1 (1) (2023) 20-30
*ASSOS İnsan ve Toplum Bilimlerinde Araştırmalar Dergisi*
https://dergipark.org.tr/tr/pub/assos
*Açık Erişim*
*ASSOS Journal of Research in Humanities and Social Sciences*
https://dergipark.org.tr/en/pub/assos
Open Access

# Türkçe-Japonca LSTM Makine Çevirisi ve Kalibrasyonu

**Ali Aycan Kolukısa**[1]

[1]Japon Dili ve Edebiyatı Bölümü, İnsan ve Toplum Bilimleri Fakültesi, Çanakkale Onsekiz Mart Üniversitesi, Çanakkale, Türkiye

**Makale Bilgisi**
*Geliş Tarihi:* 12 Temmuz 2023
*Kabul Tarihi:* 3 Kasım 2023
*Yayım Tarihi:* 1 Aralık 2023
Araştırma Makalesi

**Öz** – Makine çevirisi kavramı tarihin daha eski zamanlarında ortaya çıkmış olsa da, ilk makine çevirisinin 1933 yılında Fransa'da George Artsrouni adlı bir mucit tarafından geliştirilen bir cihaz tarafından gerçekleştirildiği yaygın olarak bilinmektedir. Ancak günümüzdeki gibi modern makine çevirisinin gelişimi ise ancak bilgisayar sistemlerinin ve doğal dil işleme tekniklerinin icadından sonra sağlanabilmiştir. Modern bilgisayarların tarihi 2. Dünya Savaşı'nda Alan Turing ile başlamış ve ardından Soğuk Savaş'ın da etkisiyle ilk modern makine çevirisi 1954 yılında Georgetown Üniversitesi ve IBM firması sayesinde Rusça'dan İngilizce'ye şeklinde gerçekleşmiştir. Ancak kural tabanlı bir algoritma üzerine geliştirilen IBM 701 adlı bilgisayar tarafından gerçekleştirilen bu çevirinin çok sınırlı sayıda kelime ve dilbilgisel kurallara dayalı olarak çalışmaktaydı. Makine çevirisinin evriminde en önemli rol oynayan faktörlerden biri ise, hiç şüphesiz yapay sinir ağları olmuştur. Yapay sinir ağları 1940'lı yılların başında keşfedilmiş olsa da modern bilgisayar teknolojileri yardımıyla 21. yüzyılın başından itibaren derin öğrenme modelleri aracılığıyla çeviri alanında kullanılmaya başlanmıştır. Özellikle 2013 yılında Kalchbrenner ve Blunsom tarafından sunulan makale çok ilgi görerek yapay sinir ağlarının olanaklarından makine çevirisi alanında faydalanılma yoluna girilmiştir. Bu çalışmada yapay sinir ağlarından biri olan Uzun-Kısa Vadeli Bellek (LSTM)'ten faydalanılarak oldukça düşük boyutlu bir eğitim verisi ile Türkçe-Japonca makine çeviri uygulaması yapılmıştır. Düşük bir veri kullanımında ortaya çıkabilecek olası sorunlar belirlenmeye çalışılarak bu tür bir veri ile en verimli şekilde makine çevirisinin yapılabilmesi için gerekli kalibrasyonun ne şekilde yapılması gerektiği ele alınmıştır.

**Anahtar Kelimeler** *LSTM, makine çevirisi, Türkçe, Japonca, Python*

# Turkish-Japanese LSTM Machine Translation and Calibration

[1]Department of Japanese Language and Literature, Faculty of Humanities and Social Sciences, Çanakkale Onsekiz Mart University, Çanakkale, Türkiye

**Article Info**
*Received*: 12 Jul 2023
*Accepted*: 3 Nov 2023
*Published*: 1 Dec 2023
Research Article

**Abstract** – Although the concept of machine translation is created in older times in the history, it is widely known that the first machine translation was carried out in 1933 by a device developed by an inventor named George Artsrouni in France. However, it was different than the today's machine translation and the development of modern machine translation could be achieved only after the invention of the computer systems and natural language processing techniques. Besides, the history of the modern computers started with Alan Turing in the WWII and after that with the effects of the Cold War, the first machine translation was developed in 1954 by the IBM 701 computer from Russian to English, in cooperation with Georgetown University. It is developed on a rule-based algorithm and this machine translation is known to work with a very limited number of words and grammatical rules. The factor that played the most important role in the development of machine translation was undoubtedly artificial neural networks. Although artificial neural networks were discovered in the early 1940s, they were built with the help of modern computer technologies and used in the field of translation through deep learning models since the beginning of the 21st century. Especially in 2013, the paper presented by Kalchbrenner and Blunsom attracted a lot of attention, and then the possibilities of artificial neural networks tended to be used more actively in the field of machine translation. In this study, Long Short-Term Memory (LSTM) architecture, which is one of the types of artificial neural networks, was created by Python codes and a Turkish - Japanese machine translation application was carried out with a very low amount of training data. Thus, possible problems that may arise in the use of low amount of data are identified and how these problems can be overcome for an efficient machine translation is pointed out.

**Keywords** *LSTM, machine translation, Turkish, Japanese, Python*

[1]aliaycan.kolukisa@comu.edu.tr (Sorumlu Yazar/Corresponding Author)

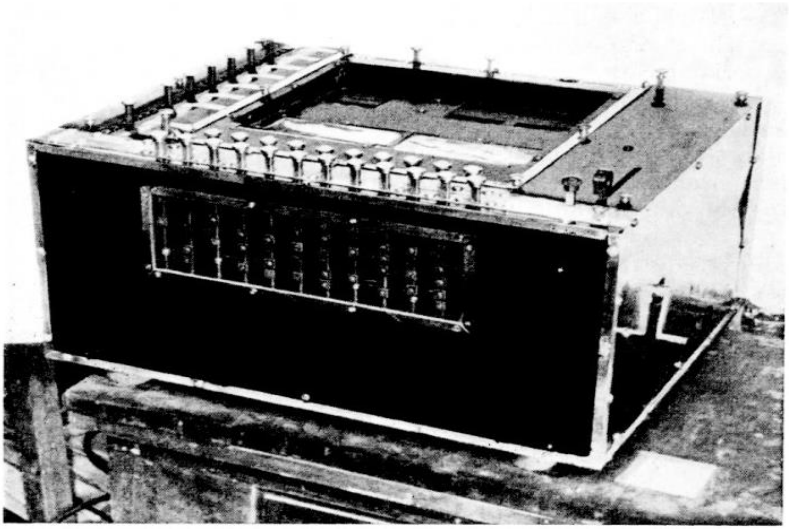
## 1. Giriş

Makine çevirisi kavramından bahsetmeden önce, çevirinin ne olduğundan bahsetmek gerekmektedir. Çeviri tanımını, Cicero'dan başlayarak (bkz. Göktürk, 2000), Jakobson (1958), Catford (1965), Newmark (1981) ya da Esen (2008), Köksal (2008), Munday (2012) ve Stolze (2013)'ün tanımlarını teker teker irdeleyerek de yapabiliriz. Ancak burada çeviriyi, belirli bir topluluk tarafından meydana getirilen kültür içerisinde o topluluk bireyleri tarafından birbirleriyle iletişim amacıyla kullanılan kodların, farklı bir topluluk tarafından oluşturulmuş kültür içerisine aynı etki ve sonuçları doğuracak şekilde eşdeğerlikli aktarımıdır şeklinde ifade etmenin yerinde olacağı düşünülmektedir. Daha basit hali ile kaynak dildeki bir metin yahut sözlü bir ifadeye ait iletişim kodlarının, hedef dilde aynı etki, tepki, anlam ve sonuçları doğuracak şekilde yeniden kodlanmasıdır denilebilir. Bu bağlamda, başka bir dile anlaşılır biçimde aktarılması işlemi ile bu işlem sonucunda elde edilen çıktılar olarak tanımlanabilir. Bu noktada, çevrilecek metin ya da sözlü ifade "kaynak dildeki veri" ; çevirisi yapılan ya da yapılması planlanan çıktılar ise "hedef dildeki veri" olarak düşünülebilir. Diğer bir yandan, bu her iki verinin kullanım amacına uygunluğu ve birbiri ile uyuşarak aynı tepki, etki ve sonuçları doğuracak şekilde yeniden kodlanmış olması ise bizlerin elde edilen o veriye ne derece güvenerek onu bilgi olarak kullanabileceğimizi belirler. Öte yandan çoğu zaman bir çeviride, kaynak dilde verilen mesajların en az kayıpla hedef dile aktarması istenir. Bu tür durumlarda en az hata payı ile kontrollü bir çeviri yapılması tercih edildiğinden, insan faktörüne güvenilir. Ancak, bu tür çevirilerin zamansal ve maddi açıdan bir bedeli mevcut olduğundan belirli ölçüde külfeti de olabilmektedir. Diğer bir yandan ise, hata faktörünün tolere edilebildiği ve aslen anlaşılmayan bir dildeki metnin ya da sözlü ifadenin içeriğine yönelik algılamanın ve öğrenmenin ön planda olduğu durumlar da mevcuttur. İşte bu tür durumlarda ise, makine çevirisi, maddi açıdan ek maliyetler getirmemesi ve aynı zamanda hızlı olması sebebiyle tercih edilebilmektedir.

## 2. Makine Çevirisinde İlkler

İlk makine çevirisinin somut olarak gerçekleştirilmesi 1933 yılında George Artsrouni adlı bir mucit tarafından yapılmıştır. Fransa'da geliştirilen bu aygıt her ne kadar mekanik ve oldukça ilkel ama bir o kadar karmaşık bir olsa da makine çevirisinin babası olarak kabul edilmektedir (Hutchins, 1995).

**Şekil 1**
İlk Çeviri Makinesi (Daumas, 1965)



Artsrouni'nin bu icadı her ne kadar günümüzde makine çevirisi yapan sistemlerden performans açısından oldukça uzak olsa da, kaynak dilde verilen kelimeleri arayıp bulma ve hedef dilde ifade etme gibi işlevlere sahip olan, ancak daha çok kelime kelime çeviri yapan mekanik sözlük benzeri bir aygıt olarak nitelendirilmekte ve ayrıca bir dildeki kayıtları diğer üç dildeki eşdeğerliklere çevirebildiği ifade edilmektedir (Corbe, 1960; Hutchins, 2004).

Makine çevirisinin ilk ortaya çıkışı yukarıdaki kısa tarihçesinden de anlaşıldığı üzere, "word to word" yani "kelimeden-kelimeye" şeklinde, kaynak dildeki bir kelimenin hedef dildeki eşdeğerliğinin bulunması şeklinde yapılmıştır. Ancak bu durum daha düzgün, modern ve işlevsel metin çevirilerine ihtiyaç duyulması sebebiyle, karmaşık dilbilgisel kuralların da devreye girmesini zorunlu kılarak sadece kelimeden-kelimeye aktarma yapılmasının oldukça ötesinde bir problem haline dönüşmüştür. Bu probleme üretilen ilk çözüm yolu ise mevcut dilbilgisel kuralların istatistiksel olarak hesaplanarak belirli kurallar çerçevesinde kaynak dilden hedef dile aktarımının yapılması şeklinde olmuştur. Soğuk savaş döneminde, Georgetown Üniversitesi ve IBM'in işbirliği ile 1954'te gerçekleştirilen Rusça-İngilizce makine çevirisinin, bu tip kural tabanlı bir algoritmaya dayalı olarak geliştirilen modern anlamdaki ilk makine çevirisi olduğu ve oldukça sınırlı sayıda kelime ile dilbilgisel kurala bağlı olarak çalıştığı bilinmektedir (Garvin, 1967). Ancak bu gelişme sonrasında ALPAC (Automatic Language Processing Advisory Committee) komitesinin 1966 yılında istenilen sonuçların alınamadığını gerekçe göstermesi neticesinde Amerikan Savunma Bakanlığı, Doğal Dil İşleme ve Makine Çeviresi alanındaki araştırma fonlarını kesmiş ve bu alandaki çalışmalar maalesef uzun bir süreliğine rafa kaldırılmıştır (National Research Council, 1966).

## 3. Makine Çevirisi Türleri

Makine çevirisini sınıflandırmada pek çok farklı kriter ve kıstasların kullanılabilmektedir. Ancak bu çalışmada, makine çevirisini faydalanılan temel tekniklere göre iki tipte incelenmiştir.

1. Statik Makine Çevirisi

2. Dinamik Makine Çevirisi

Bunlardan, Statik Makine Çevirisi hedef dil ile kaynak dilin sözdizimsel, anlambilimsel ve biçimbilimsel yapıların çeşitli kurallara göre matematiksel olarak hesaplanmasına dayanır. IBM ve Georgetown tarafında geliştirilen ilk modern makine çevirisi örneğinde olduğu gibi, ilk zamanlarda karşılaşılan makine çevirilerinin hemen hemen tamamı bu gruba dâhil edilebilir. Kelime tabanlı, ifade tabanlı ya da örneklem tabanlı makine çevirileri gibi önceden hesaplanmış ve belirlenmiş belirli kurallara göre çeviri yapan ve bu kuralları kendi kendine geliştirmesi ya da güncellemesi pek mümkün olmayan çeviri türlerinin tamamı bu gruptadır. Bu yaklaşımda genellikle kaynak dildeki metnin çözümlemesi yapılarak dilbilimsel kurallara göre hesaplanması ve daha sonrasında da buna bağlı olarak kaynak dilden hedef dile çözümlenen bu metnin aktarımı yapılır.

Dinamik Makine Çevirisi'nde ise, temeli her ne kadar yine matematiksel hesaplara dayanıyor olsa da bu hesaplar eklenen veriye ve öğrenim miktarına göre an be an değişebilmektedir. Ayrıca eldeki veri miktarı ve bu verinin kalitesiyle doğru orantılı olacak şekilde gelişme kaydedilebilmektedir. Dolayısıyla, sistemin sahip olduğu veri ne kadar detaylı ve çok ise çevirinin başarısı da o derece yüksek olabilmektedir. Dinamik makine çevirilerinde ise makine öğrenim algoritmaları ile yapay sinir ağlarından faydalanılmaktadır. Yapay sinir ağlarının keşfi ise aslında modern bilgisayarların ortaya çıkmasından çok daha önceleri meydana gelmiştir. Özellikle insan beyninin hesap yapma özelliği modellenerek oluşturulan ilk Yapay Sinir Ağı (YSA), ömrünü insan beyninin işleyişini kavramaya adayan Donald O. Hebb ile nörolog Waren McCulloch ve arkadaşları matematikçi Walter Pitts'in, nörobiyoloji alanındaki çalışmalarını, mühendislik alanına uyarlayarak insan beynindeki sinir hücrelerini taklit edebilecek bir yapı oluşturma macerası sonucunda 1943'te ortaya çıkmıştır (McCulloch & Pitts, 1943; Keskenler & Keskenler, 2017).

## 4. Yapay Sinir Ağları

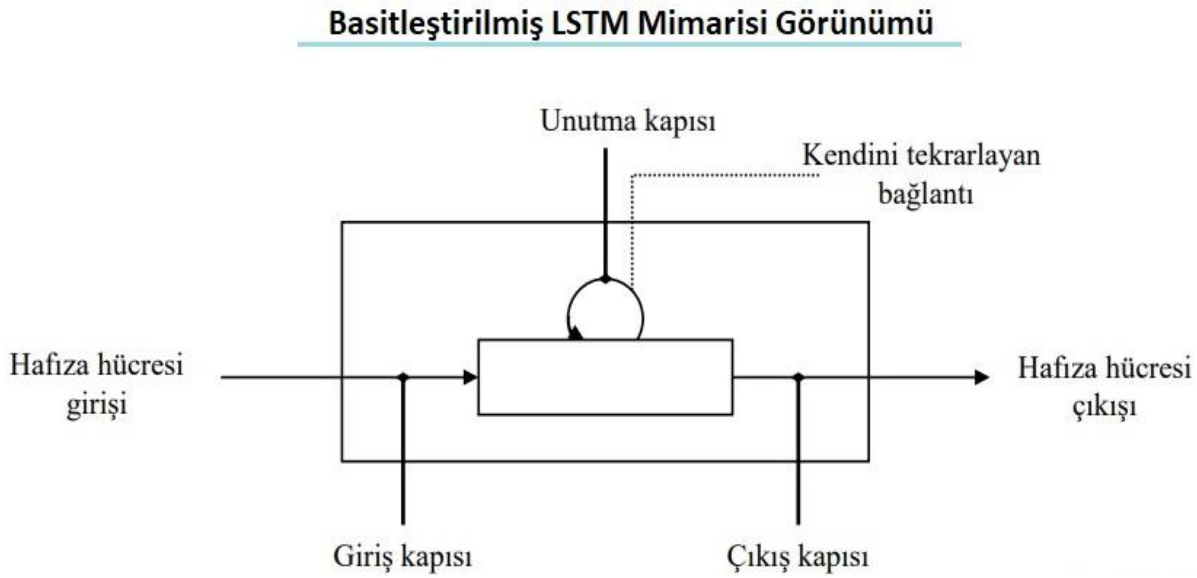
Gerçek bir sinir hücresini taklit ederek oluşturulan yapay sinir ağları aslında temelde toplama işlemi yapan küçük parçacıklar olarak düşünülebilir. Ancak YSA'larda bu toplama işlemi diğer standart toplama işlemlerinden biraz farklıdır. Yapay sinir ağlarında meydana gelen toplama işlemini diğer standart matematiksel toplayıcılardan ayıran en belirgin özelliğin, istenilen sonucu elde edebilmek için gerektiğinde

mevcut girdi sinyal değerlerinin istenildiği gibi kontrol edilebilmesi olduğu söylenebilir. Diğer standart matematiksel toplayıcılarda mevcut olmayan bu özellik, YSA'larda toplayıcıya bir eşik değerinin atanmasıyla meydana gelmektedir ve eşik değerine göre istenildiği takdirde her bir giriş sinyali ayrı ayrı kontrol edilip güncellenebilmektedir. Tabi bunun için bir geri besleme alması gerekmektedir.

## 5. LSTM ile Makine Çevirisi

Günümüz makine çevirilerinde genellikle 3 farklı tipte yapay sinir ağı mimarisinden faydalanılmaktadır. Bunlar kısaca; RNN (Tekrarlayan Yapay Sinir Ağı), LSTM (Uzun-Kısa Vadeli Bellek) ve CNN (Konveksiyonel Yapay Sinir Ağı) olarak adlandırılabilir. Bu çalışmada ise çok yüksek bir eğitim verisine sahip olmamamızdan ötürü LSTM (Uzun-Kısa Vadeli Bellek) mimarisi tercih edilmiştir.

**Şekil 2**
LSTM Mimarisi (Temür, 2019)



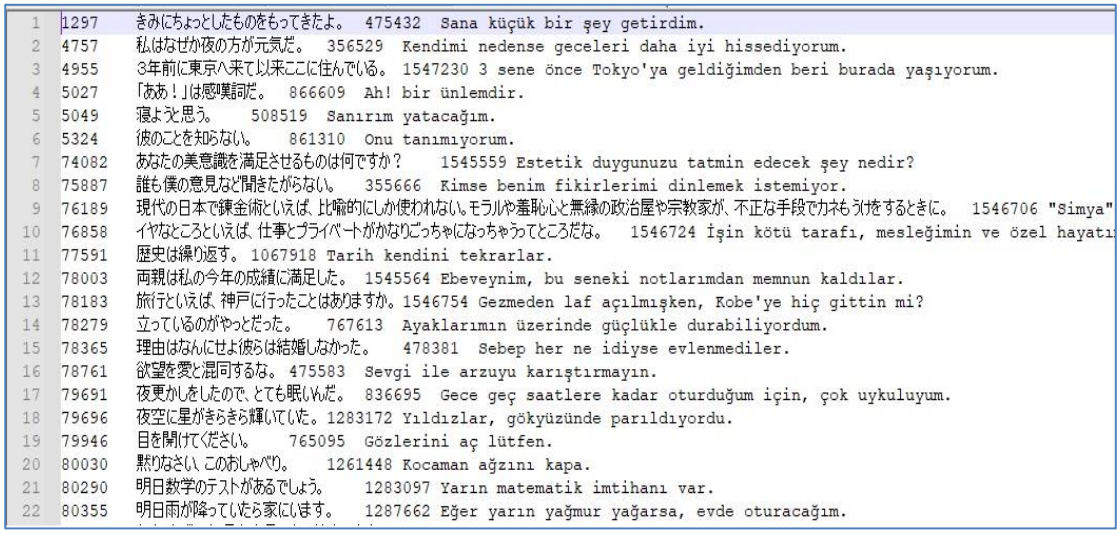

LSTM'de diğer yapay sinir ağları mimarisinden farklı olarak giriş ve çıkış kısımlarının yanı sıra bir de unutma kapısı mevcuttur. Böylece LSTM'de seçilen bilgiyi güncellemek ya da silmek mümkün olmakta ve bu sayede hangi bilginin tutulacağı ya da hangi bilginin tutulmayacağı kararlaştırılabilmektedir. Yukarıdaki şekilde; LSTM ağlarında bulunan kapılar ve hafıza hücresi görülmektedir. Kapılar hücreye erişimi kontrol eden ve tıpkı bir bilgisayarın belleğine "okuma", "yazma" ve "sıfırlama" işlemlerini yerine getiren bir denetleyici olarak düşünülebilir. Bu sayede mevcut bir değerin hangi zaman aralığında güncellenip güncellenmeyeceğine karar verilir (Temür, 2019).

## 6. LSTM Makine Çevirisi ve Eğitim Verisi

Öncelikle çalışmamızda, Türkçe ve Japonca eğitim verisi oluşturabilmek için Tatoeba sitesinden (https://tatoeba.org/) faydalanılmış ancak buradaki veriler gözden geçirilerek kısmi olarak çalışmaya dâhil edilmiştir. Daha sonra Türkçe kelime vektörel bilgileri için, önceden hesaplanmış 300 boyutlu bir Türkçe kelime vektör dosyasından faydalanılmıştır. Bilgin, Okan ve Uras (2020) tarafından hazırlanan bu dosya ise, Github sitesi (https://github.com/inzva/Turkish-GloVe) üzerinden indirilerek çalışmamıza dahil edilmiştir. Sonrasında model kurulumuna geçilmiş ve LSTM makine çevirisi için Kolukısa (2021)'de kullanılan kodlardan faydalanılmıştır. Ancak İngilizce-Türkçe için hazırlanmış olan bu kodların, önce Türkçe-Japonca LSTM çevirisi için adaptasyonu sağlanmıştır. Modelin çalıştırılması Google Colab üzerinden yapılmıştır. Veri düzenlemede ise Notepad ++ kullanılmıştır.

**Şekil 3**

Ham Veriler



```
1297    きみにちょっとしたものをもってきたよ。  475432  Sana küçük bir şey getirdim.
4757    私はなぜか夜の方が元気だ。  356529  Kendimi nedense geceleri daha iyi hissediyorum.
4955    3年前に東京へ来て以来ここに住んでいる。  1547230 3 sene önce Tokyo'ya geldiğimden beri burada yaşıyorum.
5027    「ああ！」は感嘆詞だ。  866609  Ah! bir ünlemdir.
5049    寝ようと思う。  508519  Sanırım yatacağım.
5324    彼のことを知らない。  861310  Onu tanımıyorum.
74082   あなたの美意識を満足させるものは何ですか？  1545559 Estetik duygunuzu tatmin edecek şey nedir?
75887   誰も僕の意見など聞きたがらない。  355666  Kimse benim fikirlerimi dinlemek istemiyor.
76189   現代の日本で錬金術といえば、比喩的にしか使われない。モラルや羞恥心と無縁の政治屋や宗教家が、不正な手段で力ネもうけをするときに。  1546706 "Simya"
76858   イヤなところといえば、仕事とプライベートがかなりごっちゃになっちゃうってところだな。  1546724 İşin kötü tarafı, mesleğimin ve özel hayatı
77591   歴史は繰り返す。1067918 Tarih kendini tekrarlar.
78003   両親は私の今年の成績に満足した。1545564 Ebeveynim, bu seneki notlarımdan memnun kaldılar.
78183   旅行といえば、神戸に行ったことはありますか。1546754 Gezmeden laf açılmışken, Kobe'ye hiç gittin mi?
78279   立っているのがやっとだった。  767613  Ayaklarımın üzerinde güçlükle durabiliyordum.
78365   理由はなんにせよ彼らは結婚しなかった。  478381  Sebep her ne idiyse evlenmediler.
78761   欲望を愛と混同するな。475583  Sevgi ile arzuyu karıştırmayın.
79691   夜更かしをしたので、とても眠いんだ。  836695  Gece geç saatlere kadar oturduğum için, çok uykuluyum.
79696   夜空に星がきらきら輝いていた。1283172 Yıldızlar, gökyüzünde parıldıyordu.
79946   目を開けてください。  765095  Gözlerini aç lütfen.
80030   黙りなさい、このおしゃべり。  1261448 Kocaman ağzını kapa.
80290   明日数学のテストがあるでしょう。  1283097 Yarın matematik imtihanı var.
80355   明日雨が降っていたら家にいます。  1287662 Eğer yarın yağmur yağarsa, evde oturacağım.
```

Yukarıda hazırlanan ham veriler görülmektedir. Ancak verinin direkt bu şekilde kullanılabilmesi mümkün değildir. İngilizce ve Türkçe gibi dillerde kelimeler arasında boşluk bırakılmasına rağmen Japoncada böyle bir durum söz konusu olmamaktadır. Bu durum kelimelerin belirli bir düzen dâhilinde ayrılması gerekliliğini doğurmuştur. Bunun en büyük nedeni, makine çevirilerinde makinenin kelimeleri ve diğer detayları anlamlandırabilmesi için parçalara ayırması gerektiğinden ve bu noktada da genellikle kelimeler arasındaki boşluklardan faydalanılmasından kaynaklanır. Diğer bir sorun ise, indirilen eğitim verisi her ne kadar iki dilli olsa da, yukarıda da görüldüğü üzere, bizlerin çalışmasına uygun olmayan rakamsal içerik ile sıralama hususunda da bir takım farklılıkları bünyesinde barındırmaktadır.

LSTM ve RNN makine çevirilerinde genellikle iki dilli eğitim verisi (parallel corpus dataset) kullanıldığında bu veriler birbirinden sadece TAB boşluğu ile ayrılmaktadır. Bu sebeple, verilerin öncelikle Python kodları vasıtasıyla temizleyerek uygun hale getirme zorunluluğu ortaya çıkmıştır. Rakamlardan arındırma ve "Türkçe + TAB boşluğu + Japonca" eşdeğerliği olarak sıraya koyma işlemini hazırladığımız Python kodları ile gerçekleştirildikten sonrasında ise veriler aşağıdaki hale gelmiştir.

**Şekil 4**

Düzenlenmiş Veriler

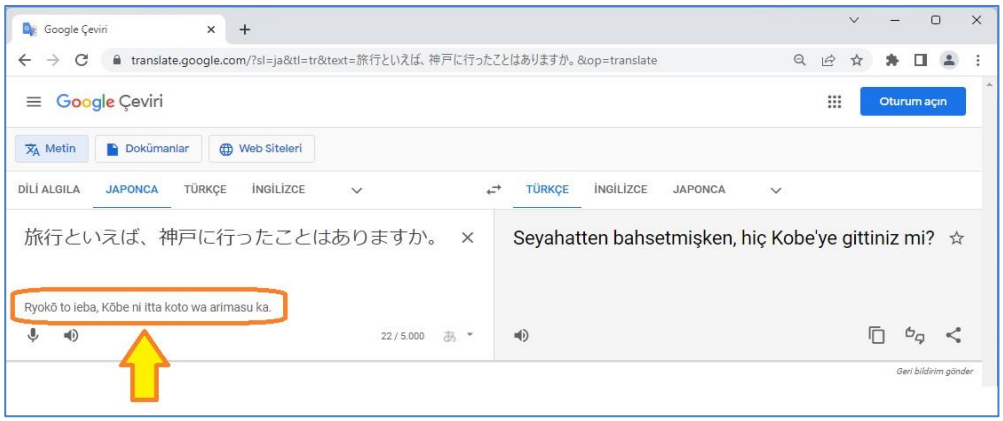
```
Sana küçük bir şey getirdim.    きみにちょっとしたものをもってきたよ。
Kendimi nedense geceleri daha iyi hissediyorum. 私はなぜか夜の方が元気だ。
3 sene önce Tokyo'ya geldiğimden beri burada yaşıyorum. 3年前に東京へ来て以来ここに住んでいる。
Sanırım yatacağım.  寝ようと思う。
Onu tanımıyorum.    彼のことを知らない。
Estetik duygunuzu tatmin edecek şey nedir?  あなたの美意識を満足させるものは何ですか？
Kimse benim fikirlerimi dinlemek istemiyor. 誰も僕の意見など聞きたがらない。
İşin kötü tarafı, mesleğimin ve özel hayatımın bu derece birbirine karışıyor olması.    イヤなところといえば、仕事とプライベートがかなりごっちゃになっちゃうってところだな。
Tarih kendini tekrarlar.    歴史は繰り返す。
Ebeveynim, bu seneki notlarımdan memnun kaldılar.   両親は私の今年の成績に満足した。
Gezmeden laf açılmışken, Kobe'ye hiç gittin mi? 旅行といえば、神戸に行ったことはありますか。
Ayaklarımın üzerinde güçlükle durabiliyordum.   立っているのがやっとだった。
Sebep her ne idiyse evlenmediler.   理由はなんにせよ彼らは結婚しなかった。
Sevgi ile arzuyu karıştırmayın. 欲望を愛と混同するな。
Gece geç saatlere kadar oturduğum için, çok uykuluyum.  夜更かしをしたので、とても眠いんだ。
Yıldızlar, gökyüzünde parıldıyordu. 夜空に星がきらきら輝いていた。
Gözlerini aç lütfen.    目を開けてください。
Kocaman ağzını kapa.    黙りなさい、このおしゃべり。
Yarın matematik imtihanı var.   明日数学のテストがあるでしょう。
Eğer yarın yağmur yağarsa, evde oturacağım. 明日雨が降っていたら家にいます。
Sabah dışarıya çıkmadan önce her zaman hava durumunu izlerim.   毎朝必ず天気予報を見てから外出します。
Mahjong taşları çok güzeller.   麻雀牌はとってもきれいなものです。
Mahjong genellikle dört kişi oynanan bir oyun.  麻雀は普通、四人で遊ぶゲームです。
Mahjong dünyada çok popüler olan oyunlardan biri.   麻雀は世界でとても有名な、ゲームのひとつです。
Mahjong en ilginç oyunlardan biri.  麻雀は最も面白いゲームのなかのひとつです。
Mahjong oynamayı biliyor musun? 麻雀のやり方を知ってる？
Mahjong'u çok seviyorum.    麻雀が大好きです。
Gerçek savaş bu hikâyeden daha çok korkunç. 本当の戦争はこの話よりもずっと恐ろしい。
Çok üzgünüm.    本当にすいません。
Hayalim çok güçlü bir Mahjong oyuncusu olmak.   僕の願いはとっても強い麻雀打ちになることです。
Benim onu iyi tanımam gerektiğini söylüyorsun ama ben onunla daha geçen hafta tanıştırıldım.   僕が彼をよく知っているはずだと君は言うが、実際は僕は先週彼に紹
Hokkaido'da şu sıralar kar yağıyor olmalı.  北海道では今ごろ雪が降っているだろう。
Annem yavaş konuşur.    母はゆっくり話す。
```

TAB Boşluğu

Ancak veriler bu duruma getirilse de, Japonca eşdeğerliklerinde kelimeler arasında boşluklar mevcut olmadığından veri henüz makinenin eğitimi için yeterli özelliklere sahip değildir. Aslında bu boşluk sorunu "Google Translate" için de geçerli bir durumdur. Yukarıdaki Japonca cümlelerden herhangi birini Google'ın çeviri modülüne atıldığında aşağıdaki şekilde bir sonuçla karşılaşılır.

**Şekil 5**

Google Translate'in Japonca Çeviride Boşluk Kullanımı




Dolayısıyla, Google Translate de görüldüğü şekilde Japonca bir cümle her ne kadar orijinalinde boşluksuz olsa da, Google bunu Latin harfleriyle (romaji) boşluklu hale dönüştürmektedir.

Aslında Japonya'da da bu tür kullanımlarda doğan ihtiyaçtan dolayı, Japon Dili ve Dilbilimi Ulusal Enstitüsü (NINJAL) tarafından UniDic [2]; Taku Kudo ile Nippon Telegraph and Telephone Corporation tarafından ortak yapılan projelendirme neticesinde MeCab [3] ve McCann (2020) tarafından ise Fugashi gibi Japonca Doğal Dil İşleme - DDİ kütüphaneleri (*Japanese NLP libraries*) oluşturulmuştur[4]. Çalışma öncesinde bu kütüphaneler sıra ile denenmiş ancak verimiz oldukça düşük seviyelerde olduğundan istenilen sonuçları elde etme hususunda verim alınamamıştır. Bunun en büyük sebeplerinden biri ise bu kütüphanelerin hemen hemen hepsinin çok fazla bölümleme yapmalarıdır. Bütünü çok fazla parçacık haline dönüştürdüklerinden, özellikle uzun cümlelerde makinenin dekoder kısmına giren Türkçe cümledeki parça sayısı ile dekoderden çıkan Japonca cümledeki parça sayısı arasında oldukça büyük fark meydana gelmekte ve bu da özellikle çok yüksek olmayan eğitim verisi ile çalışılırken çeviri hatalarına sebebiyet vermektedir. Aşağıda bu kütüphaneler kullanıldığındaki parça sayısı ile Google Translate tarafından yapılan bölümlemedeki parça sayısı görülmektedir.

**Şekil 6**

MeCab, Unidic, Fugashi Bölümlemesi


[2] Bkz. : https://clrd.ninjal.ac.jp/unidic/ ve Python versiyonu için https://pypi.org/project/unidic/

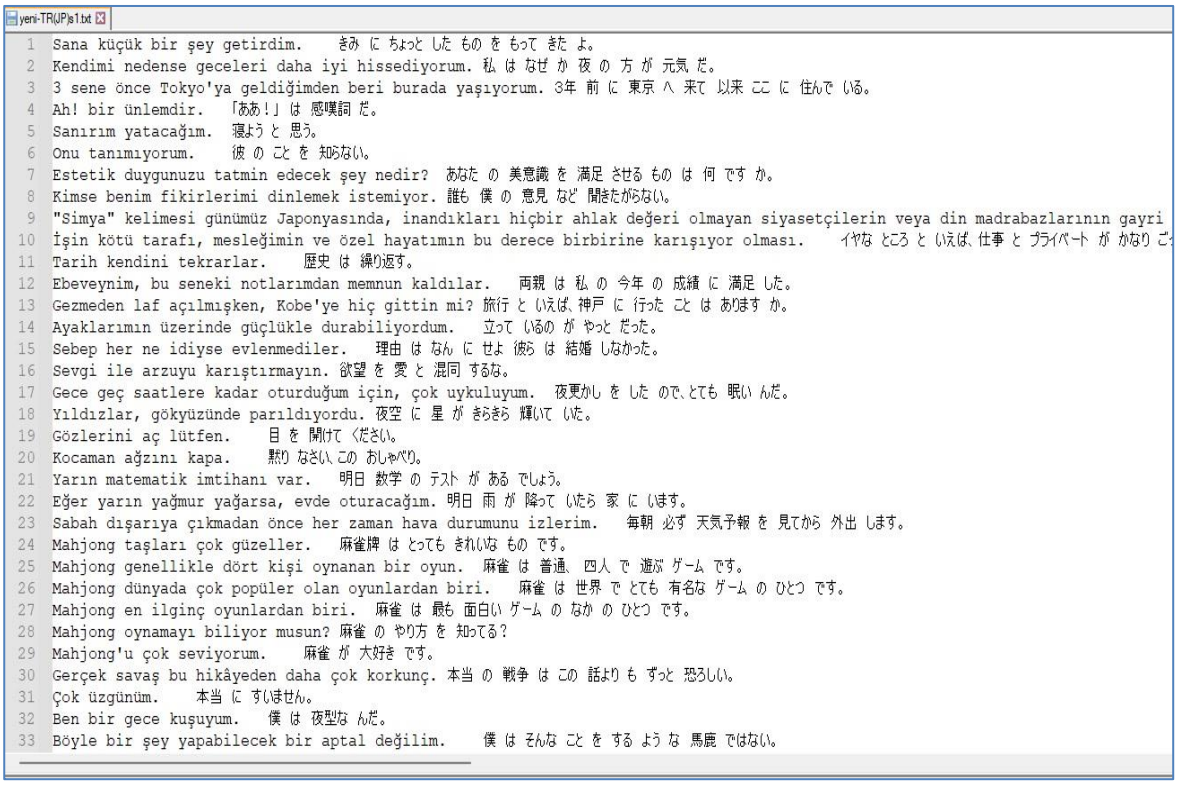
[3] Bkz.: https://taku910.github.io/mecab/ ve Python versiyonu için https://pypi.org/project/mecab-python3/

[4]Burada adı geçen kütüphaneler haricinde de *ChaSen*, *nagisa*, *Sudachi*, *Ginza*, *vb.* gibi daha pek çok Japonca DDİ kütüphanesi de mevcuttur. (Bkz. : https://towardsdatascience.com/an-overview-of-nlp-libraries-for-japanese-be1805837143)

Bu sebeple eğitim verisi satır satır elden geçirilerek, her satıdaki bileşenlerin Japonca dilbilgisel kurallar da göz önünde bulundurularak tek başına anlamlı bütünler oluşturacak şekilde manüel olarak boşlukla ayrılıp yeniden yapılandırılması yoluna gidilmiştir. Bu işlem sonrasında elde edilen eğitim verisinin anlık görüntüsü ise aşağıdaki şekilde olmuştur.

**Şekil 7**

Anlamlı Parçacıklar Olarak Boşlukla Ayrılmış Japonca Veri




## 7. Eğitim Verisi ve Model Kalibrasyonu

Eğitim verisi ve LSTM makine çeviri kodlarını Google Colab üzerinde aktardıktan sonraki aşamada makineye bu verinin 10'da 9 unu eğitimi ve 10'da 1'ni de eğitim sonuçlarını doğrulaması için ayırdık. Toplamda 2075 satırdan oluşan ve önce Türkçe, tab boşluğunun ardından ise bu Türkçe cümlelerin Japonca eşdeğerliklerin bulunduğu eğitim verisi ile kurulan LSTM modelin eğitimine başlanmıştır. İlk seferinde; BATCH_SIZE=32, EPOCHS=150 ve LSTM_NODES=128 olacak şeklinde model çalıştırıldığında "Overfitting" ("aşırı öğrenme" ya da "ezberleme") denilen sorunla karşılaşıldığı görülmüştür. Elbette bu aslında beklenilen bir durumdur çünkü düşük bir eğitim verisi 150 kez tekrarlatıldığında makine ezber yapacaktır. Modelde aşırı öğrenme olduğu durumlarda çeviri doğruluk oranı %100 olarak çıkmaktadır ancak modele farklı bir girdi verildiğinde ise istenilen sonuçları almak mümkün olamamaktadır. Bir sonraki denemede, LSTM_NODES (LSTM düğüm sayısı) haricindeki tüm diğer değerler aynı kalacak şekilde model aşağıdaki gibi güncellendiğinde ise; BATCH_SIZE=32, EPOCHS=150 ve LSTM_NODES=16 çok düşük bir başarı oranı (yaklaşık %10 altında) elde edilmiştir. 3. denemede ise, bu kez LSTM düğüm sayısını 2 katına çıkarıp 32 olarak belirlendiğinde; BATCH_SIZE=32, EPOCHS=150 ve LSTM_NODES=32 başarı oranının artığı gözlemlenir. Ancak buradaki en büyük problem EPOCH sayısının çok fazla olmasıdır. Diğer bir yandan LSTM_NODES sayısı (LSTM düğüm sayısı) ile makine öğrenmesi arasında da doğru bir orantı bulunmaktadır. Bir sonraki denemede ise modelin LSTM_NODES sayısını çoğaltıp, EPOCH sayısını azaltılarak kalibre etme yoluna gidilmiştir. BATCH_SIZE=32, EPOCHS=15 ve LSTM_NODES=386 olarak çalıştırıldığında, modelin daha verimli ve istenilene yakın sonuçlar elde ettiği görülmüştür. Aşağıda, makine çevirisi için kurulan LSTM modelinin özeti mevcuttur.

**Şekil 8**

LSTM Model Özeti

```
In [70]: model.summary()

Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 25)]         0           []

 input_4 (InputLayer)           [(None, 38)]         0           []

 embedding_2 (Embedding)        (None, 25, 300)      1677300     ['input_3[0][0]']

 embedding_3 (Embedding)        (None, 38, 386)      1997164     ['input_4[0][0]']

 lstm_2 (LSTM)                  [(None, 386),        1060728     ['embedding_2[0][0]']
                                 (None, 386),
                                 (None, 386)]

 lstm_3 (LSTM)                  [(None, 38, 386),    1193512     ['embedding_3[0][0]',
                                 (None, 386),                     'lstm_2[0][1]',
                                 (None, 386)]                     'lstm_2[0][2]']

 dense_1 (Dense)                (None, 38, 5174)     2002338     ['lstm_3[0][0]']

==================================================================================================
Total params: 7,931,042
Trainable params: 7,931,042
Non-trainable params: 0
```

Model bu ayarlarla çalıştırıldığında aşağıdaki şekilde ekran çıktıları öncelikle edilir. Bu noktada eğer aşağıdaki tablodaki "val_accuracy" değerine, diğer bir deyişle test verisi doğruluk oranına dikkat edilecek olursa 10. Epoch'tan sonra düşüş yaşanmakta olduğundan, Epoch sayısının sadece 10 olarak belirlenmesinin daha uygun olacağı düşünülebilir. Ancak devamında, 11. Epoch'tan itibaren her ne kadar val_accuracy değerinde düşüş yaşansa da, "loss" yani kayıp değerlerinde ciddi oranda bir azalma meydana geldiğinden bu şekilde bırakılması tercih edilmiş ve devamında ise eğitilen bu veri ile makinenin ne derece doğru bir çeviri yaptığı kısmına geçilmiştir.

**Şekil 9**

LSTM Modelinin Çalıştırılması

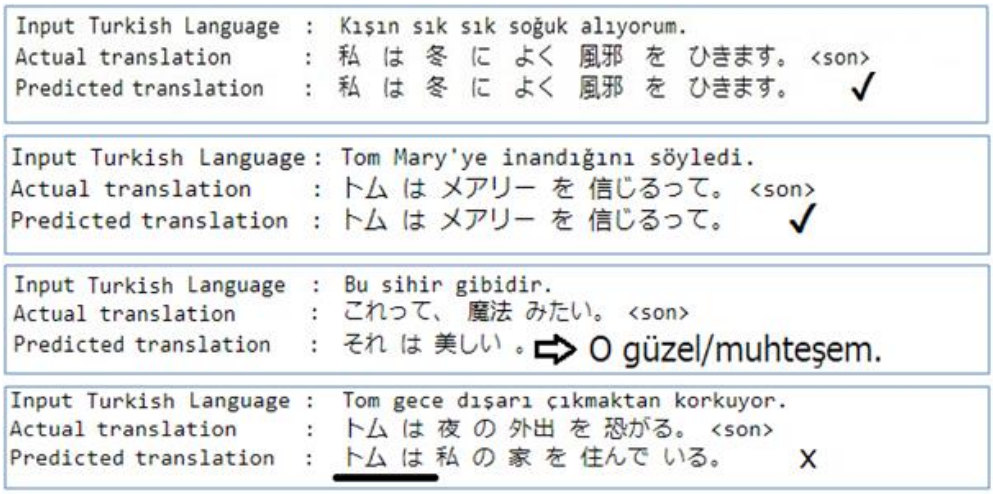
```
In [72]: r = model.fit(
             [encoder_input_sequences, decoder_input_sequences],
             decoder_targets_one_hot,
             batch_size=BATCH_SIZE,
             epochs=EPOCHS,
             validation_split=0.1,
         )

Epoch 1/15
59/59 [==============================] - 70s 1s/step - loss: 2.0277 - accuracy:
0.7486 - val_loss: 2.6433 - val_accuracy: 0.6097
Epoch 2/15
59/59 [==============================] - 62s 1s/step - loss: 1.4119 - accuracy:
0.7886 - val_loss: 2.6128 - val_accuracy: 0.6456
Epoch 3/15
59/59 [==============================] - 67s 1s/step - loss: 1.3290 - accuracy:
0.7982 - val_loss: 2.6238 - val_accuracy: 0.6636
Epoch 4/15
59/59 [==============================] - 64s 1s/step - loss: 1.2652 - accuracy:
0.8034 - val_loss: 2.6076 - val_accuracy: 0.6643
Epoch 5/15
59/59 [==============================] - 65s 1s/step - loss: 1.2051 - accuracy:
0.8075 - val_loss: 2.5923 - val_accuracy: 0.6680
Epoch 6/15
59/59 [==============================] - 67s 1s/step - loss: 1.1490 - accuracy:
0.8131 - val_loss: 2.6074 - val_accuracy: 0.6698
Epoch 7/15
59/59 [==============================] - 64s 1s/step - loss: 1.0956 - accuracy:
0.8179 - val_loss: 2.6284 - val_accuracy: 0.6743
Epoch 8/15
59/59 [==============================] - 64s 1s/step - loss: 1.0416 - accuracy:
0.8230 - val_loss: 2.6281 - val_accuracy: 0.6741
Epoch 9/15
59/59 [==============================] - 66s 1s/step - loss: 0.9864 - accuracy:
0.8282 - val_loss: 2.6383 - val_accuracy: 0.6745
Epoch 10/15
59/59 [==============================] - 63s 1s/step - loss: 0.9305 - accuracy:
0.8342 - val_loss: 2.6779 - val_accuracy: 0.6764
Epoch 11/15
59/59 [==============================] - 63s 1s/step - loss: 0.8766 - accuracy:
0.8392 - val_loss: 2.7038 - val_accuracy: 0.6751
Epoch 12/15
59/59 [==============================] - 65s 1s/step - loss: 0.8212 - accuracy:
0.8457 - val_loss: 2.7315 - val_accuracy: 0.6726
Epoch 13/15
59/59 [==============================] - 65s 1s/step - loss: 0.7658 - accuracy:
0.8530 - val_loss: 2.7854 - val_accuracy: 0.6732
Epoch 14/15
59/59 [==============================] - 63s 1s/step - loss: 0.7092 - accuracy:
0.8610 - val_loss: 2.8106 - val_accuracy: 0.6728
Epoch 15/15
59/59 [==============================] - 64s 1s/step - loss: 0.6553 - accuracy:
0.8695 - val_loss: 2.8124 - val_accuracy: 0.6683
```

## 8. Türkçe – Japonca LSTM Çeviri Sonuçları

Modelin eğitimi tamamlandıktan sonra girdi kısmına Türkçe cümleler girildiği takdirde, aşağıdaki şekilde öncelikle normalde olması gereken çeviri gereken *"Actual translation"* kısmında ve eğitilen modelin yapmış olduğu çeviri ise *"Predicted translation"* kısmında aşağıdaki şekilde görüntülenmiştir.

**Şekil 10**

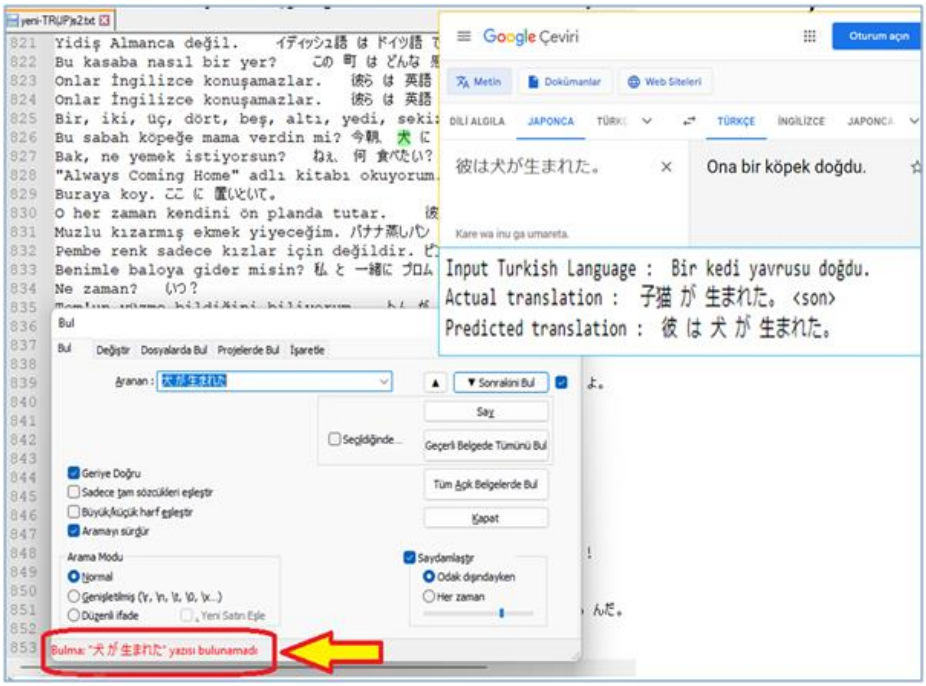
LSTM Çeviri Sonuçları




Yukarıdaki sonuçlardan da görüldüğü üzere düşük bir veriye sahip olmamıza rağmen, belirli ölçüde tatmin edici sonuçlar alabilmek mümkün olmuştur. Ayrıca, aşağıda görüldüğü gibi hatalı olan çevirilerin içerisinde de kısmi olarak doğru diyebileceğimiz yerler bulunmaktadır. Örneğin aşağıdaki çeviride, "Bir kedi yavrusu doğdu" Türkçe girdisi modele verildiğinde, "köpek" kavramının doğabilen canlı bir varlık olduğunu makine kendi çıktısı olarak üretmiş olup, kedi benzeri bir varlık olan "köpek" ile "doğmak" eylemini öğrendiklerinden yola çıkarak birleştirmeyi denemiş ve tam olarak doğru bir şeklide bunu yapamamış olsa da belirli ölçüde anlaşılabilir bir cümleyi kendi kendine üretebilmiştir. Bu durum ise, aslında daha yüksek miktarlardaki bir veri ile çalışıldığı takdirde daha güvenilir sonuçları elde etmenin mümkün olduğunu işaret etmektedir.

**Şekil 11**

Modelin Tarafından Üretilen "Köpek" Kavramı



## 9. Tartışma ve Sonuç

Yapay Sinir Ağlarının makine çevirisinde kullanımına ilişkin olarak hazırlanan bu çalışmada, LSTM mimarisi kullanılarak Türkçe → Japonca makine çevirisi gerçekleştirilmiştir. Çalışmada oldukça sınırlı miktarda bir eğitim verisi kullanılmış ve özellikle hedef dilde Japonca gibi kelimler arasında boşluk bulunmayan bir dil bulunduğu durumda yapılması gerekenler ele alınmıştır. Yapılan uygulama neticesinde ise, düşük eğitim verisi ile çalışılırken, EPOCH denilen veri üzerinde çalışılacak tur sayısını arttırmak yerine, LSTM düğüm sayısının (LSTM_NODES) çoğaltılması neticesinde daha verimli sonuçlar alınacağı ve aynı zamanda bu şekilde aşırı öğrenme olarak bilinen "Overfitting" sorununun da belirli ölçüde önüne geçilebileceği sonucuna varılmıştır.

## Yazar Katkıları

Yazar makalenin son hâlini okudu ve onayladı.

## Çıkar Çatışması

Yazar çıkar çatışması olmadığını beyan etmektedir.

## Kaynakça

Bilgin A., Okan V. ve Uras M. (2020). Türkçe GloVe - Repository for Turkish GloVe Word Embeddings . [internet] [erişim:02.09.2022 ] https://github.com/inzva/Turkish-GloVe

Catford, J. C. (1965). *A Linguistic Theory of Translation*. Oxford: Oxford University Press.

Corbe M. (1960). La Machine à Traduire Française Aura Bientôt Trente Ans. *Automatisme,* 5(3): 87-91.

Daumas M. (1965). Les Machines à Traduire de Georges Artsrouni. *Revue d'Histoire des Sciences et de Leurs Applications*; 18 (3): 283-302.

Esen Eruz S. (2008). *Akademik Çeviri Eğitimi: Çeviri Amaçlı Metin Çözümlemesi*. Multilingual, İst.

Garvin PL. (1967). The Georgetown-IBM experiment of 1954 [İnternet]. Çanakkale; 1967 [erişim tarihi:29.09.2022]. https://aclanthology.org/www.mt-archive.info/Garvin-1967.pdf

Göktürk, A., (2000). *Çeviri: Dillerin Dili*. Yapı Kredi Yayınları, İstanbul.

Hutchins W John. (1995). Machine Translation: A Brief History. In E.F.K.Koerner and R.E.Asher(Eds.*), Concise History Of The Language Sciences: From The Sumerians To The Cognitivists*. Oxford: Pergamon Press.

Hutchins J. (2004). Two Precursors of Machine Translation: Artsrouni and Trojanskij. *International Journal of Translation*, 16 (1): 11–31.

Jakobson E. (1958). *Translation a Traditional Craft: An Introductional Craft*. Classica et mediaevalia: Dissertationes Serries, ISSN 0906-2912, Gyldendal Publisher.

Kalchbrenner N, Blunsom P. (2013). Recurrent Continuous Translation Models. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA 2013; October: 1700-1709. https://www.aclweb.org/anthology/D13-1176.pdf

Keskenler M. F. & Keskenler E. F. (2017). Geçmişten Günümüze Yapay Sinir Ağları ve Tarihçesi. *Takvim-i Vekayi*, 5 (2), 8-18. Retrieved from https://dergipark.org.tr/tr/pub/takvim/issue/33375/346279

Kolukısa A. A. (2021). *Makine Çevirisi Uygulama Örneği (A Case Study of Machine Translation)* [Tezsiz Yüksek Lisans Bitirme Projesi: İzmir Katip Çelebi Üniversitesi, Yazılım Mühendisliği Ana Bilim Dalı (yayınlanmamış bitirme projesi)].

Köksal D. (2008). *Çeviri Eğitimi: Kuram ve Uygulama*. Nobel Yayın Dağıtım, Ank.

McCulloch W. S. & Pitts W. A. (1943). A logical calculus of the ideas immanent in nervous activity. *Buttetin of Mathematics and Biophysics*, 5, 115-133.

Paul McCann. (2020). fugashi, a Tool for Tokenizing Japanese in Python. In Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS), pages 44–51, Online. Association for Computational Linguistics. (https://aclanthology.org/2020.nlposs-1.7/)

Mecab-python3 1.0.5 [internet] [son erişim tarihi: 29.09.2022] https://pypi.org/project/mecab-python3/

Munday J. (2012). *Introducing Translation Studies Theories and Applications (4th Edt.)*, Routledge, NY.

Newmark, P. (1981). *Approaches to Translation*. Pergamon, Oxford and New York.

Stolze R. (2013). *Çeviri Kuramları: Giriş* (6. Baskıdan Çeviri, Çev.: Dr. Emra Durukan). Değişim Yayınları, Ist.

Tatoeba [internet] [son erişim: 02.09.2022] https://tatoeba.org/

Temür A. S. (2019*). İşletmelerin Satış Bütçelerinin Oluşturulmasında Arıma, LSTM, Hibrit Modellerin Karşılaştırılması: Üretim İşletmesi Örneği* [Doktora tezi: Sakarya Üniversitesi, İşletme Anabilim Dalı; 2019]. https://acikerisim.sakarya.edu.tr/handle/20.500.12619/68677