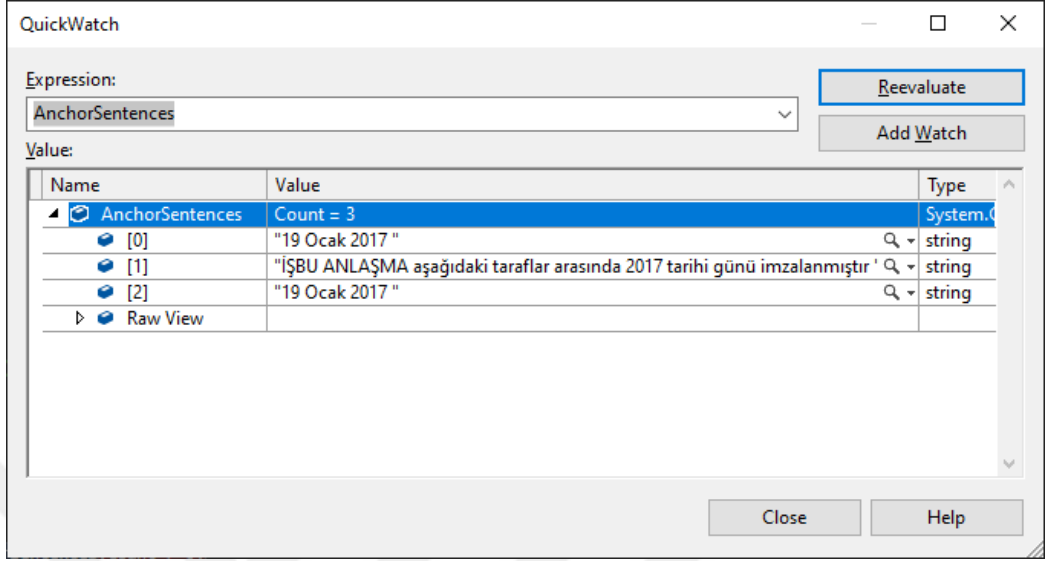


```
Package Manager Console
Package source: nuget.org
Default project: TR_EN_Sentence_Align
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, this package.
Package Manager Console Host Version 3.4.4.1321
Type 'get-help NuGet' to see all available NuGet commands.
PM> Install-Package NZemberek -Version 0.1.0
```

Şekil 4 - Zemberek kütüphanesinin yüklenmesi

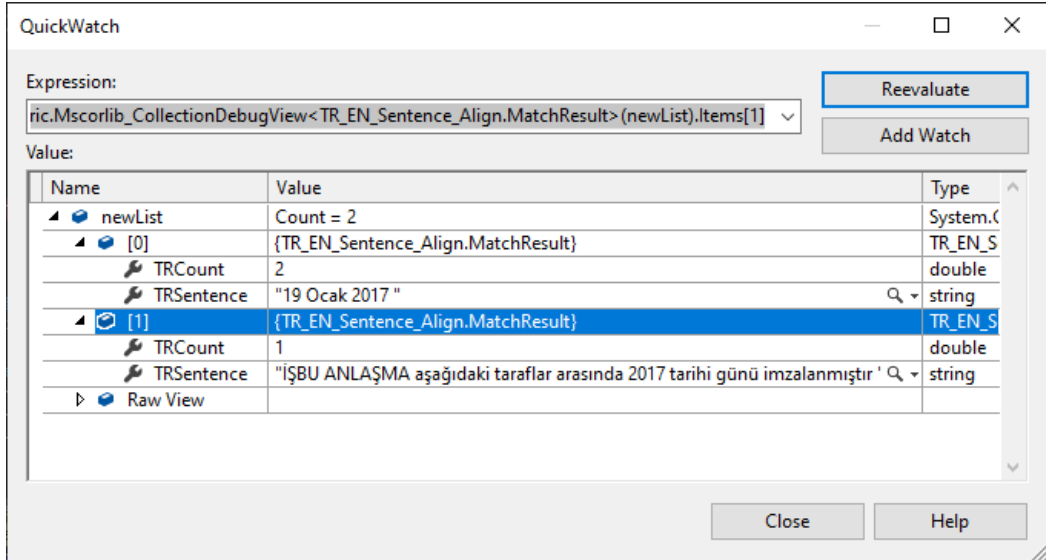
Zemberek kütüphanesi açık kaynak kodlu Türkçe Doğal Dil işleme kütüphanesidir. Java, Python ve C# desteği mevcuttur. Zemberek kütüphanesi içerisinde, Morfolojik analiz, anlam ayrımı, kelime üretimi, kelime kökü, kelime sınıflandırması, cümle sınıflandırması, yazım denetleyicisi, kelime önerme, metin düzenleme gibi paketleri mevcuttur. Eğer kelimenin kökü bulunursa listedeki kelime, kökü ile yer değiştirilir. Eğer bulunamazsa kelime olduğu gibi bırakılır. Sözlükte bulunmayan İngilizce kelime listesi içerisinde ilk elemanı alır ve Türkçe ilk cümlenin bulunduğu kelime listesi içerisinde aratılır ve eğer eşleşirse Türkçe cümleyi yeni bir liste oluşturup ekler ve sonrasında Türkçe metinler içerisinde bu işlemleri tekrarlamaktadır. Sonrasında diğer İngilizce sözlükte bulunmayan

İngilizce kelimeler ile Türkçe metinler içerisinde aramaya devam eder. Bulduklarını yeni listeye ekler.



Şekil 5 - Sözlükte bulunmayan kelimelerin geçtiği Türkçe cümleler

Sonrasında Türkçe cümlelerin bulunduğu yeni liste içerisinde cümleleri gruplar ve bir skorlama yapıp en fazla olan cümleyi alır.



Şekil 6 - Türkçe cümlelerin skorlanması

Bu sayede 1-1 karşılığını bulmuş olur. Eğer birden fazla en fazla cümle bulursa, 1-N ilişki bulmuş olur. Sonrasında N listesi içerisinde en fazla yakın olanını



bulmak için Levenshtein, mesafe algoritması kullanılmaktadır. Levenshtein mesafe algoritması 1965 yılında Vladimir Levenshtein 'den adını almıştır. Levenshtein mesafe algoritması aynı zamanda düzenleme mesafesi olarakta adlandırılır. Levenshtein mesafe algoritması temel olarak iki kelimenin birbirine olan benzerliklerini hesaplamak amacıyla kullanılır. Arama motorlarının ve Office programlarının yazım denetimi yaparken kullandığı algoritmadır.

		m	e	i	l	e	n	s	t	e	i	n
l	0	1	2	3	4	5	6	7	8	9	10	11
e	1	1	2	3	3	4	5	6	7	8	9	10
v	2	2	1	2	3	3	4	5	6	7	8	9
e	3	3	2	2	3	4	4	5	6	7	8	9
n	4	4	3	3	3	3	4	5	6	6	7	8
s	5	5	4	4	4	4	3	4	5	6	7	7
h	6	6	5	5	5	5	4	3	4	5	6	7
t	7	7	6	6	6	6	5	4	4	5	6	7
e	8	8	7	7	7	7	6	5	4	5	6	7
i	9	9	8	8	8	7	7	6	5	4	5	6
i	10	10	9	8	9	8	8	7	6	5	4	5
n	11	11	10	9	9	9	8	8	7	6	5	4

l	e	v		e	n	s	h	t	e	i	n
o	=	o	+	=	=	=	-	=	=	=	=
m	e	i	l	e	n	s		t	e	i	n

l	e		v	e	n	s	h	t	e	i	n
o	=	+	o	=	=	=	-	=	=	=	=
m	e	i	l	e	n	s		t	e	i	n

Şekil 7 - Levenstein örneği



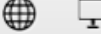







Levenshtein mesafe algoritması ile en benzer olan cümleler bulur ve 1-1 ilişki kurulmuş olmaktadır. Eğer İngilizce cümlede hiç anchor kelime yoksa veya Türkçe cümlelerin içerisinde hiç geçmiyorsa, Levenshtein algoritması yardımıyla en benzer cümle bulunmaktadır. Bu işlem İngilizce cümleler bitene kadar devam etmektedir. Sonrasında bulunan İngilizce – Türkçe cümleler kullanıcıdan alınan dizin içerisinde bir metin dosyasını yazdırılmaktadır.

## BÖLÜM 3. BULGULAR VE YORUMLAR

Bu bölümde, cümle eşleme işlemi yapılması için gerekli yazılımların hazırlanması için gerekli bilgileri verilmektedir.

### 3.1 C# Programlama Dili Hakkında

C # (Sharp) programlama dili Microsoft firması tarafından, programlama dili olan C ve C++ 'ın birleşmesinden türetilmiştir. Anders Heljsberg ve ekibi tarafından ilk olarak 2000 yılında geliştirilmesine başlanan ve 2002 yılında C# 1.0 olarak piyasaya sürülen C# programlama dili şuan 7.0 versiyonuna kadar geliştirmiş ve geliştirilmeye devam etmektedir. 2016 yılında IEEE (Elektrik ve Elektronik Mühendisleri Enstitüsü)'nün yaptığı araştırmaya göre dünyanın en popüler 6. Programlama dili olarak belirlenmiştir.

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

Şekil 8 - Dünyada C# kullanımı

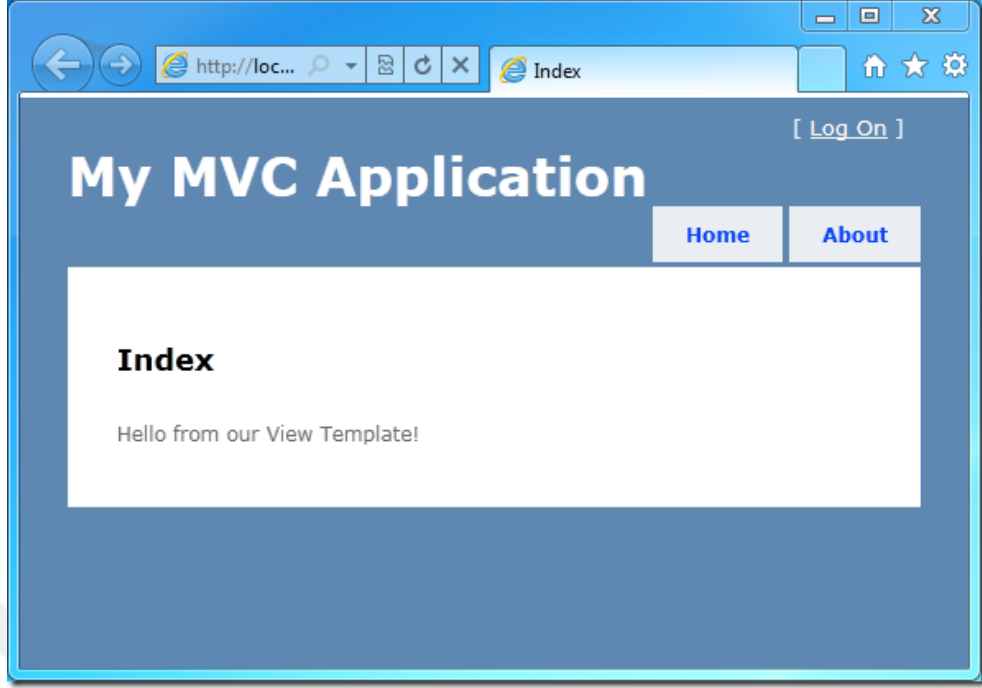
C# programlama dili ařađıdaki teknolojiler kullanılabilir.

- Web Web Uygulamaları (Asp.NET)
- Konsol uygulamaları
- Mobil uygulamalar
- Web servisleri uygulamaları
- DLL yazma
- Windows Form Uygulamaları
- Oyun tasarlama

### **3.1.1 Web Uygulamaları (ASP.NET)**

ASP.NET ile web tasarımcılar, yazılımcılar web sitesi uygulamaları geliřtirirken kullanılmaktadır. Web uygulamalarının ierisinde web formları mevcuttur. Uygulamanın grsel kısmı .aspx uzantılı dosyalarda, arka tarafta yazılan kod kısmı ise .aspx uzantılı dosyadan bađımsız bir řekilde .cs uzantılı dosyada bulunmaktadır.

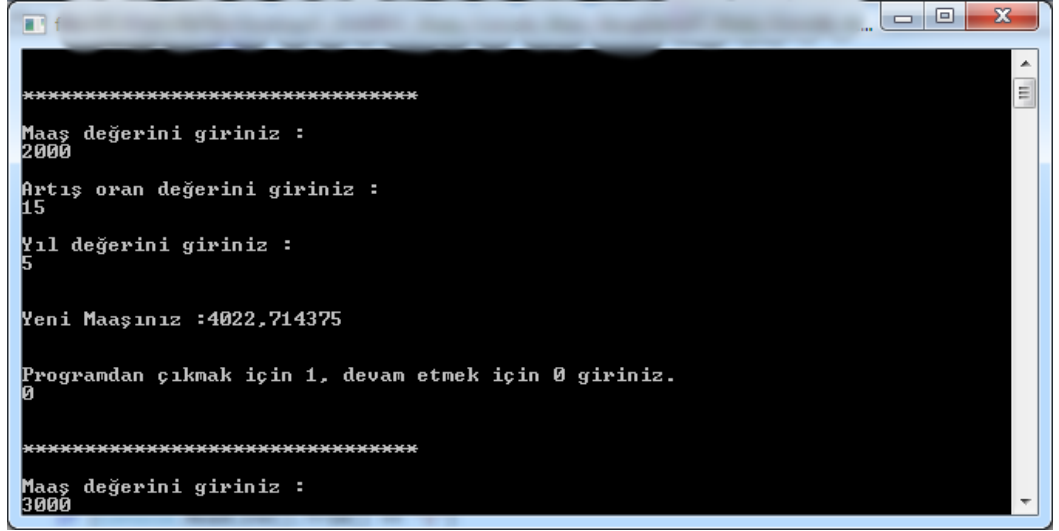
Web uygulamaları derlendikten sonra tek bir dosya olarak sunulmaktadır. Daha sonrasında Web sunucusunun IIS (İnternet Bilgi Servisleri - Internet Information Services) servisine yklenir. Yklenen web sayfasına istek gnderildiđinde, ilk olarak istek web sunucusuna gelir.



Şekil 9 - Web Uygulamaları (ASP.NET)

### 3.1.2 Konsol Uygulamaları

Grafiksel kullanıcı ara yüzünden çok, yazılımcı tarafından yazılan kodun Windows üzerinde bulunan CMD (Komut Satırı) , Kullanım amacı ise bazen uygulamaların grafik ara yüzüne ihtiyaç duyulmaz veya programlamayı öğrenirken kullanımı diğer uygulamalara göre daha hızlı bir yol olmaktadır.

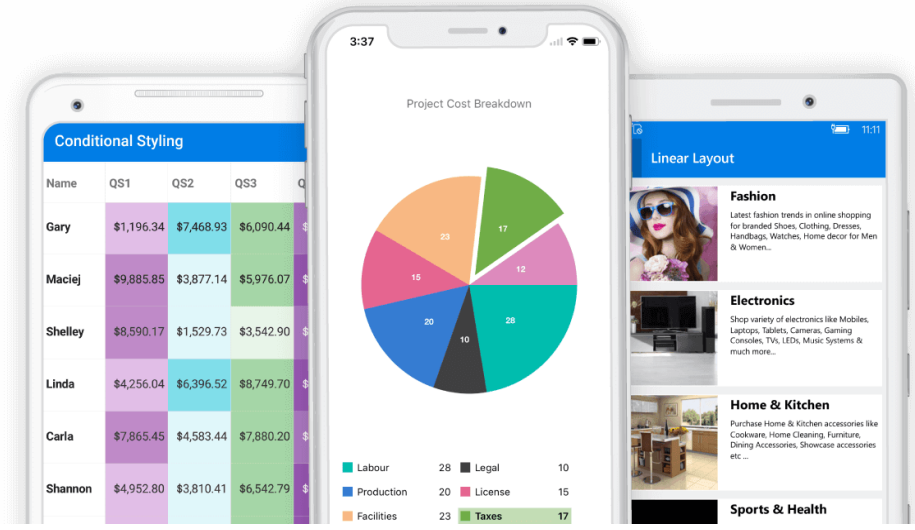


```
*****
Maaş değerini giriniz :
2000
Artış oran değerini giriniz :
15
Yıl değerini giriniz :
5
Yeni Maaşınız : 4022,714375
Programdan çıkmak için 1, devam etmek için 0 giriniz.
0
*****
Maaş değerini giriniz :
3000
```

Şekil 10 - Konsol Uygulamaları

### 3.1.3 Mobil Uygulamalar

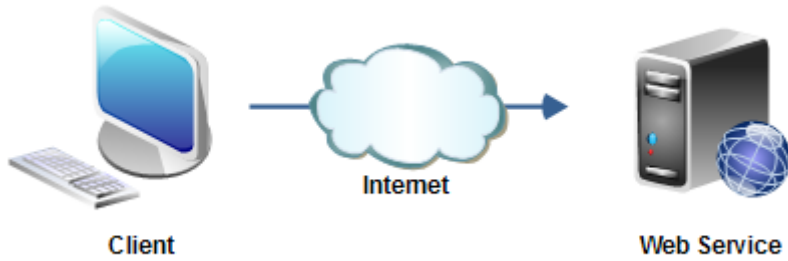
Windows Phone işletim sistemi olan telefon, tablet veya el terminalleri için uygulama geliştirme bölümüdür. Visual Studio 2015 ile birlikte Cross Platform (Çapraz Platform) desteği gelmiştir. Cross platform denildiğinde bir defa yazılan kodun farklı platformlarda çalışmasına denilmektedir. Xamarin geliştirme aracı ile Android, IOS ve Windows Phone uygulamalarını tek kodlama ile geliştirmemize olanak sağlamaktadır. Xamarin ile kod geliştirmenin en büyük avantajlarından bir tanesi Hibrit değil Native olarak uygulama geliştirme imkânı sağlamaktadır. Hibrit uygulama geliştirmenin en büyük avantajlarından bir tanesi de mesela geliştirdiğiniz uygulama ile telefonun kamerasını kullanmak istiyorsunuz. Bu noktada telefonun kamerasına direkt olarak erişebilmenize imkân sağlamaktadır. Bu sayede zaman ve maliyet açısından büyük avantajları mevcuttur.



Şekil 11 - Mobil Uygulamalar

### 3.1.4 Web Servis Uygulamaları

Web servis uygulamalarını kısacası veri iletimi için kullanılan sistemlerdir diyebiliriz Web servisler, HTTP protokolünü kullanarak veri alışverişini yapmaktadırlar. İstemci bilgisayar (client) tarafından gönderilen istekleri SOAP (Simple Object Access Protocol – Basit Nesne Erişim Protokol) XML yapısını kullanacak çalışmaktadır.



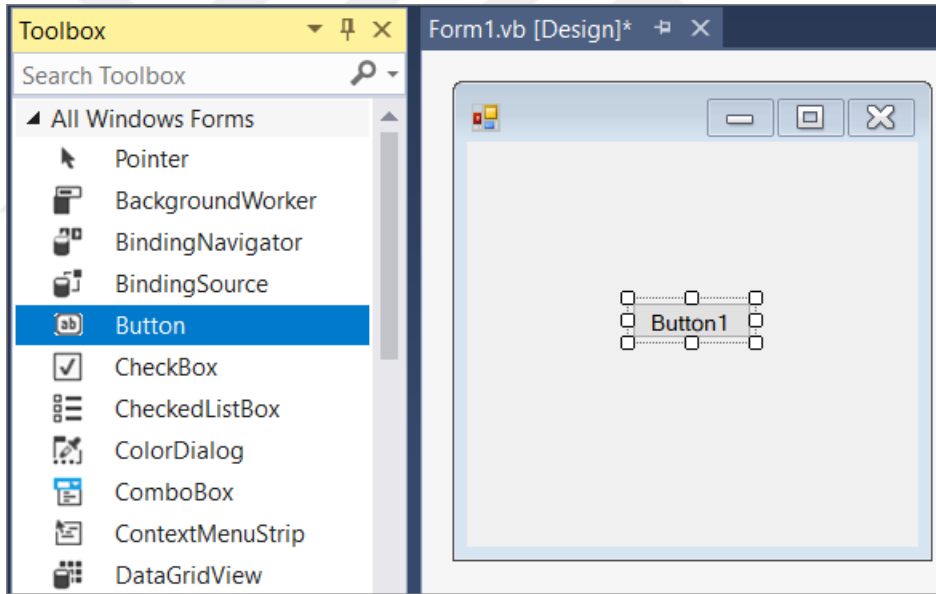
Şekil 12 - Web Servis Uygulamaları

### 3.1.5 DLL Uygulamalar

DLL, (Dynamic Link Library – Dinamik Bağlantı Kütüphaneleri), Microsoft tarafında Windows üzerinde farklı yazılımların faydalana bileneceği içerisinde makine dilinde çeşitli prosedürler içeren kitap dosyasıdır. Kullanım amacı bir den fazla programın kullanabileceği ortak bir kod havuzu olarak düşünülebilir.

### 3.1.6 Windows Form Uygulamaları

Grafiksel kullanıcı ara yüzü olarak konsol uygulamalarından ayıran en büyük özelliklerinden biridir. Windows işletim sistemi üzerinde çalışacak .exe içerisinde; Form, TextBox, CheckBox, DataGridView gibi sınıflara sahip uygulamaları içermektedir.



Şekil 13 - Windows Form Uygulamaları

### 3.1.7 Oyun Tasarlama

Unity sayesinde Windows, Android, Konsollar ve IOS için oyunlar tasarlamaya yarayan uygulamalardır. Unity, sayesinde C# ile kod yazma imkânı sağlamaktadır. C#'ın nesnel programlama sayesinde büyük ölçekli projeleri kolayca uygulama imkânı sunmaktadır.

## BÖLÜM 4. SONUÇ

Bu tez çalışmamda cümle eşleşme işlemi yapılarak, birebir çevirisi yapılmış hukuk ağırlıklı 398 adet Türkçe – İngilizce birebir çevirisi yapılmış cümle sonrasında 137 tane cümleyi doğru olarak hizalama işlemini gerçekleştirilmiştir. Toplamda 398 cümle içerisinde %34.4 gibi bir başarı elde edilmiştir. Bu 398 cümle içerisinde toplam 185 cümle içerisinde sözlükte olmayan kelime bulunmaktadır. Bu sayı toplamda 398 cümle içerisinde %46 gibi bir orana denk gelmektedir. Belirtilen 185 cümle içerisinde toplamda sözlükte olmayan kelime sayısı toplam 388'dir. Sözlükte bulunmayan kelime ile eşletirilebilen cümle sayısı 15'dir. Sözlükte bulunmayan kelime kullanılarak birden fazla eşletirilebilen cümle sayısı 58'dir. Bunlardan 38 tanesi Levenstein mesafe algoritması ile doğru eşleşmektedir. İçinde 213 tane cümle içerisinde sözlükte bulunmayan kelime yoktur. Bu cümleleri Levenstein mesafe algoritması ile cümleleri hizalamak istediğimizde, cümlelerden 84 tanesi doğru olarak cümlelerin hizalandığı gözlemlenmiştir. Bu tez çalışmamızı hukuk metinleri üzerinde gerçekleştirdik. İkinci veri seti, teknoloji alanındaki cümleleri içermektedir. 267 cümle içerisinde 147 olarak hizalama işlemini gerçekleştirilmiştir. Bu sayı toplamda 267 cümle içerisinde %55 gibi bir başarı elde edilmiştir. Bu 267 cümle içerisinde toplam 207 cümle içerisinde sözlükte olmayan kelime bulunmaktadır. Toplamda 267 cümle içerisinde %77.5 gibi bir orana denk gelmektedir. Bu 207 cümle içerisinde toplamda sözlükte olmayan kelime sayısı toplam 523'dir. Sözlükte bulunmayan kelime kullanılarak eşletirilebilen cümle sayısı 96'dir. Sözlükte bulunmayan kelime ile birden fazla eşletirilebilen cümle sayısı 76'dir bunlardan 32 tanesi Levenstein mesafe algoritması ile doğru eşleşmektedir. İçinde 60 tane cümle içerisinde sözlükte bulunmayan kelime yoktur. Bu cümleleri Levenstein mesafe algoritması ile cümleleri hizalamak istediğimizde, cümlelerden 19 tanesi doğru olarak cümlelerin hizalandığı gözlemlenmiştir. Yapılan algoritma sonrasında bilimsel, teknoloji gibi metinlerde sözlükte bulunmayan ve metin içerisinde kendini tekrarlamayan kelimelerin sayısı arttıkça cümle hizalama başarı oranının arttığı sözlükte bulunan ve kendini tekrarlayan kelime sayısının kitap, roman, hikâye gibi metinlerde ise cümle hizalama başarı oranının düştüğü gözlemlenmiştir.



Tablo 1 - Hukuk içerikli metinlerdeki cümle ve kelime sayıları

<b>Anchor Bulunan Cümle Sayısı</b>	<b>Toplam Anchor Sayısı</b>	<b>Anchor Geçmeyen Cümle Sayısı</b>	<b>Toplam Cümle Sayısı</b>
185	388	213	398

Tablo 2 - Hukuk içerikli metinlerdeki cümle eşleme işleminden sonra ki cümle sayısı

<b>Anchor Geçmeyen Levenstein İle Doğru bulunan Cümle Sayısı</b>	<b>Birden Fazla Anchor Geçen Levenstein İle Doğru bulunan Cümle Sayısı</b>	<b>Anchor İle Eşlesen Cümle Sayısı</b>	<b>Toplamda Doğru Eşleşen Cümle Sayısı</b>
84	38	15	137

Tablo 3 - Teknoloji içerikli metinlerdeki cümle ve kelime sayıları

<b>Anchor Bulunan Cümle Sayısı</b>	<b>Toplam Anchor Sayısı</b>	<b>Anchor Geçmeyen Cümle Sayısı</b>	<b>Toplam Cümle Sayısı</b>
207	523	60	267

Tablo 4 - Teknoloji içerikli metinlerdeki cümle eşleme işleminden sonra ki cümle sayısı

<b>Anchor Geçmeyen Levenstein İle Doğru bulunan Cümle Sayısı</b>	<b>Birden Fazla Anchor Geçen Levenstein İle Doğru bulunan Cümle Sayısı</b>	<b>Anchor İle Eşlesen Cümle Sayısı</b>	<b>Toplamda Doğru Eşleşen Cümle Sayısı</b>
19	32	96	147

#### 4.1 Gelecekte Yapılabilecek Çalışmalar

Bu tez çalışmamda İngilizce metin içerisinde geçen cümlede “45’t<sup>h</sup>” olan bir kelimenin Türkçe ’ye çevirisinde “45’inci” olarak çevrildiğini tespit edilmiştir. “45’t<sup>h</sup>” sözlükte olmayan kelime olduğu için, Türkçe cümleler arasında karşılığını bulamamaktadır. Bunun gibi kelimeleri yakalayıp eğer Türkçe metin içerisinde yoksa Jaccard mesafesi benzerlik algoritması ile en yakın değeri alınıp sonrasında hizalama işlemi yapılabilir.



## KAYNAKÇA

- [1] «A brief history of Natural Language Processing,» 1996. [Çevrimiçi]. Available: [https://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1\\_history.html](https://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html). [Erişildi: 15 12 2019].
- [2] «Why Natural Language Processing (NLP) is a core AI Technology,» 2018. [Çevrimiçi]. Available: <https://witanworld.com/article/2018/10/28/naturallanguageprocessing-nlp/>. [Erişildi: 11 12 2019].
- [3] T. A. S. Foundation, «Apache OpenNLP Developer Documentation,» 2011. [Çevrimiçi]. Available: <https://opennlp.apache.org/docs/1.9.1/manual/opennlp.html>. [Erişildi: 29 11 2019].
- [4] A. M. G. T. G. Şerafettin Taşçı, «Compiling a Turkish-English Bilingual Corpus and Developing an Algorithm for Sentence Alignment,» 2006. [Çevrimiçi]. Available: <https://www.cmpe.boun.edu.tr/~gungort/papers/Compiling%20a%20Turkish-English%20Bilingual%20Corpus%20and%20Developing%20an%20Algorithm%20for%20Sentence%20Alignment.pdf>. [Erişildi: 15 10 2019].
- [5] Ş. T. Ahmet Mustafa Güngör, «Turkish - English Sentence Alignment,» 2006. [Çevrimiçi]. Available: <https://www.cmpe.boun.edu.tr/~gungort/undergraduateprojects/Turkish-English%20Sentence%20Alignment.pdf>. [Erişildi: 10 09 2019].
- [6] A. L. P. A. Committee, «COMPUTERS IN TRANSLATION AND LINGUISTICS,» Washington, D. C., 1966.
- [7] E. D. Liddy, «Natural Language Processing,» New York, 2001.
- [8] W. C. K. Gale, «A program for aligning sentences in bilingual,» 1993.
- [9] J. C. L. R. L. M. Peter F. Brown, 29th Annual Meeting of the Association for Computational Linguistics, California, Berkeley: Association for Computational Linguistics, 1991, p. 169–176.
- [10] C. M. a. H. Schütze, Foundations of Statistical Natural Language Processing, Cambridge, 1999.

- [11] M. S. a. P. X. Peng Li, «Fast-Champollion: A Fast and Robust,» Beijing, 2010.
- [12] A. H. a. S. I. Omurca, «*Türkçe Metin Özetlemede Melez Modelleme,*» Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen Ve Mühendislik Dergisi, 2015.

