İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TÜRKÇE'DE VARLIK İSMİ TANIMA

YÜKSEK LİSANS TEZİ

Asım GÜNEŞ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

HAZİRAN 2018

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TÜRKÇE'DE VARLIK İSMİ TANIMA

YÜKSEK LİSANS TEZİ

Asım GÜNEŞ
(504131556)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Ahmet Cüneyd TANTUĞ

HAZİRAN 2018

İTÜ, Fen Bilimleri Enstitüsü'nün 504131556 numaralı Yüksek Lisans Öğrencisi Asım GÜNEŞ, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "TÜRKÇE'DE VARLIK İSMİ TANIMA" başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Doç. Dr. Ahmet Cüneyd TANTUĞ** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Prof. Dr. Banu DİRİ** Yıldız Teknik Üniversitesi	.....
	Dr. Öğr. Üyesi Gülşen ERYİĞİT İstanbul Teknik Üniversitesi	.....
		.....

Teslim Tarihi : **4 Temmuz 2018**
Savunma Tarihi : **4 Haziran 2018**

Aileme,

ÖNSÖZ

Tez çalışmamda yardımlarını esirgemeyen değerli hocam Doç. Dr. Ahmet Cüneyd TANTUĞ'a, hayatımın her anında bana yardımcı olan anne ve babama, varlıklarıyla bana güç veren sevgili eşim Fatma, kızlarım Nil ve Bilge'ye teşekkür ederim.

Haziran 2018 Asım GÜNEŞ

İÇİNDEKİLER

Sayfa

KISALTMALAR

DDİ : Doğal Dil İşleme
VİT : Varlık İsmi Tanıma
YSA : Yapay Sinir Ağları
RNN : Tekrarlayan Yapay Sinir Ağları
LSTM : Uzun Kısa-Dönem Hafıza
BLSTM : Çift Yönlü Uzun Kısa-Dönem Hafıza
DBLSTM : Derin Çift Yönlü Uzun Kısa-Dönem Hafıza

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

TÜRKÇE'DE VARLIK İSMİ TANIMA

ÖZET

Uzun yıllardır insanoğlunun makineler ile kendi doğal dilini kullanarak etkileşime geçmek istemesi ve bu konuda yapmış olduğu araştırmalar yapılan en heyecanlı araştırmalar arasında yer almaktadır. Robotlarla ya da bilgisayar sistemleri ile konuşmak, onların bizi anlaması, bizim adımıza araştırmalar yapması yapılan birçok bilim-kurgu çalışmasının ana temalarından olmuştur. Makineler ile insanlar arasındaki etkileşimi doğal dil üzerinden yapılmasını sağlamak, doğal dilin anlamlandırılması, diller arasında otomatik çevriminin yapılması, doğal dil ile yazılmış metinlerden bilgi çıkarımı yapılması gibi görevler Doğal Dil İşleme (DDİ) araştırma alanın çalışma konularıdır.

Yazılı metin içerisinden bilgi çıkarımı görevi ele alındığında metin içerisinde geçen tanımlı varlıkların tespit edilmesi önemli bir görev adımıdır. Metin içerisinde geçen kişi, organizasyon, yer adları, sayısal değerler, tarihsel ifadelerin tespit edilerek işaretlenmesi Varlık İsmi Tanıma (VİT) olarak adlandırılmaktadır. VİT görevinde tanınacak varlıklar ihtiyaçlara göre değişiklik gösterebileceği gibi VİT çalışmalarında genel olarak ENAMEX (kişi, organizasyon ve yer isimleri), NUMEX (sayısal eğerler, parasal ifadeler, yüzdeli ifadeler), TIMEX (tarih ve zaman ifadeleri) varlık kategorilerinin işaretlenmesi olarak değerlendirilmektedir. Bununla birlikte özelleşen sistemlerde e-posta adresleri, telefon numaraları, kitap başlıkları, proje isimleri gibi kategoriler de Varlık İsmi olarak ele alınabilmektedir. Örnekleri çoğaltmak gerekirse biyoinformatik ve kimya alanlarındaki metinlerden protein isimleri, RNA, DNA, hücre bilgileri, ilaç adları, kimyasal adlarının tespit edilmesi de Varlık İsmi Tanıma görevi olarak değerlendirilmektedir.

VİT ile ilgili çalışmalar 1990'larda başlamış ve 1996 sonrasında hız kazanmıştır. Türkçe için ise bilinen ilk çalışma ise 1999 yılında gerçekleştirilmiş olup 2009 sonrasında bu alandaki çalışmalar ilgi görmeye başlamış ve hız kazanmıştır. İngilizce gibi dillerde hem çalışma sayısının fazla olması hem de İngilizce'nin Türkçe gibi dillere göre daha basit biçimbilimsel yapısından dolayı VİT genel olarak çözülmüş bir problem olarak görülmektedir. Öte yandan Türkçe, Fince gibi dillerde dilin ek yapısı VİT görevini karmaşık hale getirmektedir. Bu durum VİT konusunu Türkçe'de halen güncel bir problem haline getirmektedir.

VİT alanında yapılan çalışmalar incelendiğinde ilk çalışmaların kural tabanlı sistemlerden oluştuğu gözlemlenmektedir. İlerleyen çalışmalarda örnek veri kümelerinin artması ile birlikte kural tabanlı geliştirilen sistemler yerlerini istatistiksel sistemlere bırakmışlardır. Türkçe'deki istatistiksel yöntemler ile geliştirilen sistemlerde özellikle makine öğrenmesi yöntemlerinden Koşullu Rastgele Değişkenler (CRF) yöntemi gerçeklenen sistemler ön plana çıkarken diğer dillerde Yapay Sinir Ağı temelli yöntemler sıklıkla kullanılmakta ve başarılı sonuçlar elde edildiği gözlemlenmektedir.

Yapay Sinir Ağları (YSA) insan sinir ağları model alınarak tasarlanan bir makine öğrenmesi metodolojisidir. Yapay sinir ağlarının temel yapısı insan sinir hücresi, nöron, model alınarak gerçeklenen yapay nöronlardan oluşmaktadır. Yapay Sinir Ağı alanındaki çalışmalar incelendiğinde bu alandaki teorik çalışmaların 1960'larda başlamış olmasına rağmen birçok teorik çalışmanın uygulanması, donanımlardaki yeni gelişmelerle ile birlikte son yıllarda mümkün hale gelmiştir.

Yapay Sinir Ağlarında nöronlar genellikle tekil olarak kullanılmazlar, problemin çözümüne bağlı olarak nöronlar farklı bağlantı modelleri ile birbiriyle ilişkilendirilerek kullanılırlar. Özellikle yapay sinir ağı yapılarının katman sayısı ve nöron sayılarının arttırılması ile birlikte oluşturulan Derin Öğrenme Sistemleri görüntü işlemeden, doğal dil işlemeye birçok araştırma alanındaki problemlerin çözümünde başarılı sonuçlar ortaya koymaktadır.

Bu tezin amacı Türkçe'de dile özgü özellikleri de kullanarak yapay sinir ağı tabanlı bir Varlık İsmi Tanıma Sistemi tasarlamak ve gerçekleştirmektir. Tez çalışması kapsamında gerçeklenen VİT Sisteminin başarısının değerlendirilebilmesi için Türkçe'de birçok çalışmada kullanılan bir Türkçe VİT Derlemi kullanılarak öğrenme ve başarım ölçüm işlemleri gerçekleştirilmiştir. Önerilen yapay sinir ağı tabanlı Türkçe VİT sistemi en iyi modelde %93.69 F_1 puanı ulaşılmıştır. Gözlemlediğimiz kadarıyla çalışma kapsamında önerdiğimiz en iyi modelde ulaşılan sonuç, literatürde en başarı sonuç olarak karşılaştığımız olarak ulaşılan %93.59 değerini az da olsa geçerek literatürdeki en başarılı Türkçe VİT sonucu haline gelmiştir.

NAMED ENTITY RECOGNITION IN TURKISH

SUMMARY

For decades humankind dreams on interacting with machines via their natural spoken languages. There are several interesting research project related to this topic. Speaking with machines or computer systems, making them understand natural languages, solving the problems for humans are the main theme of many science fiction books and movies. Making the human-machine interaction with natural languages, automated language translations, semantic analysis of textual data and many other natural language centric operations are tasks related to Natural Language Processing research area.

When the topic comes to extracting information from textual content recognition of named entities from the text is a very important step for the task. Recognition of person, organization or location information, numerical values, date and time expressions from the given natural language text is called Named Entity Recognition (NER). In many NER task recognition of person, organization or location information is called ENAMEX, recognition of numerical values, money and percentage expressions called NUMEX and recognition of date and time expressions called TIMEX entity types. Besides NER task is not limited with ENAMEX, NUMEX and TIMEX entity types.

NER studies begin in 1990's and increase popularity after 1996. Besides the first known study about NER in Turkish Language made in 1999 and popularity of NER systems for Turkish Language increase after 2009. Languages like Turkish or Finnish are agglutinative languages. They have complex morphological and affix based structures. In contrast to Turkish or Finnish, English has more simple language and morphological structure. This language structure of English easier to solve many Natural Language Processing task including NER in English. There are many research about NER task for English. It could be accepted that NER is a solved task for English. On the other hand languages like Turkish or Finnish has a more complex morphological architecture then English, besides researches are still limited in Turkish. This makes NER task still a hot topic for Turkish Language.

When researches about NER tasks are analyzed the first studies about NER are task recommends rule based systems. The researchers proposed that it's easier to design rule based systems when training data about the task is limited. On the other hand creating hand crafted rules to detect and tag named entities is a complex task. Besides rule based systems also suffers for context dependency. When changing the context of the system, performance of the system decreases gradually.

Recent studies in NER task focus on statistical and machine learning methods. There are featured studies using especially Conditional Random Fields (CRF) in Turkish NER tasks. On the other hand in many other languages NER systems based on artificial neural networks and deep learning creates outstanding performance in NER task.

Artificial Neural Networks (ANN) is a machine learning methodology which models human neural system. In human neural system neuron cells are the basic architectural structure. In artificial neural networks a neuron like structure called perceptron is the basic architectural structure which makes computational operations. Theoretical ANN studies begun in 1960's however the complex computational operations in ANN's learning step requires advanced computational hardware. This prevents usage of ANN based system until recent decades. Developments in hardware systems made ANN based systems accessible, moreover more complex architectural neural network implementations became possible in recent years.

There are several different ANN architectures focus on different machine learning problems. Basically ANN's are used with Feed-Forward Neural Network structure. In a Feed Forward Neural Network percentrons connected parallelly and serially with each other. Each percentrons in the same level called a layer and in Feed Forward Neural Networks loops are forbidden between layer. There is no memory unit in Feed Forward Neural Networks, no state or data can be transferred between usage sessions of the network. Because of the simple architecture, implementation and learning phase of a Feed Forward Network is quite simple.

On the other hand there is another neural network type called Recurrent Neural Network (RNN). In RNN's output of the network transferred as an additional parameter to the input at the next usage of the network. Also loops are acceptable between inner layers of the neural network. Thus Network can share and transfer state or data between usage sessions of the network. So RNN's perform better results compared with Feed Forward Neural Networks when processing a sequentially related data. However RNN models have computational complexity in both learning and operation phase. This made RNN implementations to request advanced computational hardware systems.

Furthermore in theoretically RNN's can handle and transfer state between usage of networks so they can handle sequentially related data better compared to Feed Forward Neural Networks. However there is a problem called Vanishing Gradients explains that transferring state between short usage sessions of network is acceptable but in long term state transfer between sessions are limited. To solve the Vanishing Gradient problem it is suggested to use a special RNN architecture called Long Short-Term Mermory (LSTM). LSTM contains a seperate memory network unit that responsible to store, update and transfer state between sessions. Thus LSTM can handle transfer an important state information between long running sessions.

Transferring state information in classic RNN or LSTM system is only forward only. In other words RNN or LSTM systems cannot solve dependencies in backward direction in a sequential data because they cannot transfer the state information to previous sessions. Bidirectional networks used in order to solve this problem. Bidirectional LSTM (BLSTM) consist of 2 LSTM layers which one operates in forward direction and other layer operates backward direction. The output of the forward and backward layers generally passed over a Feed Forward Neural Network to find the output. BLSTM structures are performs successful results when sequential data is related both in forward and backward direction.

Last, there is another RNN model that uses stacked BLSTM networks called Deep BLSTM (DBLSTM). In DBLSTM architecture BLSTM layers are serially connected to each other. In recent studies its reported that DBLSTM architecture produces

outstanding performance in contrast to single layer BLSTM networks especially in Natural Language Processing tasks.

Therefore in this thesis a Named Entity Recognition Systems for Turkish Language based on LSTM, BLSTM and DBLSTM architectures are designed and implemented. Both design of the systems and the implementation specifications are reported in detail. Besides in order to measure and compare the performance of the implemented systems a well known Turkish NER Dataset used in the experiment section. As a result we reach an F_1 score of %93.69 in the best system. As far as we observe the best result in literature has %93.59 F_1 score. Our best system result reaches and smoothly improves the best result in literature.

1. GİRİŞ

Doğal Dil İşleme (DDİ) makineler ile insanların doğal diller üzerinden haberleşebilmesi için sistem tasarlamak ve gerçeklemek ile ilgilenen bir araştırma alanıdır [1]. İnsan-Makine etkileşimini doğal dil düzeyinde gerçeklemeyi amaçlayan bu araştırma alanında doğal dilin makineler tarafından analiz edilmesi, anlamlandırılması ve sentezlenmesi önemli çalışma konularıdır. Bilginin analizinde ve anlamlandırılmasında önemli bir yere sahip olan Varlık İsmi Tanıma (VİT) genel anlamda bir metinde geçen varlıkların bulunarak önceden tanımlı kişi, yer ve organizasyon gibi sınıflardan bir tanesine atanması işlemine denilmektedir.

1.1 Varlık İsmi Tanıma

VİT çalışmaları Doğal Dil İşleme, Veri Madenciliği ve Veri Çıkarımı gibi alanlarda önemli bir yere sahiptir. Konu üzerine yapılan ilk çalışmalardan biri Lisa F. Rau (1991) tarafından gerçekleştirilmiş olup yazar çalışmasında problemi metin içerisinde geçen şirket isimlerini işaretleme olarak ele almıştır. Daha sonra yapılan birçok çalışmada işaretlenecek varlık olarak farklı kavramlar kullanılsa da güncel çalışmaların bir çoğu CoNLL 2003 [3] ve MUC-6 [4] konferansları kapsamında ortak yürütülen faaliyetlerdeki tanımlamaları ve ölçüm yöntemlerini referans almaktadır. CoNLL, VİT işlemini genel olarak metin içerisinde geçen ve ENAMEX olarak adlandırılan kişi (Person), organizasyon (Organization) ve yer adı (Location) bilgilerini işaretleme çalışması olarak tanımlanmaktadır. MUC ise problemi biraz daha geniş ölçekte ele almaktadır. MUC, ENAMEX veri tipine ek olarak TIMEX (tarih ve saat) ve NUMEX (sayısal değerler, parasal değerler ve yüzde ifadeleri) değerlerini de problemin içerisinde işaretlenecek veriler olarak değerlendirmektedir. Tüm bunlarla birlikte Varlık İsmi Tanıma ENAMEX, TIMEX ve NUMEX veri tipleri ile sınırlı olmayıp farklı alanlardaki çalışmalarda ilgili alana özgü varlıkların tanınması ve işaretlenmesi için de kullanılmaktadır. Konuyu örneklendirmek gerekirse uygulama alanına bağlı olarak e-posta adresleri, telefon numaraları, kitap başlıkları, proje

isimleri gibi kategoriler de Varlık İsmi olarak ele alınabilmektedir. Benzer şekilde biyoinformatik ve kimya alanlarındaki metinlerde geçen protein isimleri, RNA, DNA, hücre bilgileri, ilaç adları, kimyasal adlarının tespit edilmesi ve işaretlenmesi de Varlık İsmi Tanıma kapsamında değerlendirilmektedir.

1.2 Yapay Sinir Ağları

İnsanlar gibi düşünen, davranan bilgisayarlı sistemlerinin geliştirilmesi uzun yıllardır insanoğlunun en heyecanlı araştırma konularından biridir. Bu konu makine öğrenmesi alanınında ele alınmaktadır. Makine öğrenmesi alanında çözümlenecek probleme göre birçok metodoloji bulunmaktadır. Yapay Sinir Ağları (YSA), insan sinir ağlarını model alınarak tasarlanan bir makine öğrenmesi metodolojisidir. Yapay sinir ağlarının temel yapısı insan sinir hücresi, nöron, model alınarak gerçeklenen yapay nöronlardan oluşmaktadır. YSA'larda nöronlar genellikle tekil olarak kullanılmazlar, problemin çözümüne bağlı olarak nöronlar farklı bağlantı modelleri ile birbiriyle ilişkilendirilerek kullanılırlar. Örneğin görüntü işleme çalışmalarında çoğunlukla Konvolüsyonel Yapay Sinir Ağları (Convolutional Neural Networks - CNN) kullanılırken, Doğal Dil İşleme çalışmalarında metinlerin sözcük dizisi olarak ele alınması ve çözümleme işleminin yapılmasından dolayı çoğunlukla Tekrarlayan Yapay Sinir Ağları (Recurrent Neural Network - RNN) ile başarılı sonuçlar elde edilmektedir.

Karmaşık yapıdaki Yapay Sinir Ağlarının hem gerçeklenmesi hem de eğitim işlemleri gelişmiş donanımların kullanılmasını gerektirir. Son yıllarda teknolojideki donanımsal gelişmeler karmaşık yapılardaki Yapay Sinir Ağlarının gerçeklenmesini imkanlı hale getirmiştir. Karmaşık ve katmanlı yapılarda uygulanan Yapay Sinir Ağları ile yapılan çözümlere Derin Öğrenme Çözümleri, öğrenme işlemine ise Derin Öğrenme adı verilmiştir. Derin öğrenme son yıllarda birçok farklı problemin çözümünde başarılı sonuçlar verdiği çeşitli çalışmalarda gözlemlenmiştir.

1.3 Hipotez

Bu çalışmada Türkçe dili için özelleşmiş bir VİT sistemleri önerilmiş, önerilen modeller farklı yapılandırmalar ile gerçeklenmiş ve başarıları raporlanmıştır. Makine öğrenmesi tekniklerinden faydalanarak tasarlanan VİT sisteminde Tekrarlayan Yapay

Sinir Ağlarının bir çeşidi olan Uzun Kısa-Dönem Hafıza (Long Short-Term Memory) kısaca LSTM yapısı normal, çift yönlü ve katmanlı olmak üzere farklı yapılandırmalar ile ortaya konmuştur.

Çalışma kapsamında daha önce yapılan VİT çalışmaları incelenmiş, kullanılacak yapay sinir ağı modelini besleyecek girdi özellikleri seçimi detaylı bir şekilde çalışılmıştır. Önerilen yapay sinir ağı tabanlı Türkçe VİT sistemi en iyi modelde %93.69 F1 puanı ulaşılmıştır. Bu çalışma kapsamında önerdiğimiz en iyi modelde ulaşılan sonuç, literatürdeki araştırmalardan gözlemlediğimiz kadarıyla en başarı sonuç olarak karşılaştığımız Güngör ve diğ. (2017) tarafından ulaşılan %93.59 değerini az da olsa geçerek literatürdeki en başarılı Türkçe VİT sonucu haline gelmiştir. Bu tez çalışmasının devamı 2'inci bölüm Varlık İsmi Tanıma sistemleri, 3'üncü bölüm Yapay Sinir Ağı sistemleri, 4'üncü bölüm önerilen Türkçe Varlık İsmi Tanıma sisteminin tasarımı, 5'inci bölüm yürütülen deney ve deneyin sonuçları, 6'ıncı bölüm çalışmanın sonuçları ve gelecekte yapılması planlanan ek çalışmalar şeklindedir.

2. VARLIK İSMİ TANIMA

2.1 Tanımı

Varlık İsmi Tanıma (VİT) metin içerisinde tanımlı varlıkların işaretlenmesi işlemidir. VİT kapsamında yapılan çalışmalar MUC-6 (Message Understanding Conference-6) öncesinde Bilgi Çıkarımı (Information Extraction) alanının bir alt araştırma konusu olarak ele alınmaktaydı. MUC konferansları yapısal olmayan haber içeriklerinden savunma alanı ile ilgili aktiviteler hakkında bilgi çıkarımı görevine odaklandığında, araştırmacılar görevin yerine getirilebilmesi için metin içerisinde geçen kişi, organizasyon, yer isimleri, sayısal değerler, tarih ve zaman bilgileri, parasal ifadeler ve yüzdeli ifadelerin tespit edilmesinin önemli olduğunu farketmişlerdir. Gözlemlediğimiz kadarıyla ilk kez MUC-6 konferansı ile birlikte R. Grishman ve Sundheim (1996) metin içerisinde işaretlenecek bu veriler için "Varlık İsmi" (Named Entity) terimini ortaya atmışlardır. Daha sonra yapılan çalışmalarda tanımlanan kategorilerdeki varlık isimlerini tanıma işlemlerine ve bu işlemleri yerine getirecek sistemlere daha sonra Varlık İsmi Tanıma ve Sınıflandırma Sistemleri adı verilmiştir.

VİT ile ilgili çalışmalar 1990'larda başlamış ve 1996 sonrasında hız kazanmıştır (D. Nadeau ve S. Sekine 2007). Birçok kaynakta belirtildiği ve bizim de gözlemlediğimiz kadarıyla bu alanda yapılan ilk araştırmalardan biri Lisa F. Rau (1991) tarafından gerçekleştirilmiş olup yazar çalışmasında problemi metin içerisinde geçen şirket isimlerini işaretleme olarak ele almıştır. Türkçe için ise bilinen ilk çalışma ise 1999 yılında gerçekleştirilmiş olup 2009 sonrasında bu alandaki çalışmalar ilgi görmeye başlamış ve hız kazanmıştır.

2.2 Türkçe'de Varlık İsmi Tanıma Çalışmaları

Türkçe'de VİT çalışmaları incelendiğinde bu konuda karşılaştığımız ilk yayın Cucerzan and Yarowsky (1999) tarafından yapılan çalışmadır. Bu çalışmada yazarlar dilin biçimbilimsel ve bağlamsal özelliklerini istatistiksel yöntemler ile öğrenerek dilden bağımsız bir sistem oraya koymuşlardır. Yazarlar Türkçe, Romence, Yunanca,

Hindice ve İngilizce diller için VİT sistemi ortaya koymuşlar, ortaya koyduları sistem ile Türkçe'de %53.04 F_1 puanına ulaşmışlardır. Türkçe için yapılan ilk kapsamlı çalışma Tür ve diğ. (2003) tarafından gerçekleştirilmiştir. Yazarlar çalışma kapsamında işaretli bir veri kümesi hazırlamışladır. Tür ve diğ. (2003) tarafından hazırlanan veri kümesi daha sonra yapılan birçok VİT çalışmasında referans veri kümesi olarak kullanılmıştır. Yazarlar Saklı Markov Modellerini (HMM) kullanarak geliştirdikleri VİT sistemi ile %91.56 F_1 puanına ulaşmışlardır.

Literatür araştırmalarında gözlemlediğimiz kadarıyla Türkçe VİT çalışmaları 2008 sonrası hız kazanmıştır. Bayraktar ve Temizel (2008) dilbilgisi kurallarından yola çıkarak en çok kullanılan fiiller için kural tabanlı bir sistem oluşturmuşlardır. Kişi isimleri için oluşturulan bu sistemde yazarlar kişi ismi algılamada %81.97 F_1 puanına ulaşmışlardır. Küçük ve Yazıcı (2009) kişi isimleri, tanınmış ünlülerin isimleri, yer ve organizasyon isimlerinden oluşan sözlük destekli kural tabanlı bir sistem oluşturmuşlardır. Çalışma kapsamında oluşturdukları sistem ile güncel metinler için %78.7 F_1 puanına ulaşmışlardır. Yeniterzi (2011) dilin biçimbilimsel özelliklerinden yola çıkarak alışılmışın dışında bir modelleme yöntemi ile veriyi temsil etmeyi denemiş, Koşullu Rassal Ağlar (Conditional Random Fields-CRF) kullanarak %88.9 F_1 puanına ulaşmıştır.

Tatar ve Çiçekli (2011) gözetimli öğrenme (supervised learning) yöntemi ile dilin sözcük, biçimbilim, imla ve bağlam özelliklerine ait VİT kurallarını otomatik olarak öğrenen ve daha sonra öğrenilen kurallar ile kural tabanlı çalışan bir sistem geliştirmişlerdir. Özkaya ve Diri (2011) kural tabanlı sistemleri ve CRF yöntemini birlikte kullanarak oluşturudukları hibrit yöntem ile e-posta içerikleri için VİT sistemi ortaya koymuşlardır. Şeker ve Eryiğit (2012) CRF yöntemi ile dilin sözcüksel ve biçimbilimsel özelliklerinden faydalanarak geliştirdikleri sistem ile %91.94 F_1 puanına ulaşmışlardır. Şeker ve Eryiğit (2017) daha sonra yaptıkları detaylı çalışmada ENAMEX veri tipi için %92.15 F_1 sonucuna ulaşmışlardır.

Karşılaştığımız kadarıyla Türkçe'de VİT alanındaki yapay sinir ağı tabanlı yöntem ilk kez Demir ve Özgür (2014) tarafından uygulanmıştır. Yazarlar yapay sinir ağı kullanarak geliştirdikleri modellerinde %91.85 F_1 puanına ulaşmışlardır. Bu yayında yapmış olduğumuz çalışmaya en yakın çalışmalar Kuru ve diğ. (2016) ile Güngör ve diğ. (2017) tarafından gerçekleştirilmiştir. Kuru ve diğ. (2016) DBLSTM

yapısı ile karakter kodlama kullanarak dilden bağımsız öğeler ile çok dilli bir VİT sistemi geliştirmişlerdir. Yazarlar çalışmalarında Türkçe için %91.30 F_1 puanına ulaşmışlardır. Güngör ve diğ. (2017) ise sözcük vektörleri (Word Vectors/Word Embeddings), karakter kodlama (Character Embedding) ve biçimbilimsel kodlama kullanarak BLSTM ile %93.59 F_1 puanına ulaşmışlardır.

Tez çalışması kapsamında incelenen Türkçe Varlık İsmi Tanıma sistemleri, kullandıkları yöntemler, seçilen etiketler, değerlendirme metodolojileri ve raporlanan başarımları Çizelge 2.1'de detaylı olarak verilmiştir. Çalışmalar farklı veri kümeleri, farklı değerlendirme yöntemleri ve farklı etiket seçimleri ile gerçeklenmesinden dolayı aralarında doğrudan raporlanan başarımlara göre karşılaştırılması doğru yaklaşım olmayacaktır. Her bir çalışmanın literatüre katkısı ayrı ayrı ele alınması gerekmektedir. Ayrıca Çizelge 2.1'de yer alan değerlendirme yöntemleri ile ilgili hesaplama prensipleri Bölüm 2.8'de detaylıca alınmıştır.

Çizelge 2.1 : Türkçe'de Varlık İsmi Tanıma Sistemleri.

Çalışma	Yöntem	Varlıklar	Değerlendirme Yöntemi	Raporlanan Başarım
Cucerzan ve Yarowsky (1999)	İstistiksel	Kişi,Yer İsmi	CoNLL	%53.04
Tür ve diğ. (2003)	Saklı Markov Modelleri (HMM)	ENAMEX	MUC	%91.56
Bayraktar ve Temizel (2008)	Kural Tabanlı	Kişi İsmi	MUC	%81,97
Küçük ve Yazıcı (2009)	Kural Tabanlı	ENAMEX,TIMEX,NUMEX	CoNLL*	%78.7
Yeniterzi (2011)	CRF	ENAMEX	CoNLL	%88.9
Tatar ve Çiçekli (2011)	Hibrit	ENAMEX,TIMEX	MUC	%91.08
Özkaya ve Diri (2011)	Hibrit	ENAMEX		
Şeker ve Eryiğit (2012)	CRF	ENAMEX	CoNLL	%91.94
Şeker ve Eryiğit (2012)	CRF	ENAMEX	MUC	%94.59
Demir and Özgür (2014)	Yapay Sinir Ağları	ENAMEX	CoNLL	%91.85
Kuru ve diğ. (2016)	Yapay Sinir Ağları	ENAMEX	CoNLL	%91.30
Şeker and Eryiğit (2017)	CRF	ENAMEX	CoNLL	%92.15
Şeker and Eryiğit (2017)	CRF	ENAMEX,TIMEX,NUMEX	CoNLL	%92.15
Güngör ve diğ. (2017)	Yapay Sinir Ağları	ENAMEX	CoNLL	%93.59

2.3 VİT Sistemlerinin Özellikleri

2.3.1 Dile bağımlılık

VİT sistemleri dile bağımlı ve dilden bağımsız olarak çeşitli şekillerde ele alınabilmektedir. Dilden bağımsız sistemler herhangi bir dile özgü özellikler içermeden VİT problemini çözmeyi amaçlamaktadır. Bu sistemler özellikle dile bağlı özellikler olan dil bilgisi veya dil modelini kullanmadan çözüme giden sistemlerdir. Bu sistemlerin geliştirilmesindeki temel motivasyon geliştirilen sistemin farklı dillere de kolayca uygulanabilmesi ve Varlık İsmi Tanıma işlemlerini gerçekleştirebilmesidir.

Öte yandan dile bağımlı sistemler, belirli bir dil için özelleştirilerek çalışılmış VİT sistemlerdir. Bu sistemler belirli bir dil için kullanılabilecek tüm özellikleri sistemin girdileri olarak kullanmayı ve başarıyı en üst düzeye taşımayı hedeflemektedir. Bu tip sistemlerdeki ana motivasyon seçilen dil için en üst düzey başarıyı elde etmektir. Geliştirilen sistemin farklı dillere kolayca uyarlanabilmesi genellikle olumlu bir özellik olarak görülmekle birlikte dile bağımlı geliştirilen sistemlerin odaklandığı ana gereksinim değildir.

2.3.2 İçerik sınıfı çeşitliliği

VİT sistemleri genellikle içerik sınıflarından bağımsız olarak geliştirilirler. Fakat sistemin geliştirilmesinde kullanılan sözlükler, eğitim kümeleri veya yöntemlerden dolayı çoğunlukla belirli bir içerik sınıfı için özelleşmiş sistemler haline gelirler. Örneğin Küçük ve Yazıcı (2009) kişi isimleri, tanınmış ünlülerin isimleri, yer ve organizasyon isimleri sözlükleri ile destekledikleri kural tabanlı bir sistem geliştirmişlerdir. Geliştirdikleri sistemi daha sonra tarihi metinler ve çocuk hikayelerine uyguladıklarında konsept değişikliğine bağlı içerik sınıfının değişmesinden dolayı çalışmalarında geliştirdikleri sistemin başarısının ciddi bir şekilde düştüğünü raporlamışlardır. Bu sebeple geliştirilen VİT sistemleri geliştirme esnasında kullanılan veri kümelerine ve veri kümlelerinin içerik sınıflarına dolaylı olarak bağımlı olduğu söylenebilir.

Tüm bunlarla birlikte bazı sistemler başlangıçta belirli bir içerik sınıfındaki varlıkları tanıma görevine özgü geliştirilmektedir. Örneğin ihtiyaca bağlı olarak protein ve hücre bilgileri ile varlıkları işaretleyecek, Biyoinformatik alanı için özelleşmiş bir VİT sistemi geliştirilebilir.

2.3.3 Varlık tipleri

VİT alanında yapılan ilk çalışmalarda problem özel isimlerin tanınması şeklinde ele alınmıştır. Birçok çalışmada ele alınan özel isimler kişi (Person), organizasyon (Organization) ve yer adı (Location) şeklindedir. MUC daha sonra kişi, organizasyon ve yer isimlerini ENAMEX varlık kategorisi olarak tanımlamıştır. CoNLL, kişi, organizasyon ve yer adı varlık tiplerinin dışına düşecek varlıklar için diğer (Miscellaneous) şeklinde bir varlık tipi tanımlamış ve bu varlık tipini konferanslarında ENAMEX varlık kategorisine dahil edilmiştir. MUC, ENAMEX veri tipine ek olarak TIMEX (tarih ve saat) ve NUMEX (sayısal değerler, parasal değerler ve yüzde ifadeleri) değerlerini de problemin içerisinde işaretlenecek varlıklar olarak değerlendirmektedir. CoNLL ve MUC kapsamında değerlendirilen varlık kategorileri ve tipleri Çizelge 2.2'de detaylı olarak verilmiştir.

Çizelge 2.2 : CoNLL ve MUC varlık tipleri.

Varlık Kategorisi	Varlık Tipi	Kısaltma	MUC	CoNLL
ENAMEX	Kişi	PER	X	X
ENAMEX	Organizasyon	ORG	X	X
ENAMEX	Yer Adı	LOC	X	X
ENAMEX	Diğer	MISC		X
NUMEX	Sayı	NUMBER	X	
NUMEX	Para	MONEY	X	
NUMEX	Yüzde	PERC	X	
TIMEX	Tarih	DATE	X	
TIMEX	Saat	TIME	X	

Tüm bunlarla birlikte Varlık İsmi Tanımlama ENAMEX, TIMEX ve NUMEX varlık kategorileri ile sınırlı değildir. Uygulama alanına bağlı olarak e-posta adresleri, telefon numaraları, kitap başlıkları, proje isimleri gibi kategoriler de Varlık İsmi olarak ele alınabilmektedir. Biyoinformatik ve kimya alanlarında protein isimleri, RNA, DNA, hücre bilgileri, ilaç adları, kimyasal adları Varlık İsmi olarak ele alınmaktadır.

Varlık isimleri kimi çalışmalarda hiyerarşik olarak ele alınabilmektedir. Kişi isimleri "tanınmış kişiler" alt kategorisinin altında "Bilim İnsanları", "Politikacılar" gibi daha alt kategorileri ayrılabilmekte, yer adları ise "Şehir", "Ülke" isimleri gibi daha alt kategorilere ayrılabilmektedir. S. Sekine ve Nobata (2004) çalışmalarında ENAMEX varlık kategorisi için 200 civarında hiyerarşik alt varlık tipi tanımlamıştır.

2.4 Kullanım Alanları

Varlık İsmi Tanıma, bilgi çıkarımının alt konusu olmakla birlikte metin sınıflandırmadan arama algoritmalarına, soru-cevap sistemlerinden makine çevirisi sistemlerine farklı birçok DDİ çalışmasında kullanılmaktadır. Kullanım alanları ve kullanım amaçları şu şekilde örneklendirilebilir:

- **Haber Metinlerinin Sınıflandırması:** Her gün yüz binlerce yeni haber içeriğinin yayınlandığı bir dünyada haber içeriklerinin sınıflandırılması, ilgilenilen haber içeriklerinin filtrelenebilmesi önemlidir. Metin sınıflandırma işlemi birçok metin sınıflandırma algoritması ile başarılı bir şekilde gerçekleştirilebilirken sınıflandırma problem örneğin "Savunma sanayi şirketleri ile ilgili teknoloji haberleri" gibi daha spesifik bir konu olduğunda metnin içerisinde geçen varlıkların doğru tespiti sınıflandırıcının başarısını arttıracaktır.

- **Arama algoritmaları:** Arama algoritmaları metinsel dokümanlar arasında aranan anahtar kelimelere göre en anlamlı sonuçları içeren dokümanları bulma görevini yerine getirmeye çalışırlar. Arama işleminden önce doküman içerisinde geçen varlık isimlerinin tespiti ve ilgili dokümana etiket olarak eklenmesi yöntemi arama performansını olumlu yönde etkilemektedir.

- **Makine Çeviri Sistemleri:** Makine çeviri sistemlerinde varlık isimlerinin tanınması ENAMEX varlık tipleri kadar TIMEX ve NUMEX veri tipleri için de önem arz etmektedir. Çeviri sistemlerinde ENAMEX varlık isimleri çoğunlukla değiştirilmeyeceği, TIMEX ve NUMEX ifadelerin de anlamsal çıkarımlarının yapılması çevirinin kalitesini arttıracağı için başarılı bir VİT sistemi dolaylı olarak Makine Çeviri Sisteminin de başarısının artmasına katkıda bulunmaktadır.

Kullanım örneklerini çoğaltmak gerekirse Otomatik Özetleme Sistemleri, Soru Cevap Sistemleri, İçerik Tavsiye Sistemleri gibi sistemler de doğrudan veya dolaylı olarak VİT sistemleri ile etkileşime geçerler. Ayrıca VİT sistemlerinin kullanım alanları yukarıda örneklendirilenlerle sınırlı olmayıp birçok Doğal Dil İşleme görevinde farklı amaçlarla sistemlerin başarısını arttırmak için kullanılmaktadır.

2.5 Varlık İsmi Tanıma Yöntemleri

VİT alanında yapılan çalışmalarda çözülecek problemin yapısına, elde bulunan veri kümesinin içeriğine bağlı olarak kural tabanlı, istatistiksel veya hibrit birçok farklı yöntem kullanılmaktadır. İstatistiksel yöntemler üzerinde yapılan çalışmalar için işaretli veri kümesinin sağlanması temel gereksinimdir. İşaretli veri kümesinin kısıtlı olduğu ilk çalışmalarda araştırmacıların kural tabanlı sistemler üzerinde çalıştığı görülmektedir. İstatistiksel yöntemler için uygun işaretli veri kümelerinin oluşturulması ve literatüre kazandırılması ile birlikte yapılan çalışmalarda makine öğrenmesi yöntemi kullanımlarının hız kazandığı gözlemlenmektedir.

2.5.1 Kural tabanlı sistemler

Kural tabanlı sistemlerde genellikle tanımlanacak varlıklarının listesinin bulunduğu sözlüklerle birlikte elle oluşturulan kurallar uygulanır. S. Sekine ve Nobata (2004) geliştirdikleri kural tabanlı sistem ile 200 civarında hiyerarşik kategorideki varlık ismileri için bir tanıma sistemi geliştirmişlerdir. Türkçe'de Küçük ve Yazıcı (2009) dilbilgisi özelliklerinden faydalanarak tanımlanacak varlıklar için hazırladıkları sözlük yardımı ile Türkçe için kural tabanlı varlık ismi tanıma sistemi geliştirmişlerdir.

Gözlemlediğimiz kadarıyla kural tabanlı sistemler genellikle istatistiksel yöntemler için gerekli öğrenme verisinin olmadığı veya kısıtlı olduğu durumlarda dilin yapısal özelliklerinden faydalanılarak geliştirilmiş olduğu öne çıkmaktadır. Genel olarak dilbilgisi kurallarının yanı sıra bilinen tanımlı varlık listelerinden faydalanan kural tabanlı sistemler VİT problemini uygulandıkları kapsam içerisinde belirli bir başarı ile çalışmaktadırlar, fakat kapsam değiştiğinde sistemin başarısı ciddi bir şekilde durumda etkilenmektedir. Küçük ve Yazıcı (2009) yaptıkları çalışmada güncel metinler için %78.7 F_1 puanına ulaşmış iken geliştirdikleri sistemi herhangi bir değişiklik yapmadan tarihsel metinlere ve çocuk masallarına uyguladıklarında başarının sırasıyla %55

ve %69.3 F_1 puanına düştüğünü raporlamışlardır. Yazarlar sistem başarımında bu düşüşün güncel içerikler için oluşturulmuş ve kullanılan bilinen varlık listelerinin tarihsel metinlerdeki tarihi varlık isimlerini ve çocuk masallarındaki varlık isimlerini kapsamamasından dolayı meydana geldiğini çalışmalarında raporlamışlardır.

2.5.2 İstatistiksel yöntemler

Türkçe'deki VİT çalışmaları incelendiğinde Saklı Markov Modelleri (HMM), CRF ve Yapay Sinir Ağı yöntemleri kullanılarak geliştirilen VİT sistemleri ön plana çıkmaktadır. Özellikle CRF kullanımı ile Şeker and Eryiğit (2017) ve yapay sinir ağları yapısı ile Gungor ve diğ. (2017) çalışmaları sistem başarımı olarak ön plana çıkmaktadır.

2.5.3 Hibrit yöntemler

İngilizce ve diğer dillerdeki VİT çalışmaları incelendiğinde kural tabanlı ve istatistiksel yöntemlerin birlikte kullanıldığı hibrit yöntemler ile geliştirilen sistemler hakkında örneklere rastlanmaktadır. Türkçe'de ise yaptığımız literatür araştırmalarında gözlemlediğimiz kadarıyla bu konudaki tek çalışma Özkaya and Diri (2011) tarafından geliştirilen ve e-posta sistemlerinde kişi isimlerinin tespiti için kullanılan sistemdir.

2.6 Veri Temsil Özellikleri

VİT çalışmalarında içeriğe ait veri çeşitli temsil özellikleri seçilerek kural tabanlı sistemler ya da istatistiksel sistemlere girdi olarak kullanılmaktadır. VİT alanında yapılan çalışmalarda temsil özelliklerini Sözcük Özellikleri, Sözlük Özellikleri ve İçerik Özellikleri olmak üzere 3 kategoride toplayabiliriz.

2.6.1 Sözcük özellikleri

Sözcük özellikleri, sözcüklerin biçimbilim ve yazım özelliklerinden yola çıkılarak elde edilen özellikleri kapsamaktadır. Sözcük özellikleri genel olarak aşağıdaki gibidir.

- **Karakter özellikleri**: Sözcüğün yazımında kullanılan karakterlerin durumlarıdır. "İlk harfi büyük", "Tümü büyük", "Tümü küçük", "Karışık" gibi durumlar kullanılır.

- **Özel karakterler**: Sözcüğün içinde geçen tırnak (’), kesme işareti(-), ve işareti (&) gibi işaretlerdir.
- **Sayısal özellikler**: Sözcüğün rakamlardan oluşması, kesirli veya ondalıklı bir sayıyı ifade etmesi, roma rakamlarından oluşması gibi özellikleri içerir.
- **Biçimbilimsel özellikler**: Sözcüğün kökü ve ekleri ile ilgili bilgileri içerir.
- **Sözcük tipi**: Sözcük tipi (Part-of-speech) bir sözcüğün isim, fiil, sıfat, zamir, zarf gibi tiplerden birinde olma durumu ile ilgili bilgiyi içerir.

2.6.2 Sözlük özellikleri

VİT çalışmalarında işaretlenecek bir sözcüğe karar verilirken ön tanımlı bir sözlüğün karar destek sistemi olarak kullanılması genel olarak uygulanan bir yöntemdir. Sözcüklerin farklı varlık tiplerine denk gelebileceği için bir sözcüğün sözlükte bulunup bulunmaması doğrudan bir sözcüğün tipine karar vermek için yeterli bilgiyi sağlamayabilir.

Örneğin:

"Koç Holding Yönetim Kurulu Başkan Vekili Ali Koç yaptığı açıklamada ..."

cümlesindeki "Koç" sözcüğü ele alındığında cümledeki ilk "Koç" sözcüğü bir organizasyon iken ikinci "Koç" kelimesi bir kişi bilgisini ifade etmektedir.

Kimi zaman ise durum biraz daha karmaşıktır. Örneğin:

"Yoğurt Teknoloji çalışanın üzerine yoğurt döküldü."

cümlesinde ise ilk "Yoğurt" bir organizasyonu belirtirken ikinci yoğurt herhangi bir varlık ismini temsil etmemektedir.

Sözlük kullanımında aşağıdaki genellikle aşağıdaki listeler ile ilgili sözlükler oluşturulmaktadır.

- **Varlık Listeleri**: Şirket, Organizasyon, Kişi, Kıta, Ülke, Şehir, Yer İsimleri vb.
- **Varlık Kanıtları Listeleri**: Tanımı varlı ön ekleri, Unvan bilgileri vb.

2.6.3 İçerik ve derlem özellikleri

İçerik ve derlem özellikleri tüm doküman içerisindeki bilgiler veya dokümanın kendi özelliklerinden yola çıkarak elde edilebilmektedir. Genel VİT sistemlerinde temsil özelliği olarak kullanılan içerik ve derlem özellikleri şu şekilde sınıflandırılabilir.

- **Çoklu tekrar ve büyük-küçük harf kullanımı:** Thielen (1995) , Ravin ve Wacholder (1997) ve Mikheev (1999) çalışmalarında metin içerisinde çoklu tekrar eden ve içerikte farklı büyük-küçük harf yazımlarına rastlanan sözcüklerin cümle başı gibi büyük harfle başlamasının bir varlık ismi olmasından mı yoksa cümle başı olmasından mı kaynaklandığı belirli olmayan sözcüklerin cümle içerisinde geçen durumlardaki yazım özelliklerinden faydalanmışlardır. Ayrıca çeşitli çalışmalarda bir sözcüğün kaç kere büyük harfle veya küçük harfle yazıldığına ait istatistikleri temsil özelliği olarak kullanılmıştır.
- **Doküman özellikleri ve bilgileri:** VİT işleminin uygulanacağı kimi içerikler için dokümanın özellikleri varlık isminin çıkarımı için önem arz edebilmektedir. Örneğin bir e-posta içeriği için e-postanın başlık bilgileri kişi isimleri çıkarımını kolaylaştıracaktır.
- **Çok Sözcüklü Varlık İsimleri:** Derlemlerdeki n-gram istatistiklerinden yola çıkarak çok sözcüklü ifadeler elde edilebilmektedir. Bununla birlikte Da Silva ve diğ. (2004) çok sözcüklü ifadelerin nadiren küçük harflerle yazıldığına, çoğunlukla baş harfleri büyük yazılan ifadeler olduğunu önermişlerdir. Bunun dışında bir ifadenin çok sözcüklü olup olmadığını bilmek VİT sisteminin başarımını etkileyeceği için temsil özelliği olarak yapılan çalışmalarda kullanılmaktadır.

2.7 Etiket Temsil Yöntemleri

Bilgi çıkarımı çalışmalarında cümleleri sözcüklere göre bölümleyerek çözümlemek sıkça kullanılan bir yöntemdir. VİT çalışmalarında ise sözcüklerin Varlık İsmi olarak işaretlenmesinde farklı temsil yöntemleri bulunmaktadır. İşaretleme yöntemi olarak sıklıkla RAW, IOB1, IOB2, IOE2 ve BILOU [24] temsilleri kullanılmaktadır.

2.7.1 RAW temsil

Cümle sözcüklere bölündükten sonra her bir sözcük bağlı olduğu tipe ait etiketi almaktadır. Herhangi bir varlık tipine ait olmayan sözcükler "O" etiketi ile temsil edilir.

2.7.2 IOB1 temsili

IOB1 temsilinde cümle içerisindeki varlık ismi tamlamasındaki sözcüklerin etiketleri I- ön eki ile temsil edilmektedir. Arada varlık ismi dışı sözcük olmaksızın arka arkaya gelen varlık isimlerinde ilk varlığın tüm etiketleri I- ön eki ile temsil edilirken, takip eden varlıkların ilk sözcükleri B- ön eki ile temsil edilir.

2.7.3 IOB2 temsili

IOB2 temsilinde cümle içerisindeki varlık ismi tamlamasındaki ilk sözcüklerin etiketleri B- ön eki ile devam eden sözcüklerin etiketleri ise I- ön eki ile temsil edilmektedir. Varlık tek bir sözcükten oluşması durumunda varlığın etiketi B- ön ekini alır.

2.7.4 IOE1 temsili

IOE1 temsilinde cümle içerisindeki varlık ismi tamlamasındaki sözcüklerin etiketleri I- ön eki ile temsil edilmektedir. Arada varlık ismi dışı sözcük olmaksızın arka arkaya gelen varlık isimlerinde son varlığın tüm etiketleri I- ön eki ile temsil edilirken, son varlıktan önceki varlıkların son sözcükleri E- ön eki ile temsil edilir.

2.7.5 IOE2 temsili

IOB2 temsilinde cümle içerisindeki varlık ismi tamlamasındaki son sözcüklerin etiketleri E- önceki sözcüklerin etiketleri ise I- ön eki ile temsil edilmektedir. Varlık tek bir sözcükten oluşması durumunda varlığın etiketi E- ön ekini alır. Ayrıca varlık ismi olmayan sözcükler O etiketini alır.

2.7.6 BILOU temsili

BILOU temsilinde varlıklar birden fazla sözcükten oluşuyorsa ilk sözcük etiketi B- ön ekini, son sözcüğün etiketi ise L- ön ekini alır. Varlık ismi tamlamasında arada kalan sözcüklerin etiketleri I- ön ekini almaktadır. Tek sözcükten oluşan varlık isimlerinin etiketleri ise U- ön ekini almaktadırlar.

2.8 Başarım Ölçütleri

VİT sistemlerinin başarısı kesinlik (precision) ve hatırlama (recall) değerlerinin geometri ortalaması olan F1 puanı ile ifade edilmektedir. Kesinlik (precision) ve hatırlama (recall) değerleri hesaplanırken bir VİT tespitinin doğru olup olmadığını farklı şekillerde değerlendirilebilmektedir. Kimi yöntemlerde tespit edilen bir varlık isminin sınırlarında yapılan hata göz ardı edilirken, kimi yöntemlerde ise varlık isminin sınırlarının da doğru tespit edilmesi beklenmektedir. Gözlemdiğimiz çalışmalarda genel olarak MUC, CoNLL ve ACE olmak üzere sıkça kullanılan 3 farklı ölçüm ve değerlendirme yöntemi bulunmaktadır.

2.8.1 MUC değerlendirme yöntemi

MUC konferanslarında R. Grishman ve Sundheim (1996) tarafından tanımlanan VİT görevlerinde önerilen sistemlerin değerlendirilmesi için kulklanılan bir değerlendirme yöntemidir. MUC yönteminde değerlendirme iki eksende gerçekleştirilmektedir. Birinci eksende sistemin doğru varlık tipini (TYPE) bulup bulmadığı ölçülür. Doğru varlık tipi tespitinde tespit edilen varlığın sınırları başka bir varlık ile kesişmediği sürece doğru veya hatalı olması gözardı edilir. İkinci eksende ise varlık metninin tespiti (TEXT) ölçümlendirilir. Doğru varlık metni tespitinde varlık tipinin doğru seçilip seçilmediğine bakılmaksızın varlığın sınırlarının doğru tespit edilip edilmediğine bakılır.

Hem TYPE hem de TEXT verileri için üç farklı metrik hesaplanır.

- COR: Doğru tespitlerin sayısı
- ACT: Sistemin tespit ettiği varlık sayısı

- POS: Veri kümesinde gerçekteki varlık sayısı

MUC değerlendirme yöntemi sistemin genel başarısını mikro-ortalama F_1 değeri (MAF) ile ölçmektedir. MAF değeri hesaplaması için tüm varlıklar ve tüm eksenler için elde edilen COR, ACT ve POS değerlerinin toplamı üzerinden MICRO_PRECISION ve MICRO_RECALL değerleri hesaplanır. MICRO_PRECISION ve MICRO_RECALL değerlerinin hesaplanmasına ait formüller sırasıyla (2.1) ve (2.2) eşitliklerinde verilmiştir.

$$\text{MICRO\_PRECISION} = \frac{\sum_{i=1}^{N} \text{COR}_i}{\sum_{i=1}^{N} \text{ACT}_i} \tag{2.1}$$

$$\text{MICRO\_RECALL} = \frac{\sum_{i=1}^{N} \text{COR}_i}{\sum_{i=1}^{N} \text{POS}_i} \tag{2.2}$$

$$\text{MAF} = 2 \times \frac{\text{MICRO\_PRECISION} \times \text{MICRO\_RECALL}}{\text{MICRO\_PRECISION} + \text{MICRO\_RECALL}} \tag{2.3}$$

Daha sonra elde edilen MICRO_PRECISION ve MICRO_RECALL değerlerin harmonik ortalaması üzerinden MAF elde edilir. MAF değerinin hesaplanmasına ait formül (2.3) eşitliğinde verilmiştir.

2.8.2 CoNLL değerlendirme yöntemi

CoNLL değerlendirme yöntemi MUC değerlendirme yönteminden farklı olarak tespit edilen varlıkların hem sınırlarının hem de varlık tiplerinin doğru olup olmadığını değerlendirir. Varlık tipi veya sınırının doğru olmaması durumunda tespiti doğru kabul etmez. Bunun dışında MUC değerlendirme yönteminde olduğu gibi CoNLL değerlendirme yönteminde de veriler için üç farklı metrik hesaplanır.

- COR: Doğru tespitlerin sayısı
- ACT: Sistemin tespit ettiği varlık sayısı
- POS: Veri kümesinde gerçekteki varlık sayısı

CoNLL değerlendirme yöntemi de MUC değerlendirme yöntemi gibi sistemin başarısını mikro-ortalama F_1 değeri (MAF) ile ölçmektedir.

MUC değerlendirme yönteminde olduğu gibi MAF değeri hesaplaması için öncelikle elde edilen COR, ACT ve POS değerlerinin toplamı üzerinden MICRO_PRECISION ve MICRO_RECALL değerleri hesaplanır. Bu hesaplamada MUC değerlendirme yönteminden farklı olan durum, MUC değerlendirme yönteminin TYPE ve TEXT eksenlerini ayrı ayrı ele alıyor olması, CoNLL değerlendirme yönteminin ise bir tespitin doğru olarak ele alınabilmesi için işaretlenen verinin hem TYPE hem de TEXT ekseninde doğru olarak işaretlenmesini beklemesidir. MICRO_PRECISION ve MICRO_RECALL değerlerinin hesaplanmasına ait formüller sırasıyla (2.5) ve (2.5) eşitliklerinde verilmiştir.

$$\text{MICRO\_PRECISION} = \frac{\sum_{i=1}^{N} \text{COR}_i}{\sum_{i=1}^{N} \text{ACT}_i} \tag{2.4}$$

$$\text{MICRO\_RECALL} = \frac{\sum_{i=1}^{N} \text{COR}_i}{\sum_{i=1}^{N} \text{POS}_i} \tag{2.5}$$

$$\text{MAF} = 2 \times \frac{\text{MICRO\_PRECISION} \times \text{MICRO\_RECALL}}{\text{MICRO\_PRECISION} + \text{MICRO\_RECALL}} \tag{2.6}$$

Daha sonra elde edilen MICRO_PRECISION ve MICRO_RECALL değerlerin harmonik ortalaması üzerinden MAF elde edilir. MAF değerinin hesaplanmasına ait formül (2.6) eşitliğinde verilmiştir.

2.8.3 ACE değerlendirme yöntemi

ACE değerlendirme yöntemi MUC ve CoNLL değerlendirme yöntemlerine göre biraz daha karmaşık olduğu söylenebilir. ACE değerlendirme yöntemleri MUC ve CoNLL'da olduğu gibi tüm kategori ve değerlendirme eksenlerindeki doğrulara ve hatalara aynı şekilde yaklaşmak yerine ağırlıklı puanlama sistemi ile sistemin başarısını ölçmeyi hedeflemektedir.

ACE değerlendirme yönteminde genel kullanılan bir puanlama sistemi olmamakla birlikte, uygulanacak puanlama sistemi değerlendirilecek VİT sisteminin optimize edilmek istenen parametrelerine göre oluşturulmaktadır. Örneğin geliştirilecek olan sistemde kişi isimlerinin tespiti organizasyon isimlerinin tespitine göre daha önemli ise her bir kişi isminin tespiti 1 puan iken organizasyon isminin tespiti ise 0.2 gibi bir puan ile değerlendirilebilir.

ACE değerlendirme sisteminde elde edilen puana Varlık Tespit ve Tanıma Değeri (Entity Detection and Recognition Value - EDR) denilir. EDR değeri 2.7'deki gibi hesaplanır.

$$\text{EDR} = \%100 - \text{CEZALAR} \tag{2.7}$$

CEZA ise 2.8'de belirtildiği şekilde hesaplanır.

$$\text{CEZALAR} = \sum \frac{\text{Kategorideki Hatalı Varlık Sayısı} \times \text{Kategorinin Ağırlığı}}{\text{Toplam Varlık Sayısı}} \tag{2.8}$$

ACE özellikle belirli konularda özelleşmesi istenen sistemlerin değerlendirilmesi için başarılı bir ölçüm ve değerlendirme yöntemidir. Bununla birlikte parametrelerin değişken olması farklı çalışmalar arasındaki başarıların EDR değeri üzerinden karşılaştırılamasını olanaksız hale getirmektedir. Ayrıca değerlendirme yönteminin karmaşıklığı sistemin hata analizini zorlaştırmaktadır.

3. YAPAY SİNİR AĞLARI

3.1 Makine Öğrenmesi

Beyin insan bedenindeki en harika organlardan biridir. Görme, işitme, koklama, tatma, dokunma gibi duyu organlarından gelen bilgileri işleyebilmesi ve anlamlandırabilmesinin yanında anılarımızı saklama, duygularımızı ve davranışlarımızı kontrol etme görevlerini yürütür.

İnsan beyni bilgisayarlara göre daha farklı bir yapıda, daha farklı kabiliyetler ile çalışır. Bilgisayarlar aritmetik operasyonların gerçekleştirilmesinde ve algoritmik akışların uygulanmasında oldukça hızlı ve kabiliyetli iken insan beyni ise daha kompleks karar işlemlerinde oldukça başarılıdır. Bilgisayarlar insanlara göre aritmetik işlemleri daha hızlı gerçekleştirebilirler, fakat bir süper bilgisayar için bile kompleks olabilecek problemler çoğu zaman insan beyni için günlük sıradan görevlerdir. Örneğin ses tanıma, yüz tanıma, nesneleri takip etme gibi işlemler bir bilgisayar için kompleks algoritmaların çalıştırılmasını gerektiren komplike işlemler olmakla birlikte bu işlemler yeni doğan bir kaç aylık bir bebek beyni için bile kolayca gerçekleştirilebilen görevlerdir.

İnsanoğlu uzun yıllardır insan gibi davranan robot asistanların hayali kurulmaktadır. İnsan beyni tarafından çözülebilen birçok problemi insanların yerine çözerek evlerimizi temizleyecek, arabalarımızı sürecek, hastalıkları tespit edebilecek sistemlerin geliştirilmesi amacıyla birçok çalışma yürütülmektedir. Fakat geliştirilmesi amaçlanan sistemlerin gerçeklenebilmesi için çözülmesi gereken bir takım problemler bulunmaktadır. Çözülmesi gereken problemlerin çoğu insan beyni için bir kaç mikrosaniyelik problemler iken bilgisayarlar için yüksek komplekslikte hesaplama problemi içermektedir. İnsan beyni için oldukça basit olan bu görevleri yüksek komplekslikte hesaplama problemi içermesini ve gerçekleştirilmesinin zorlu olmasının yanı sıra kimi zaman beynin bu işlemleri nasıl gerçekleştirdiğini algoritmik olarak tanımlayamamaktayız. Aslında işlemlerin tam bir kural kümesine veya algoritmaya dönüştürülememesinden dolayı bu yeni problemin çözümünde klasik algoritmik

programlama mantığı ile ilerlenememektedir. Bu yeni problem "Öğrenme" problemi olarak tanımlanmakta, makine öğrenmesi ve derin öğrenme alanlarının araştırma konusu olarak ele alınmaktadır.

Makine öğrenmesi, problemi doğrudan kural tabanlı bir akış ile çözmeye çalışmak yerine daha önceden deneyimlenen ve kayıt altına alınan veri kümelerinindeki örnekler üzerinden özellikleri istatistiksel yöntemler ile öğrenerek çözümlemeyi hedefler. Bu işlem aslına bakarsanız insan beyni için de benzer şekilde gerçekleşmektedir. İnsan, duyu organlarından gelen sinyalleri beyni ile işleyebilecek kabilyette doğar. Zaman içerisinde ona gösterilen örnekler ile öğrenme işlemini gerçekleştirir. Örneğin gösterilen bir köpek resminin içerisindeki varlığın bir köpek olduğunu tanımak, kişinin daha önceden deneyimlediği örnekler ile ilişkilidir. Öğrenme işlemi örnekler üzerinden gerçekleşir. İki yaşındaki bir çocuğa köpek, şekli, rengi, boyutları ile ilgili bilgiler vererek değil, örneklerle göstererek ve hata yaptığında düzelterek öğretilir.

Makine öğrenmesinde iki önemli unsur bulunmaktadır. Birincisi öğrenme işleminde kullanılacak metodolojidir. Destek Vektör Makinaları (Support Vektör Machines - SVM), Karar Ağaçları (Decision Trees), Şartlı Rastgele Alanlar (Conditional Random Fields - CRF), Yapay Sinir Ağları (Artificial Neural Networks) gibi birçok öğrenme yöntemi bulunmaktadır. İkincisi önemli unsur ise veridir. Öğrenme işlemi mutlaka problem ile ilgili zaman içerisinde oluşturulmuş veya kayıt altına alınmış bir veri kümesi üzerinden gerçekleştirilmektedir.

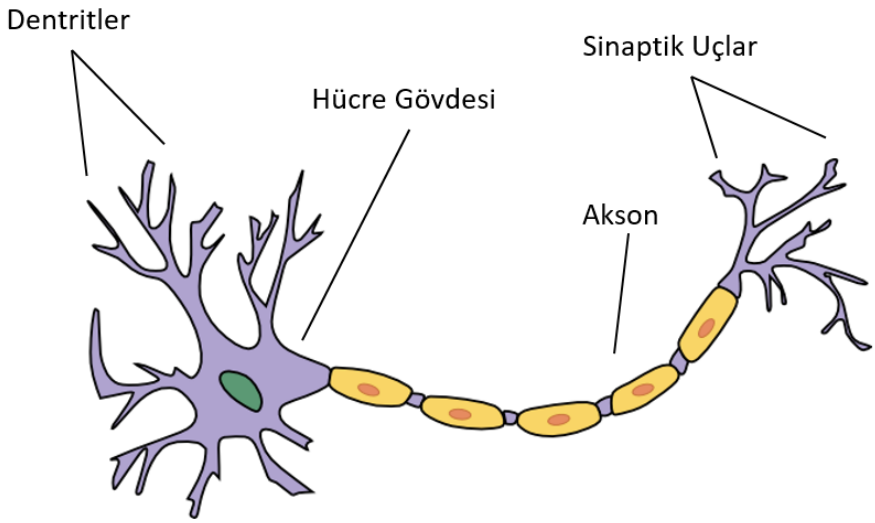
Makine öğrenmesi işlemlerinde iki temel aşama bulunmaktadır. Birincisi öğrenme aşamasıdır. Öğrenme aşamasında sistem daha önceden hazırlanmış veri kümesini kullanarak öğrenme işlemini gerçekleştirir. İkinci aşama ise tahmin aşamasıdır. Tahmin aşamasında sisteme girdi özellikler verilerek sistemin ilgili girdi özelliklere göre sistem çıktısını tahmin etmesi istenir.

Öğrenme işlemlerinde kullanılan metodolojiler temelde gözetimli ve gözetimsiz olmak üzere iki farklı kategoride toplanabilir. Gözetimli öğrenme yöntemlerinde genellikle sınıflandırma görevleri yerine getirilir. Gözetimli öğrenme yönteminde veri kümesinin içerisinde girdi özellikleri ve çıktı etiketleri/sınıfları bulunmaktadır. Öğrenme aşamasında sağlanan veri kümesinden faydalanarak belirli bir girdiye karşılık sistemin çıktı etiketi/sınıfının ne olması gerektiği öğrenilir. Tahmin aşamasında ise sonucu

tahmin edilmek istenen girdi özellikler sisteme verilerek çıktı etiketi/sınıfının sistem tarafından tahmin edilmesi istenir. Gözetimsiz öğrenme yönteminde ise genellikle kümeleme, anomali tespiti gibi görevler yerine getirilir. Gözetimsiz öğrenme yönteminde veri kümesinde sadece girdi özellikleri bulunmaktadır.

Tüm bunlarla birlikte bir makine öğrenmesi sisteminin başarısı kullanılan metodolojinin yanı sıra veri kümesinde seçilen özellikler ile doğrudan ilişkilidir. Makine öğrenmesi sistemlerinin tasarımında Özellik Mühendisliği (Feature Engineering) olarak adlandırılan adım sistemin başarısını doğrudan etkilemektedir. Sistemin girdi ve çıktılarında kullanılacak verilerin seçimi bu adımda titizlikle yürütülmesi gerekmektedir.

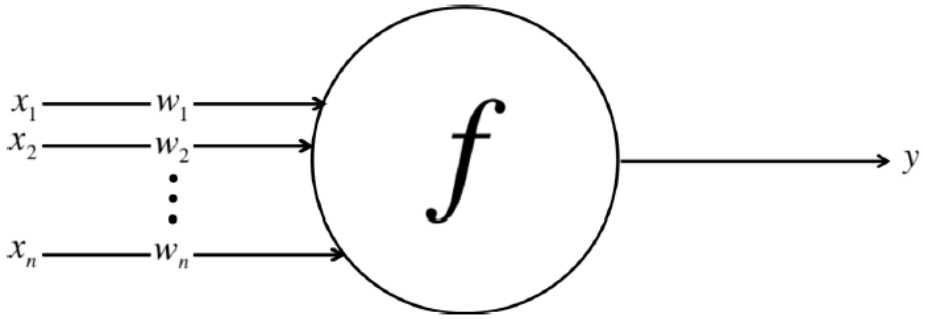
3.2 Yapay Sinir Ağı Yapıları



Şekil 3.1 : Biyolojik nöron hücresi.

Yapay sinir ağları insan beyninin çalışma modeline en yakın makine öğrenmesi metodolojilerinden biridir. İnsan sinir ağları nöron adı verilen sinir hücrelerinden oluşmaktadır (Şekil 3.1). Nöronlar hücre gövdesine bağlı dentritler, akson, sinaptik uçlardan oluşmaktadır. Bir kaç milimetreküp insan beyninde 10.000'in üzerinde nöron hücresi ve her bir hücrede hücreler arası 6.000'in üzerinde sinaps bağlantısı (dentrit-sinaptik uç bağlantısı) bulunmaktadır. Bir nöron hücresi için sinyal girişi dentrit uçlarından olur, dentritlerden gelen sinyal hücre içerisinde dönüştürülerek sinapslar üzerinde diğer hücrelere aktarılır.

Yapay sinir ağları, insan sinir hücrelerinin çalışma yapısı model alınarak geliştirilmiştir. Yapay sinir ağlarında insan sinir ağı yapısındaki nöronlara karşılık yapay olarak matematiksel modeli oluşturulmuş yapay nöronlar/işlem birimleri bulunmaktadır. Yapay nöronların çalışma modeli ilk defa Warren S. McCulloch ve Walter H. Pitts. (1943) tarafından Şekil 3.2'deki gibi modellenmiştir.



Şekil 3.2 : Perceptron matematiksel modeli.

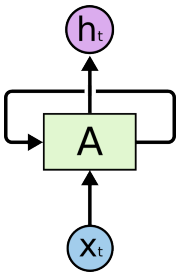
Warren S. McCulloch ve Walter H. Pitts (1943) tarafından geliştirilen modele göre bir nöron n adet x girdisi bulunmaktadır. Bu girdiler $x_1, x_2 \ldots x_n$ şeklinde gösterilmektedir. Her bir girdinin kendisi ile ilişkili w_1, w_2 .... w_n ağırlığı ile çarpılır. Tüm çarpımlar toplandıktan sonra toplama b sabit terimi de eklenerek sonucun elde edildiği, 3.1 eşitliğinde de belirtildiği gibi bir lineer matematiksel işlemdir.

$$y = f(x.w + b) \tag{3.1}$$

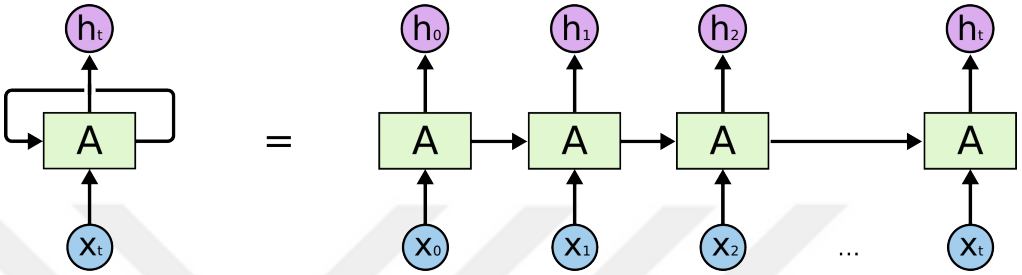
3.3 Tekrarlayan Yapay Sinir Ağları

Tekrarlayan Yapay Sinir Ağları (Recurrent Neural Network) ya da RNN, dizi verisi üzerinde işlem yapmak üzere özelleşmiş bir yapay sinir ağı modelidir. Genellikle dizinin elemanları zamana bağlı girdi ve çıktılar olarak ele alınmaktadır. RNN modellerinin temelleri yapay sinir ağı modelinin parametrelerinin modelin farklı bölümlerinde paylaşılması prensibinden yola çıkmaktadır. Burada önemli olan parametre paylaşımıdır, zira her bir zaman indisindeki veriyi ayrı ayrı hesaplanılması durumunda dizi bir bütün olarak ele alınamayacak ve dizi içerisindeki elemanlar arası ilişkiler çözümlenemeyecektir.

Tekrarlayan Yapay Sinir Ağları (Recurrent Neural Networks - RNN) ilk defa 1980'lerde ortaya atılmıştır. Fakat RNN yapılarının gerçeklenme komplekslikleri nden



Şekil 3.3 : Tekrarlayan yapay sinir ağı (kapalı) Olah (2015).



Şekil 3.4 : Tekrarlayan yapay sinir ağı (açık) Olah (2015).

dolayı bilgisayar donanımlarındaki gelişmelerden sonra popüler ve uygulanabilir hale gelmişledir.

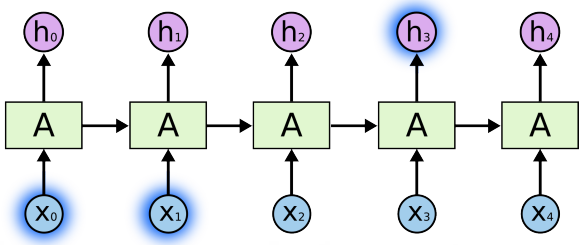
Tekrarlayan Yapay Sinir Ağları, İleri Beslemeli Ağlardan farklı olarak bir tekrarlama katmanı barındırırlar. Barındırdıkları tekrarlama katmanı yapay sinir ağının kullanımları arasında durum bilgisini saklanması ve tekrar kullanılmasını sağlamaktadır. Bu durum en basit anlamda ileri beslemeli bir ağın çıkış bilgisinin saklanarak bir sonraki yapay sinir ağı kullanımında girdi işareti ile birlikte sistem tekrar verilmesi olarak düşünülebilir. Örnek bir tekrarlayan yapay sinir ağı yapısı Şekil 3.3'de verilmiştir. Şekil 3.3'de belirtilen Tekrarlayan Yapay Sinir Ağı yapısının açılmış, İleri Beslemeli ifade edilen hali Şekil 3.4'de verilmiştir.

RNN yapılarında tekrarlama çözülecek probleme bağlı olarak farklı şekillerde modellenebilir. Yapay sinir ağının çıktısının bir sonraki girdi işareti ile birlikte verilebileceği gibi aynı katmanda veya katmanlar arasındaki perceptronlar arasında da tekrarlayan bağlar oluşturulabilir.

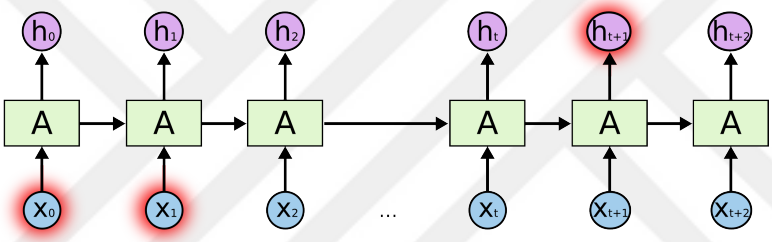
3.3.1 Uzun kısa-dönem hafıza

RNN yapıları teoride dizi işlemlerinde yapay sinir ağının kullanımları arasında durum bilgisini saklamak ve aktarmak işlemlerini yürütse de durum bilgisi yapay sinir ağının

kullanımları arasında tekrar tekrar hesaplanarak aktarılır. Bu durum önemli bir bilginin dönüşüme uğramadan uzun süreler dizi elemanları arasında aktarımına imkan tanımamaktadır. Diğer bir deyiş ile RNN yapılarında dizi içerisindeki kısa süreli bağımlılıklar Şekil 3.5'de görüldüğü gibi başarılı bir şekilde yönetilebilirken Şekil 3.6'de görüldüğü üzere RNN yapıları uzun süreli bağımlılıkların çözümlenmesinde başarım yüksek değildir.

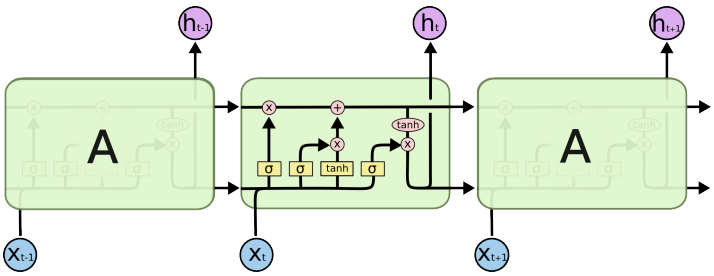


Şekil 3.5 : Kısa süreli bağımlılıklar Olah (2015).



Şekil 3.6 : Uzun süreli bağımlılıklar Olah (2015).

Uzun süreli dizi elemanları arasındaki bağımlılıkların durum bilgisi üzerinden aktarılması probleminin çözümü için Hochreiter ve Schmidhuber (1997) Uzun Kısa-Dönem Hafıza (Long Short-Term Memory - LSTM) mimarisini önermiştir. Bu mimari daha sonra Gers ve diğ. (1999) ve Gers ve diğ. (2002) tarafından geliştirilmiş ve günümüzde yaygın uygulanan modeline ulaşmıştır.



Şekil 3.7 : LSTM yapısı Olah (2015).

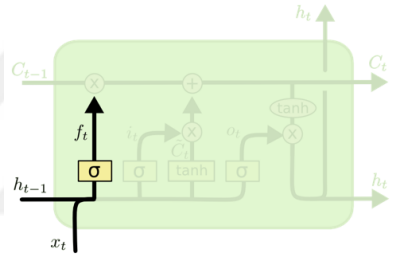
LSTM yapısı Şekil 3.7'de gösterilmiştir. LSTM yapısı RNN ağlarında olduğu gibi h_t çıktısı x_t girdisi bir önceki durum olan h_{t-1} durum bilgisi ile hesaplanır. Klasik RNN yapılarından farklı olarak durum bilgisi hafıza hücresi denilen ayrı bir yapay

sinir ağında saklanır. Hafıza hücresinin görevi durum bilgisini saklamak, her bir adımda belirli bir seviyede unutmak ve yeni durum bilgisini bir önceki durum bilgisiyle birleştirerek saklamaya devam etmektir.

LSTM yapısını adım adım incelemek gerekirse, LSTM yapısı genel olarak 3 ayrı bölümden oluşmaktadır. Bunlar Unutma Kapısı, Girdi Kapısı ve Çıktı Kapısıdır.

Unutma kapısının amacı durum bilgisinin ne kadarının unutulacağı, ne kadarının ise bir sonraki aşamaya aktarılacağına karar vermektir. Bu işlem Şekil 3.8 görülebileceği gibi LSTM yapısının içerisindeki ayrı bir yapay sinir ağı tarafından yönetilmektedir. Unutma kapısına ait yapay sinir ağının matematiksel modeli (3.2) eşitliğinde gösterilmektedir.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{3.2}$$



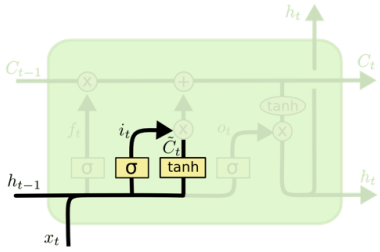
Şekil 3.8 : LSTM unutma kapısı Olah (2015).

İkinci adımda girdi kapısı bölümünde Şekil 3.9'da da görülebileceği gibi t anındaki x_t girdisi ve bir önceki h_{t-1} çıktısına göre bilgisine eklenecek olan bilgiyi hesaplayan ayrı bir yapay sinir ağı yapısı bulunmaktadır. Girdi kapısına ait yapay sinir ağının matematiksel modeli ve durum bilgisine eklenecek olan $\tilde{C}_t$ değerinin hesap formülü (3.3) eşitliğinde gösterilmektedir.

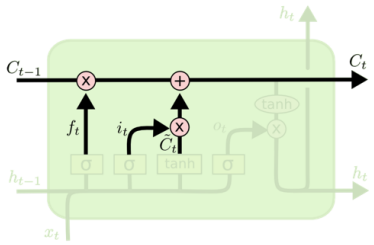
$$\begin{aligned} i_t &= \sigma(W_i.[h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c.[h_{t-1}, x_t] + b_c) \end{aligned} \tag{3.3}$$

Ardından hafıza hücresinin yeni durum bilgisi (3.4) eşitliğinde belirtildiği gibi hesaplanır. Bu işlem ile ilgili yapı Şekil 3.10'de gösterilmektedir.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.4}$$



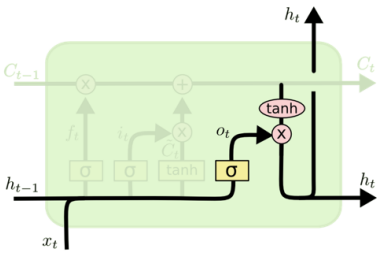
Şekil 3.9 : LSTM girdi kapısı Olah (2015).



Şekil 3.10 : LSTM durum bilgisinin değişimi ve aktarımı Olah (2015).

Son olarak çıktı kapısında sistemin t anındaki h_t çıktısı hesaplanır. Bu işlem Şekil 3.11'de belirtildiği gibi ayrı bir yapay sinir ağı yapısı tarafından gerçekleştirilir. İşlemin matematiksel modeli (3.5) eşitliğinde gösterilmektedir.

$$\begin{aligned} o_t &= \sigma(W_o.[h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{3.5}$$

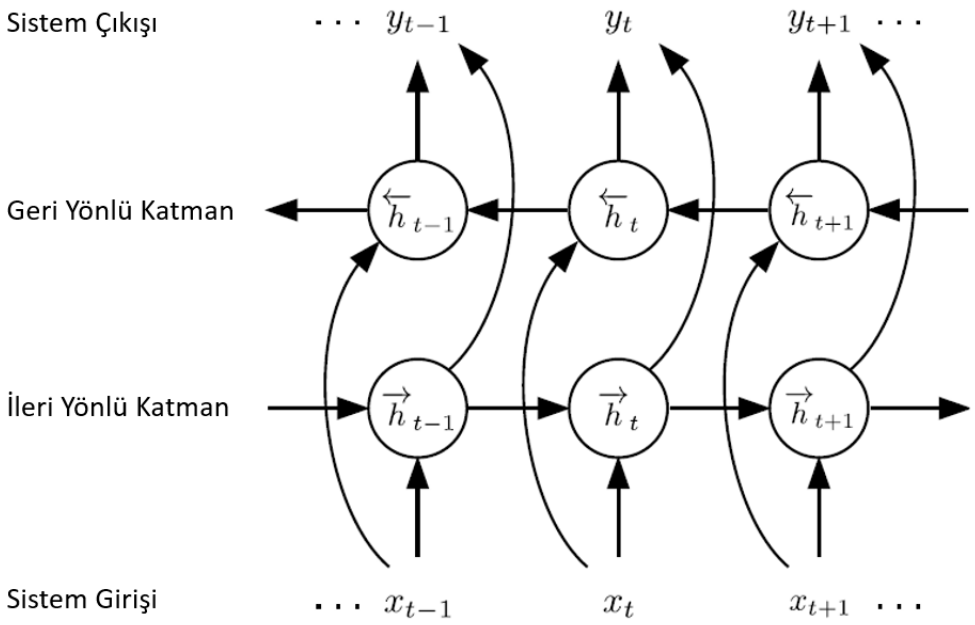


Şekil 3.11 : LSTM çıktı kapısı Olah (2015).

3.3.2 Çift yönlü uzun kısa-dönem hafıza

LSTM yapay sinir ağı yapıları dizi üzerindeki bağımlılıkları zamanda ilerleyen bir şekilde öğrenebilmektedir. Diğer bir deyiş ile t anındaki h_t çıktısı, t ve t'den önceki girdilere bağlı olarak değer almaktadır. t anındaki sistem çıktısının t'den sonraki girdiler ile bağımlılığının olması durumunda RNN yapıları ve LSTM yapıları bu bağımlılıkların çözümlenmesini gerçekleştirememektedir. Doğal dil işleme çalışmalarında ise cümlede içerisinde bir sözcüğün görevine kimi zaman daha sonraki sözcüklerin bilgilerine göre karar vermeniz gerekebilmektedir. Bu t anında h_t çıktısının

t anından sonraki girdiler ile de ilişkilendirilmesi gerektiği anlamına gelmektedir. Bu tip problemlerin çözümünde Çift Yönlü LSTM yapıları kullanılmaktadır. Çift yönlü kullanılan LSTM (Bidirectional Long Short-Term Memory – BLSTM) temel olarak iki adet LSTM yapısının veriyi normal sırada ve ters sırada işleme alması ve sonuçları birleştirmesi prensibi ile çalışmaktadır.



Şekil 3.12 : Çift yönlü uzun kısa-dönem hafıza yapısı.

Şekil 3.12'de gösterildiği gibi y_t sistem çıktısı ileri yönlü LSTM çıktısı $\overrightarrow{h_t}$ ve geri yönlü LSTM çıktısı $\overleftarrow{h_t}$ işaretlerinin birleşimidir.

$$y_t = \overrightarrow{h_t} + \overleftarrow{h_t} \tag{3.6}$$

Çift Yönlü Uzun Kısa-Dönem Hafıza (BLSTM) yapıları dizinin tüm elemanları arasındaki ilişkilerin çözümlenmesinde başarılı sonuçlar vermesi sebebiyle tasarlanan VİT sisteminde kullanılmıştır.

3.3.3 Derin çift yönlü uzun kısa-dönem hafıza

Derin öğrenme ve derin yapay sinir ağlarının başarılı sonuçlar verdiği son yıllarda farklı çalışmalarda raporlanmaktadır. Daha önce yapılan VİT çalışmaları incelendiğinde Kuru ve diğ. (2016) yaptıkları çalışmada Derin Çift Yönlü Uzun Kısa-Dönem Hafıza (DBLSTM) ile sistem başarısının arttığını raporlamışlardır.

DBLSTM yapıları BLSTM yapılarının katmanlı bir şekilde birbirlerine seri bağlanması ile elde edilmektedir.

Kullanılan BLSTM katmanı arttıkça sistem girdi dizisi elemanları arasındaki bağımlılıkları daha başarılı bir şekilde çözümleyebilmektedir. DBLSTM yapılarının başarısı Bölüm 5'de konu ile ilgili deney sonuçlarında ele alınmıştır. DBLSTM yapılarının başarısını bu bölümde de kısaca örneklendirmek gerekirse tez çalışması kapsamında yapılan deneylerde, temel veri kümesi üzerinde tek katmanlı bir BLSTM yapısı ile çift katmanlı bir BLSTM yapısı arasında %12,31 F_1 puanı artışı bulunmaktadır. Bununla birlikte BLSTM katman sayısının artması başarıyı olumlu etkilemesiyle beraber sistemin karmaşıklığını ve donanım ihtiyacını arttırmaktadır.

4. TÜRKÇE VARLIK İSMİ TANIMA SİSTEMİ

Bu bölümde tez çalışması kapsamında önerilen Türkçe Varlık İsmi Tanıma sisteminin amacı, özellik seçimleri ve önerilen yapay sinir ağı modelleri ile ilgili bilgiler verilmiştir.

4.1 Amaç

Tez çalışması kapsamında Türkçe metin içerisinde bulunan Kişi, Organizasyon ve Yer isimlerini tespit ederek yüksek başarım ile işaretleme işlemi gerçekleştirecek yapay sinir ağı temelli bir "Türkçe Varlık İsmi Tanıma" sistemi modellenmesi ve geliştirilmesi amaçlanmış, bu konudaki gerekli çalışmalar yürütülmüştür. Çalışmamız başlangıçta sayı, para, yüzde ifadeleri, tarih ve saat varlık tiplerini de içermek üzere tasarlanmış olmakla birlikte bu varlık tiplerinin kural tabanlı sistemler ile daha hızlı ve performanslı bir şekilde tespit edilebileceği öngörülerek ilgili varlık tiplerinin tespiti ileriki çalışmalara bırakılmış, çalışmaya sadece kişi, organizasyon ve yer isimlerinin tespiti dahil edilmiştir.

4.2 Özellik Seçimi

Makine öğrenmesi sistemlerinde kullanılan öğrenme yöntemleri ve algoritmaların yanı sıra verinin tanımlanmasında seçilecek özellikler ve özelliklerin temsil şekilleri ortaya konulan sistemin başarısına doğrudan etkisi bulunmaktadır. Geliştirilen Türkçe Varlık İsmi Tanıma sistemi için sözcük vektörleri, karakter tabanlı sözcük vektörleri, biçimbilimsel özellikler, yazımsal özellikler, bilinen kişi, organizasyon ve yer isimleri sözlükleri girdi özellikleri olarak seçilmiştir.

4.2.1 Sözcük vektörleri

Sözcüklerin temsili Doğal Dil İşleme çalışmalarının temelini oluşturmaktadır. Birçok çalışmada sözcük temsili işleminde bir-kodlanmış (one-hot) vektör kullanılmaktadır. Bir-kodlanmış vektör oluşturma işlemi genel olarak temsil edilmesi plananan tüm

sözcüklerden bir sözlük oluşturulması ile başlar. Bir-kodlanmış vektör kullanımında V sözlükteki sözcük sayısı (4.1); w_i, i indisindeki sözcük olmak üzere w_i sözcüğünü temsil edecek bir x vektörü (4.2)'deki gibi tanımlanmaktadır.

$$|x| = V \tag{4.1}$$

$$\begin{aligned} \forall k! = i; x_k &= 0 \\ k = i; x_k &= 1 \end{aligned} \tag{4.2}$$

Örneğin "Kırmızı", "Yeşil", "Mavi", "Elma", "Armut", "Portakal" sözcüklerinden oluşan bir sözlükte sözcüklerin olası bir-kodlanmış vektör temsili (4.3)'de gösterilmiştir.

$$\begin{aligned} w_0 &= \text{"Kırmızı"} \quad & w_0 &= [1,0,0,0,0,0] \\ w_1 &= \text{"Yeşil"} \quad & w_1 &= [0,1,0,0,0,0] \\ w_2 &= \text{"Mavi"} \quad & w_2 &= [0,0,1,0,0,0] \\ w_3 &= \text{"Elma"} \quad & w_3 &= [0,0,0,1,0,0] \\ w_4 &= \text{"Armut"} \quad & w_4 &= [0,0,0,0,1,0] \\ w_5 &= \text{"Portakal"} \quad & w_5 &= [0,0,0,0,0,1] \end{aligned} \tag{4.3}$$

Bir-kodlanmış vektör temsilinde sözcük vektörleri arasındaki uzaklık d, eşitlik (4.4)'de belirtildiği gibi öklid uzaklığı ile ölçülebilir.

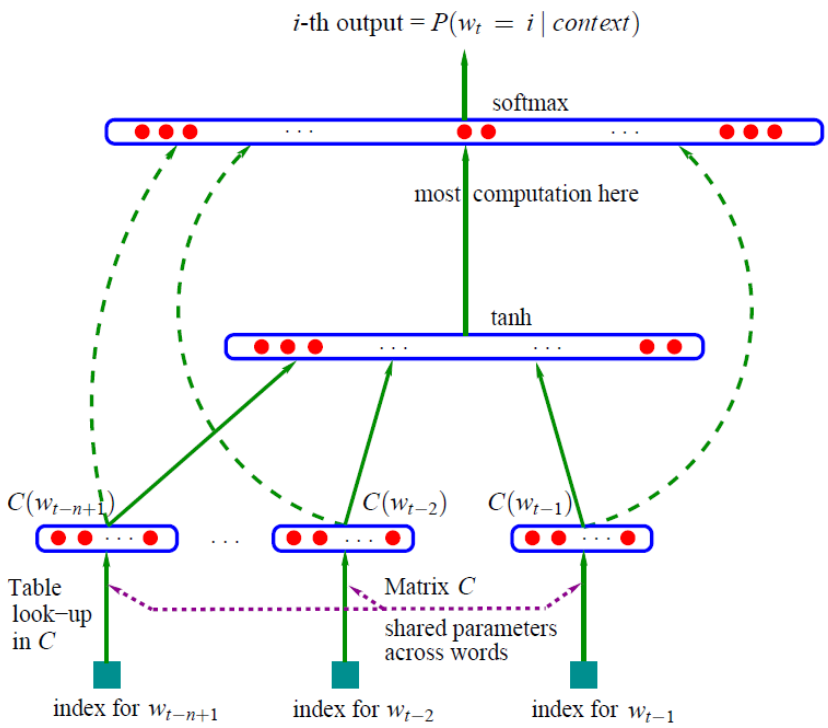
$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4.4}$$

Örnek (4.3) temsilinde sözcükler arası uzaklıklar incelendiğinde "Kırmızı", "Yeşil", "Mavi" birer renk, "Elma", "Armut", "Portakal"ın birer meyve olması gibi anlamsal özelliklerden bağımsız olarak tüm sözcükler vektörlerinin öklid uzaklıkları aynı olduğu gözlemlenebilir. Bu durum VİT çalışmalarında iki tane istenmeyen sonuç doğurmaktadır. Birincisi anlamsal olarak birbirine yakın ve uzak olan sözcüklerin uzaklıklarının aynı olması metnin anlamsal temsilinin oluşturulmasını zorlaştırmaktadır. İkincisi sözlükteki sözcük sayısının artması, temsil vektörünün boyutunun büyümesine neden olmakta, bu gerçeklenecek makine öğrenmesi sistemini boyutunu büyütmekte ve sistemin karmaşılığını arttırmaktadır. Ayrıca sözcük sayısının

artması veri kümesinin içerisinde sözcüklerin rastlanma olasılıklarını düşürmekte dolaylı olarak seyrek veri problemi oluşturmaktadır. Örneğin 100.000 sözcükten oluşan bir sözlükten üretilen bir içerikte 10 sözcüğün ark arkaya rastlanması potansiyel olarak $100.000^{10} = 10^{50}$ şekilde gerçekleşebilir.

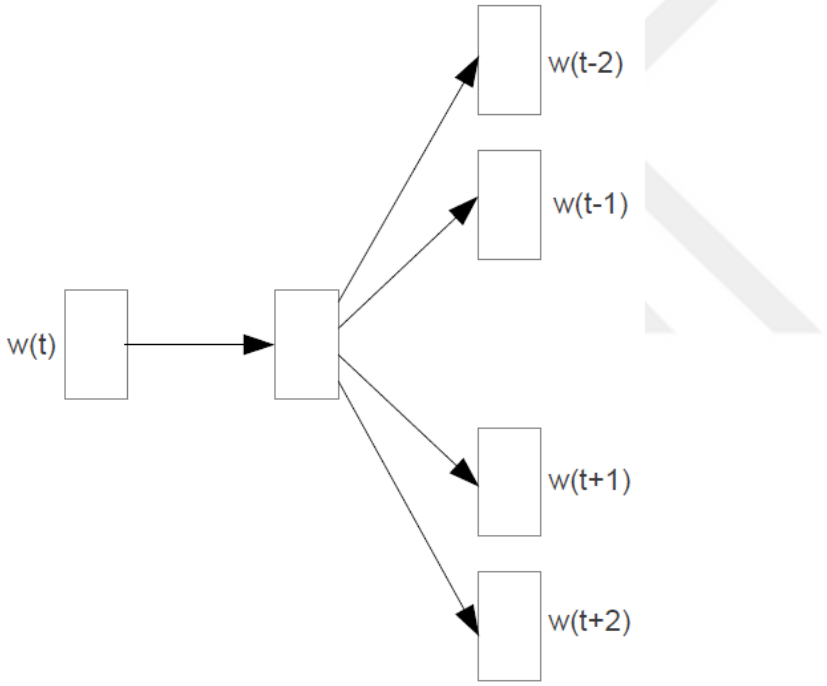
Sözcükleri anlamsal bütünlükleri doğrultusunda temsil edebilmek ve seyrek veri problemini aşmak amacıyla ilk defa Bengio ve diğ. (2003) tarafından bir sözcüğün temsilinin n boyutlu sürekli uzayda bir vektör olarak oluşturulması fikri ortaya atılmıştır. Bengio ve diğ. (2003) özellikle seyrek veri problemini çözümlendirmek ve sözcükleri anlamsal bütünlükleri doğrultusunda temsil edebilmek amacıyla geliştirdikleri yöntemde 3 ana konuya odaklanmışlardır:

1. Sözlükteki her bir sözcüğü reel sayılardan oluşan bir sözcük özellik vektörü ile temsil etmek
2. Ortak olasılık yoğunluk fonksiyonunu sözcük vektörünün özellikleri ile ifade etmek
3. Eş zamanlı olarak sözcük özellik vektörlerini ve olasılık dağılım fonksiyonunu parametrelerini öğrenmek



Şekil 4.1 : Bengio ve diğ. (2003) tarafından önerilen sözcük vektörü oluşturma yapay sinir ağı modeli.

Önermiş oldukları yöntemde her bir sözcüğü çok boyutlu sürekli uzayda bir nokta olarak ele almışlar, sözcüğün uzaydaki yeri sözcüğün vektörü ile ifade edilmektedir. Uzaydaki her bir boyut sözcüğün özelliği olmak üzere 30, 60, 100 boyutlu uzaylarda çalışmalarını yürütmüşlerdir. Önerilen işlem yapay sinir ağı ile gerçekleştirilen bir gözetimsiz öğrenme (unsupervised learning) işlemidir. Sözcük vektörlerinin oluşturulması işlemi genel olarak farklı kategorideki metinlerden dengeli bir şekilde oluşturulmuş büyük bir derlemin vektör oluşturma için tasarlanmış bir yapay sinir ağına öğretilmesi gerçekleştirilmektedir. Öğrenme işlemi sonuncunda elde edilen ağırlık matrisi sözcük vektörlerini içermektedir. Bengio ve diğ. (2003) tarafından önerilen sözcük vektörü oluşturma görevini yürüten yapay sinir ağı modeli Şekil 4.1'de belirtilmiştir.



Şekil 4.2 : Mikolov ve diğ. (2013) çalışmasına ait Skip-gram yapay sinir ağı modeli.

Bengio ve diğ (2003)'ten sonra birçok çalışmada konu ile ilgili çeşitli yöntemler önerilmiştir. Collobert and Weston (2008), Turian ve diğ. (2010), Mikolov ve diğ. (2013) yapay sinir ağı modelinde değişiklikler yaparak sözcük vektörlerinin anlamsal özellik temsilini iyileştirmeye çalışmışlardır. Özellikle Mikolov ve diğ. (2013) yaptıkları çalışmada daha önceki çalışmalardan farklı olarak lineer olmayan saklı katmanı çıkartarak, CBOW ve Skip-gram olarak iki yeni log-lineer yapay sinir ağı modeli önermişlerdir. Önermiş oldukları yöntem öğrenme karmaşılığını azaltarak öğrenme işleminin daha hızlı gerçekleşmesini sağlamıştır. Ayrıca önermiş oldukları

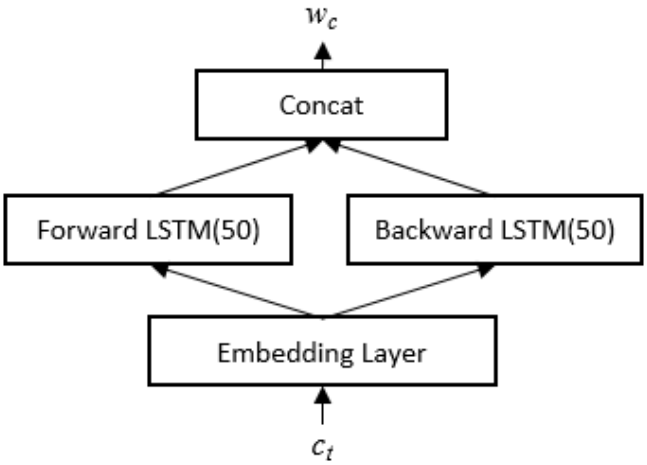
Skip-gram yöntemi daha önce yapılan çalışmalara göre anlamsal ve sözdizimsel olarak daha başarılı sözcük vektörü temsili üretmiştir.

Bu çalışmada anlamsal ve sözdizimsel olarak başarılı sonuçlar üretmesi sebebiyle sözcük temsilinde sözcük vektörleri yöntemi kullanılmıştır. Sözcük vektörlerinin oluşturulmasında ise yapay sinir ağı modeli Şekil 4.2'de gösterilmiş olan Mikolov ve diğ. (2013) tarafından önerilen Skip-gram modeli uygulanmıştır.

4.2.2 Karaketer tabanlı sözcük vektörleri

VİT sistemlerinde karakter tabanlı sözcük vektörlerinin kullanımının sistemin genel başarısını arttırdığını kanıtlayan çeşitli çalışmalar bulunmaktadır. Türkçe'de Kuru ve diğ. (2016) ile Güngör ve diğ. (2017) tarafından yapılan çalışmalarda karakter tabanlı sözcük vektörlerinin başarımı arttırdığı raporlanmaktadır.

Temel prensipte sözcüğün içerisindeki tüm karakterlerin vektörlerinin bir yapay sinir ağından geçirilerek sözcüğün nihai vektörünün hesaplanması ile oluşturulur. Karakter tabanlı sözcük vektörleri bölüm 4.2.1'de belirtilen sözcük vektörlerinden farklı olarak sözcüklerin anlamsal bilgisinin temsili yerine dilin karakter tabanlı bir yapay sinir ağı modelini oluşturmayı amaçlar.



Şekil 4.3 : Karakter tabanlı sözcük vektörü yapay sinir ağı.

Çalışma kapsamında karakter tabanlı sözcük vektörü oluşturma işlemi için Şekil 4.3'de belirtilen yapay sinir ağı modeli tasarlanmıştır. Tasarlanan modelde c_t bir-kodlanmış karakter vektörü, w_c karakter tabanlı sözcük vektörü olmak üzere, w_c karakter tabanlı sözcük vektörü oluşturma süreci şu şekildedir:

1. Sözcüğün karakterlerini temsil eden bir-kodlanmış karakter vektörleri c_t yerleştirme katmanından (Embedding Layer) geçirilir.

2. Yerleştirme katmanı her bir karakter için bir karakter vektörü üretir.

3. Elde edilen karakter vektörleri Çift Yönlü Uzun Kısa-Dönem Hafıza katmanından geçirilerek işlemin sonunda elde edilen sistem çıktısı w_c karakter tabanlı sözcük vektörü olarak ele alınır.

4.2.3 Biçimbilimsel özellikler

Türkçe Ural-Altay dil ailesinden gelen bitişken, hayli fazla yapım ve çekim eklerine sahip bir dildir. Türkçe'de sözcükler kök ve köke eklenen birçok yapım veya çekim ekinden oluşabilmektedir. Türkçe, sahip olduğu yapım ve çekim ekleri itibaryle bir kök sözcükten milyonlarca sözcük oluşturabilen üretken bir dildir (Oflazer 2003, Hankamer 1989). Türkçe gibi üretken dillerde üretken olmayan dillere oranla metin içerisinde daha fazla sayıda farklı sözcük yüzey formuna rastlanır.

Sözcükler temsil ettikleri anlam veya kullanım alanları itibariyle türlere ayrılmaktadırlar. Türkçe'de isim, fiil, sıfat, zamir, zarf, bağlaç, edat ve ünlem gibi sözcük türleri bulunmaktadır. Her sözcük türünün görevi ve kullanım alanları birbirinden farklı olmakla birlikte tümcenin oluşturulmasında farklı işlevleri yerine getirmektedirler.

Türkçe'de bir sözcük yapım ve çekim ekleri alabilmektedir. Yapım eki bir sözcüğü başka bir anlama çevirerek yeni sözcükler üretilmesini sağlamaktadır. Kimi zaman yapım ekleri sözcüğü bir türden başka bir türe de dönüştürebilmektedir. Çekim ekleri ise sözcüğün durum, zaman, iyelik, şahıs, tekillik-çoğulluk gibi özelliklerini ifade etmektedir.

Doğal Dil İşleme çalışmalarında biçimbilim sözcüğün türü ve ekleri ile ilgilenmektedir. Sözcüğün türü, aldığı yapım ve çekim ekleri sözcüğün biçimbilimsel özellikleridir. Biçimbilimsel özelliklerinin analiz edilmesine biçimbilimsel analiz denir. Analiz işlemi biçimbilimsel analiz aracı ile yapılır.

arabamız: araba+Noun+A3sg+P1pl+Nom (4.5)

Örneğin (4.5)'de "arabamız" sözcüğünün biçimbilimsel analiz sonuçları yer almaktadır. Analizde sözcüğün kökünün "araba" olduğu, türünün isim (+Noun) olduğu, tekil (A3sg), 3'üncü çoğul şahıs (+P1pl), yalın halde (+Nom) olduğunu belirtmektedir.

Yapım ekleri ile ise sözcüğe yeni anlamlar yüklemekte, kimi zaman ise sözcük tür dönüşümü yapmaktadırlar. Örneğin (4.6)'da belirtilen "sağlamlaştırdığımızda" sözcüğü "sağlam" kökünden gelmektedir. Sözcüğün türü başta sıfattır, daha sonra aldığı eklerle fiil ve isim türlerine dönüşmüştür. Sözcüğün en son isim türünde olması sözcüğü bir cümle içerisinde kullandığınız zaman isim olarak davranmasına neden olacaktır. Ayrıca cümle içerisinde kullanıldığı yerde sözcük son yapım ekinden sonra aldığı çekim eklerinin karakteristiklerini gösterecektir.

Yapım ekleri biçimbilimsel analizde "^DB" ifadesi ile gösterilir.

$$\begin{aligned} \text{sağlamlaştırdığımızda: sağlam+Adj\^{}DB} & \\ \text{+Verb+Become\^{}DB} & \\ \text{+Verb+Caus+Pos\^{}DB} & \\ \text{+Noun+PastPart+A3sg+Pnon+Loc} & \end{aligned} \tag{4.6}$$

Tüm bunlarla birlikte Türkçe gibi hayli fazla yapım ve çekim ekinin olduğu biçimbilimsel yönden zengin olan bir dilde biçimbilimsel analiz bir sözcük için her zaman tek bir çözüm üretmemektedir. Sözcük üretiminde yer alan kökleri ve eklerin benzerliklerinden dolayı bazı durumlarda bir sözcük için birden fazla çözüm üretilebilmektedir. Örneğin (4.7)'de "elması" sözcüğünün olası farklı biçimbilimsel çözümlemeleri gösterilmiştir.

$$\begin{aligned} \text{elması: elmas+Noun+NAdj+A3sg+Pnon+Acc} & \\ \text{elmas+Noun+NAdj+A3sg+P3sg+Nom} & \\ \text{elma+Noun+A3sg+P3sg+Nom} & \end{aligned} \tag{4.7}$$

Çözümlemeye göre "elması" sözcüğü için olası şu çözümlemeleri bulunmaktadır.

- "elmas" kökünde türen bir isim olup tekil, 3. tekil şahıs, yalın halde olması
- "elmas" kökünde türen bir isim olup tekil, şahıs eki almamış, -i halinde olması

- "elma" kökünde türen bir isim olup tekil, 3. tekil şahıs, yalın halde olması

Bir sözcüğün birden fazla biçimbilimsel çözümü olduğu durumlarda içerikten faydalanılarak biçimbilimsel belirsizlik giderme işlemi yapılır. Genel olarak biçimbilimsel belirsizlik giderme işlemi komşu sözcüklerin bilgilerini ele alarak yapılan bir işlemdir.

Çalışmamızda veri kümesinin biçimbilimsel çözümleme ve belirsizlik giderme işlemleri yapıldıktan sonra her sözcük için biçimbilimsel bir çözüm elde edilmiştir. Biçimbilimsel temsil özellikleri seçilirken sözcüğün yapım eki alıp almamış olması durumu göz önünde bulundurulmuştur. Temsil özellikleri seçiminde:

- Eğer sözcük yapım eki almışsa son yapım ekinden sonraki sözcük türü ve çekim ekleri temsil özelliklerine eklenmiştir
- Eğer sözcük yapım eki almamışsa elde edilen sözcüğün türü ve çekim ekleri temsil özelliklerine eklenmiştir.

Sözcüğün yapım eki alması durumunda son yapım ekinden önceki ekler sözcüğün durum, zaman, iyelik, şahıs, tekillik-çoğulluk gibi özelliklerini temsil edemeyeceği için göz ardı edilmiştir.

4.2.4 Yazımsal özellikler

Metin içerisindeki imla bilgileri ve sözcüklerin yazımsal özellikleri bilgi çıkarımı algoritmalarının çalışmasında destekleyici bilgiler sağlamaktadır. Örneğin Türkçe'de ve birçok dilde özel varlıkların ilk harfleri büyük yazılmaktadır. Her ilk harfi büyük olan sözcük için bir varlık ismi olduğu söylenemeyecek olsa da bir sözcüğün ilk harfinin büyük olması VİT işlemi için önemli bir bilgi içermektedir. Yine Türkçe'de özel varlık isimlerinden sonra gelen ekler tırnak ayracı (') kullanılarak yazılmaktadır, bu sebeple sözcüğün içerisinde geçen tırnak işareti yine sözcük hakkında bilgi vermektedir.

Yapılan çalışmada geliştirilecek makine öğrenmesi sistemi için anlamsal bilgi çıkarımı açısından önemli olabilecek yazımsal özellikler özellik kümesine dahil edilmiştir. Yazımsal özelliklerden aşağıdakiler girdi kümesi için seçilmiştir:

- **Tümü Büyük:** Sözcüğün tüm harflerinin olma durumu
- **Tümü Küçük:** Sözcüğün tüm harflerinin küçük olma durumu
- **İlk Harfi Büyük:** Sözcüğün ilk harfinin büyük olma durumu
- **Ayraç İçermesi:** Sözcüğün içerisinde tırnak ayracı (') bulunması durumu
- **Nokta İçermesi:** Sözcüğün içerisinde nokta (.) bulunması durumu
- **Sayısal Olması:** Sözcüğün tüm alfanumerik karakterlerinin rakam olması durumu

4.2.5 Tanımlı varlık sözlük özellikleri

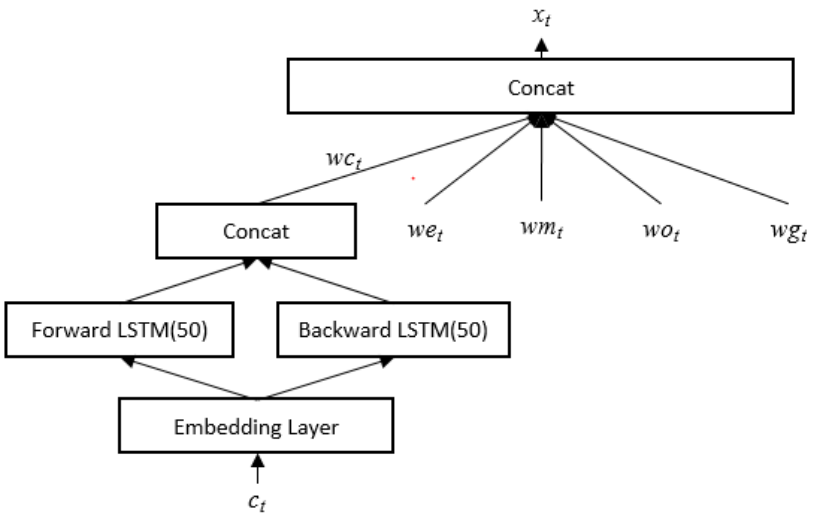
Çalışma kapsamında tanımış varlık isimlerinden oluşan referans sözlükler oluşturulmuştur. Tanınacak kişi, organizasyon ve yer adı kategorileri için ayrı ayrı sözlükler oluşturularak veri kümesinin içerisindeki tüm sözcüklerin ilgili sözlüklerde bulunup bulunmama durumu yapay sinir ağı yapısına bir girdi özelliği olarak eklenmiştir. Sistemde tasarlanan tanımlı varlık sözlük özellikleri aşağıdaki gibidir:

- Kişi sözlüğünde bulunma durumu
- Organizasyon isimleri sözlüğünde bulunma durumu
- Yer adları sözlüğünde bulunma durumu

4.2.6 Temsilin oluşturulması

Çalışma kapsamında önerilen sistemde sözcük girdi kümesinde sözcük vektörü, sözcüğün karakter tabanlı sözcük vektörü, biçimbilimsel özellikleri, yazımsal özellikleri ve tanımlı varlık sözlük özellikleri ile temsil edilmiştir.

Seçilen tüm özelliklerden temsil oluşturma işlemi seçilen özelliklerin vektörel olarak birleştirilmesi ile gerçekleştirilir. Girdi kümesindeki tek bir girdiyi ifade eden x_t girdisi Şekil 4.4'de belirtildiği gibi oluşturulur. Şekil 4.4'de c_k, sözcüğün karakter vektörleri dizisini; wc_t, karakter tabanlı sözcük vektörünü; we_t, Skip-gram sözcük vektörlerini; wm_t, sözcüğün biçimbilimsel özelliklerini; wo_t, sözcüğün yazımsal özelliklerini; wg_t ise sözcüğün sözlük özelliklerini belirtmektedir.



Şekil 4.4 : Girdi temsilinin oluşturulması.

4.3 Sınıf ve Etiket Seçimi

Çalışma kapsamında ENAMEX varlık kategorisindeki kişi, organizasyon ve yer adı sınıfları seçilmiştir. Veri etiketleme işlemleri Bölüm 2.7'de belirtilen etiket temsil yöntemlerinden "RAW Temsil" yöntemi kullanılmıştır. "RAW Temsil" yöntemi kapsamında seçilen varlık sınıfları ve etiketler Çizelge 4.1'de belirtilmiştir.

Çizelge 4.1 : Seçilen sınıflar ve etiketler.

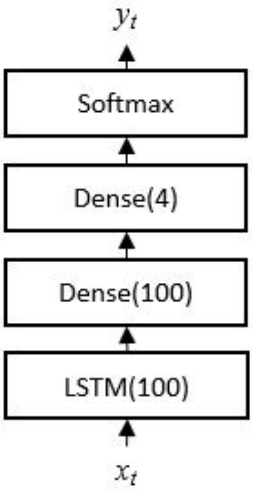
Varlık Sınıfı	Etiket
Kişi	PER
Organizasyon	ORG
Yer Adı	LOC

4.4 Yapay Sinir Ağı Modelleri

Çalışma kapsamında 3 farklı yapıda yapay sinir ağı modeli farklı konfigrasyon parametreleri ile uygulanmış ve başarımları karşılaştırılmıştır. Bölüm 5'te detaylı olarak konfigrasyon parametreleri ve deney sonuçları sunulan yapay sinir ağları şu şekildedir:

4.4.1 Uzun kısa-dönem hafıza modeli

Cümleyi bir sözcük dizisi olarak ele alarak yapay sinir ağında dizi operasyonlarındaki başarılarından dolayı RNN (Tekrarlayan Yapay Sinir Ağı) yapısı kullanılmasına karar verilmiştir. Çalışma kapsamında önerilen modelde temel kurgu klasik RNN

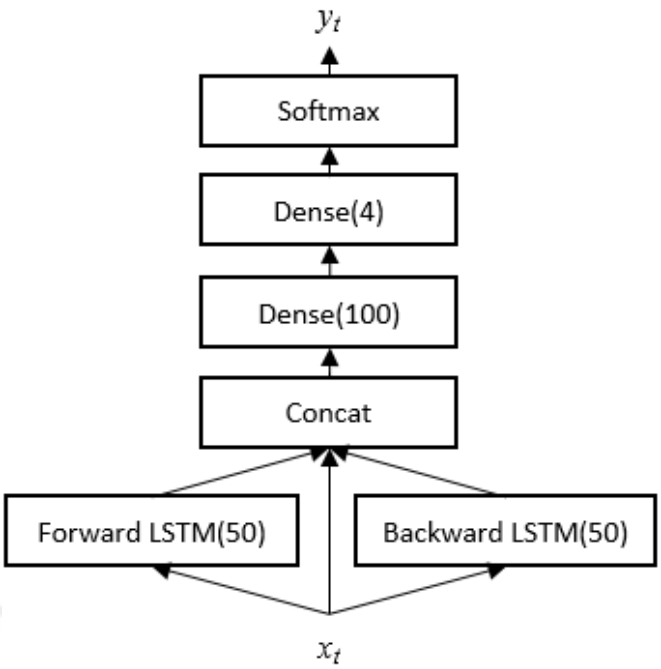


Şekil 4.5 : Uzun kısa-dönem hafıza modeli.

yapılarının uzun dizi işlemlerinde geçmiş indislere ait verilerin anlamlandırılması konusundaki zaafiyetini ortadan kaldırmak için bu konuda daha başarılı olan Uzun Kısa-Dönem Hafıza (LSTM) yapısı üzerine odaklanılmıştır. Temel modelde şekil 4.5'de gösterildiği gibi tek katmanlı ileri yönlü LSTM yapısı çıkışına uygulanan sırasıyla 100 ve 4 algılayıcıdan (perceptron) oluşan katmanlı bir yapay sinir ağı modeli gerçeklenmiştir. 100 algılayıcıdan oluşan yapay sinir katmanında çıkışında RELU aktivasyonu kullanılmıştır. Sistem çıkışında ise olasılık dağılımlarını normalize etmek için sonuçlar softmax katmanından geçirilmiştir.

4.4.2 Çift yönlü uzun kısa-dönem hafıza modeli

LSTM yapay sinir ağı yapıları tek yönlü çalışmaktadır. Diğer bir deyiş ile LSTM ilerleyen bir dizi üzerinde işlem yaparken dizinin başlangıcından sonuna doğru tek yönlü ilerlemektedir. Tek yönlü ilerlemenin getirdiği dezavantajları ortadan kaldırmak için Çift Yönlü Uzun Kısa-Dönem Hafıza (BLSTM) yapısı kullanılmıştır. Çalışma kapsamında önerilen model BLSTM olarak olarak ileri (forward) ve geri (backward) yönlü çalışan iki adet LSTM yapısından oluşmaktadır. BLSTM oluştulurken kullanılacak LSTM yapılarının iç katman genişliği 50 olarak belirlenmiştir. Şekil 4.6'de de görülebileceği gibi ileri ve geri yönlü çalışan iki LSTM yapay sinir ağı yapısının çıkışı birleştirilmiş daha sonra sırasıyla 100 ve 4 algılayıcıdan (perceptron) oluşan katmanlı bir yapay sinir ağı modeli gerçeklenmiştir. 100 algılayıcıdan oluşan yapay sinir katmanında çıkışında RELU aktivasyonu kullanılmıştır. Sistem



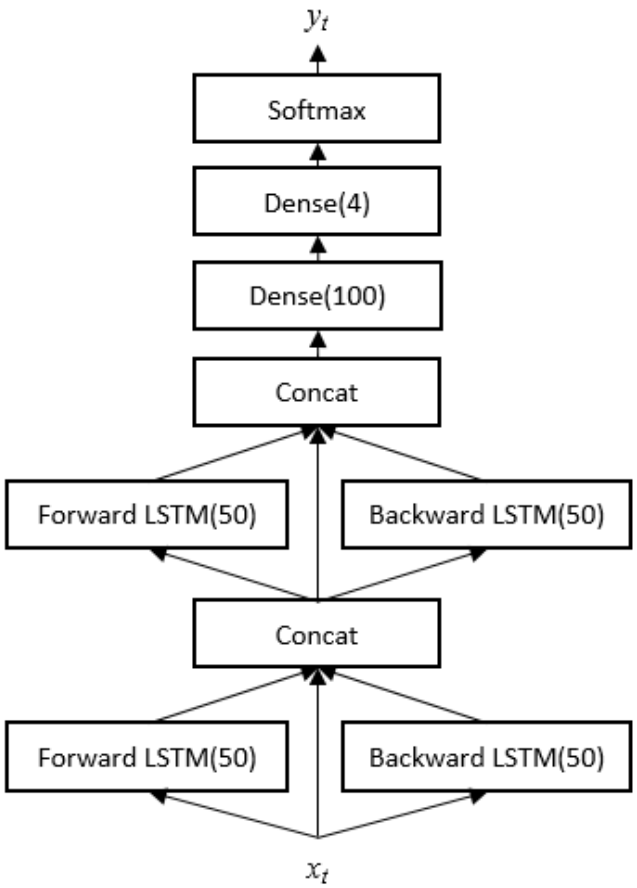
Şekil 4.6 : Çift yönlü uzun kısa-dönem hafıza modeli.

çıkışında ise olasılık dağılımlarını normalize etmek için sonuçlar softmax katmanından geçirilmiştir.

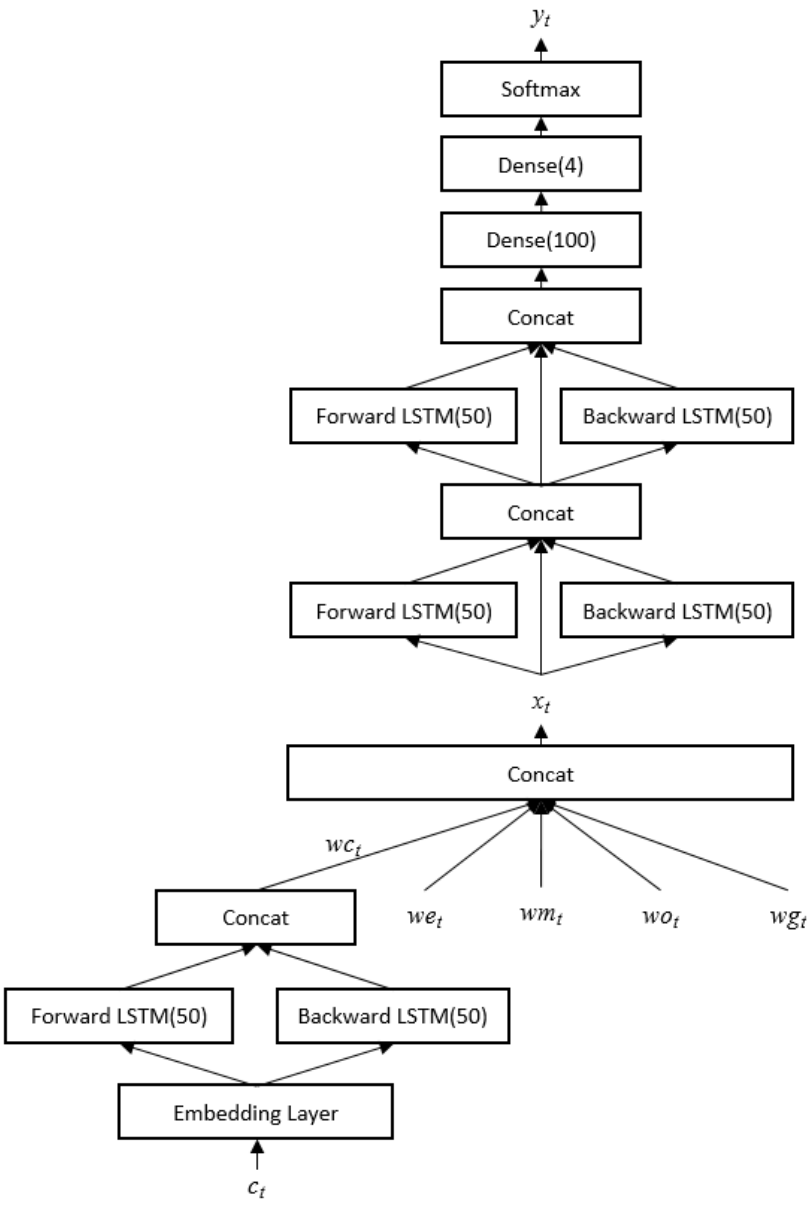
4.4.3 Derin çift yönlü uzun kısa-dönem hafıza modeli

Çalışma kapsamında farklı sayılarda katmana sahip BLSTM modelleri uygulanmıştır. Arka arkaya bağlanan BLSTM katmanları arasında bir önceki katmanın girdisi bir sonraki katmana da iletilecek şekilde bir veri yolu mimarisi oluşturulmuştur. Önerilen DBLSTM modelinde her bir BLSTM katmanın girişi bir önceki katmanın çıktısı ve girdisinin birleştirilmiş verisinin içermektedir. BLSTM katmanlarının çıkışına ise sırayla 100 ve 4 algılayıcıdan (perceptron) oluşan katmanlı bir yapay sinir ağı modeli gerçeklenmiştir. 100 algılayıcıdan oluşan yapay sinir katmanında çıkışında RELU aktivasyonu kullanılmıştır. Sistem çıkışında ise olasılık dağılımlarını normalize etmek için sonuçlar softmax katmanından geçirilmiştir.

Çalışma kapsamında önerilen iki katmanlı çift yönlü uzun kısa-dönem hafıza modeli Şekil 4.7'de belirtilmiştir. Şekil 4.8'de ise seçilen özelliklerin de birleştirilmesi ile birlikte sistemin detaylı tam diyagramı detaylı verilmiştir.



Şekil 4.7 : İki katmanlı çift yönlü uzun kısa-dönem hafıza modeli.



Şekil 4.8 : İki katmanlı çift yönlü uzun kısa-dönem hafıza modeli (tam diyagram).

5. DENEY ve DENEYSEL SONUÇLAR

Tez çalışması kapsamında Bölüm 4'de önerilen yapay sinir ağı modeli gerçeklenmiş, tavsiye edilen özellik seçimi uygulanarak veri kümesi oluşturulmuştur. Çalışmada tavsiye edilen yapay sinir ağı modelleri ve özellik seçimleri kontrollü deneylerle farklı kombinasyonlarda ele alınarak modeldeki ve özelliklerindeki değişikliklerin sistemin başarımına etkisi gözlemlenmiştir. Bu bölümde seçilen veri kümesi, veri kümesinin normalizasyon ve dönüşüm işlemleri, önerilen yapay sinir ağı modellerinin parametre seçimleri, yapay sinir ağı modeli ve deney yazılımının gerçeklenmesi, yürütülen deneyler ve deneylerin sonuçları bulunmaktadır.

5.1 Veri Kümesi

Çalışma kapsamında Tür ve diğ. (2003) çalışmasında derlenmiş olan veri kümesinin Şeker ve Eryiğit (2017) tarafından ENAMEX, TIMEX ve NUMEX varlık kategorilerini kapsayacak şekilde 7 varlık tipi için (kişi, organizasyon, yer isimleri, tarih, saat, parasal değerler ve yüzde ifadeleri) için işaretlenmiş versiyonu kullanılmıştır. İlgili veri kümesinin yapay sinir ağı için uygun hale getirilmesi amacıyla bir takım ön işlemler ve dönüşüm işlemleri gerçekleştirilmiştir. Özellik seçiminde belirtilen sözcük vektörleri, karakter tabanlı sözcük vektörleri, biçimbilimsel özellikler ve yazımsal özellikler için ön işlemler yapılmış ve yapay sinir ağı modeline uygun girdi verileri oluşturulmuştur.

Sözcük vektörlerinin oluşturulmasında Mikolov ve diğ. (2013) tarafından önerilen Skip-gram yöntemi kullanılmıştır. Yöntemin uygulanmasında ihtiyaç duyulan derlem için Sak ve diğ. (2008) tarafından oluşturulan Türkçe Haber Derlemi kullanılmıştır. Sözcük vektörleri oluşturma işleminin sonucunda 547K sözcükten oluşan bir sözcük vektörleri sözlüğü üretilmiştir. Sözcük vektörleri oluşturulmasına ilişkin detaylı çalışma 5.1.1'de ele alınmıştır.

Skip-gram yöntemi ile oluşturulan sözcük vektörlerinin yanı sıra seçilen özellikler arasında bulunan karakter tabanlı sözcük vektörleri de elde edilmiştir. Bu konuda yapılan çalışma ile ilgili detaylar 5.1.2'de ele alınmıştır.

Biçimbilimsel özelliklerin tespiti için sırasıyla biçimbilimsel çözümleme ve biçimbilimsel belirsizlik giderme işlemleri yapılarak her bir sözcüğün kök, sözcük türü, yapım ve çekim ekleri ayrıştırılmıştır. Bu konuda yapılan çalışma ile ilgili detaylar 5.1.3'de ele alınmıştır.

Tüm bunlarla birlikte yazımsal özellikler ile ilgili ön işlemler yapılarak veri kümesine dahil edilmiştir. İlgili çalışmalar 5.1.4'de belirtilmiştir.

Son olarak kişi, organizasyon ve yer adlarından oluşan bilinen varlık isimlerine ait kontrol listeleri oluşturulmuştur. Listelerin toplandığı kaynak ve toplanma yöntemi ile ilgili detaylar 5.1.5'de belirtilmiştir.

Son olarak Şeker ve Eryiğit (2017) tarafından ENAMEX, NUMEX ve TIMEX varlık kategorileri işaretlenerek hazırlanmış veri kümesinden sadece kişi, organizasyon ve yer ismi etiketlerinden oluşan ENAMEX kategorisindeki varlık etiketleri seçilmiş, diğer etiketler ise kaldırılarak diğer kategorilerdeki sözcükler etikesiz olarak ele alınmıştır. Etiketleri oluşturulması işlemi Bölüm 5.2'de detaylı bir şekilde ele alınmıştır.

5.1.1 Sözcük vektörlerinin oluşturulması

Sözcük vektörleri oluşturma yapay sinir ağı ile gerçekleştirilen bir gözetimsiz öğrenme (unsupervised learning) işlemidir. Bu işlemin gerçekleştirilmesi için sözcük vektörlerinin öğrenileceği geniş kapsamlı bir derlem ve işlemi gerçekleştirecek bir yapay sinir ağı modeli gereklidir. Çalışma kapsamındaki sözcük vektörleri oluşturma işlemi Bölüm 4.2.1'de detaylı bir şekilde belirtildiği gibi Mikolov ve diğ. tarafından önerilen Skip-gram yöntemi ile oluşturulmuştur. Sözcük vektörlerinin oluşturulması işleminde Skip-gram modelini gerçekleyen açık kaynaklı word2vec[1] aracı kullanılmıştır. Geniş kapsamlı Türkçe derlem olarak Sak ve diğ. (2008) tarafından derlenen, 184M sözcükten oluşan, Türkçe Haber Derlemi kullanılmıştır. Kullanılan derlem, sözcük vektörleri oluşturulmadan önce bir dizi normalizasyon

[1] https://github.com/tmikolov/word2vec

işleminden geçirilmiştir. Sak ve diğ. (2008) derleminin sözcük vektörleri oluşturulması esnasındaki kullanımına ait istatistikler Çizelge 5.1'de verilmiştir.

Çizelge 5.1 : Sözcük vektörlerinin oluşturulmasında kullanılan derlemin istatistikleri.

Sözcük Sayısı	184M
Token Sayısı	212M
Sözlük Boyutu	547K

Normalizasyon çalışması kapsamında derlemin içerisindeki geçen noktalama işaretlerinin histogramı çıkartılmış ve seyrek veri problemine sebep olabilecek noktalama işaretleri düzenlenmiştir. Örneğin "Türkiye'de" ve "Türkiye‘de" örneğinde olduğu gibi farklı karakterledeki kesme işaretleri birleştirilmiştir. Bununla birlikte içeriğin tamamı küçük harfe çevrildikten sonra sözcük vektörü hesaplama işlemi gerçekleştirilmiştir.

5.1.2 Karakter tabanlı sözcük vektörlerinin oluşturulması

Karakter tabanlı sözcük vektörlerinin oluşturulması işlemi yapay sinir ağı modeli Bölüm 4.2.2'de detaylı bir şekilde ele alınmıştır. Karakter tabanlı sözcük vektörleri oluşturacak yapay sinir ağı modelinin gerçeklenmesi Keras kütüphanesi ile gerçekleştirilmiştir. Embedding layer gerçeklemesi Keras Embedding Layer ile oluşturulmuştur.

5.1.3 Biçimbilimsel analiz işlemleri

Türkçe Varlık İsmi Tanıma Veri kümesindeki sözcüklerde Oflazer (1994) tarafından geliştirilen 2 seviyeli biçimbilimsel çözümleyici kullanılarak biçimbilimsel çözümlemesi yapılmıştır. Toplam 469.596 sözcükten oluşan veri kümesinde 167.430 sözcüğün tek bir biçimbilimsel çözümü bulunmaktadır, 95.031 sözcük için ise bir biçimbilimsel çözümü bulunamamıştır. Geriye kalan 207.135 sözcük için birden fazla olası biçimbilimsel çözüm bulunmaktadır. Biçimbilimsel işleminde elde edilen sonuçlar Çizelge 5.2'de paylaşılmıştır.

Biçimbilimsel çözümleme sonucu birden fazla çözüme sahip sözcükler için Sak ve diğ. (2008) tarafından geliştirilen biçimbilimsel belirsizlik giderici kullanılarak yapılan çözümlemenin belirsizlik giderme işlemleri gerçekleştirilmiştir. Belirsizlik giderme işlemleri sonrasında her bir sözcük için bir biçimbilimsel çözüm atanmıştır.

Çizelge 5.2 : Biçimbilimsel analiz istatistikleri.

1 biçimbilimsel çözümü olan sözcük sayısı	167.430
2 biçimbilimsel çözümü olan sözcük sayısı	123.269
3 biçimbilimsel çözümü olan sözcük sayısı	33.628
4 biçimbilimsel çözümü olan sözcük sayısı	36.261
5 ve 5'ten fazla biçimbilimsel çözümü olan sözcük sayısı	13.977
Biçimbilimsel çözümü bulunamayan sözcük sayısı	95.031
Toplam sözcük sayısı	469.596

Biçimbilimsel analiz işlemleri sonucunda Bölüm 4.2.3'de detaylı bir şekilde anlatıldığı gibi son yapım ekinden sonraki sözcük türü ve çekim ekleri veri kümesine dahil edilmiştir. Veri kümesine dahil edilen sözcük türleri ile ilgili istatistiksel bilgi Çizelge 5.3'de verilmiştir.

Çizelge 5.3 : Sözcük türlerine göre biçimbilimsel analiz sonucu dağılımı.

Sözcük Türü	Etiketi	Sayısı
Sıfat	Adj	48.936
Zarf	Adverb	17.500
Bağlaç	Conj	18.162
Belirteç	Det	16.113
	Dup	68
Ünlem	Interj	154
İsim	Noun	214.600
Sayı	Num	3.882
	Postp	10.498
Zamir	Pron	6.905
Soru	Ques	981
Fiil	Verb	36.766
Bilinmiyor	?	95.031
Toplam		469.596

5.1.4 Yazımsal özeliklerin analizi

Varlık ismi tanıma işlemlerinde kullanılan veri kümesi ön işlemden geçirilerek veri kümesinin içerisindeki her bir sözcüğün için yazımsal özellikleri çıkarılmıştır. Sözcüğün ilgili yazımsal özelliğe sahip olması durumu 1, sahip olmaması durumu 0 olarak veri kümesine eklenmiştir. Yazımsal özelliklere göre sözcük sayıları Çizelge 5.4'de belirtilmiştir.

Çizelge 5.4 : Yazımsal özelliklere göre sözcük sayıları.

Yazımsal Özellik	Sözcük Sayısı
Tümü Büyük	6.649
Tümü Küçük	314.531
İlk Harfi Büyük	90.973
Ayraç İçermesi	51.034
Nokta İçermesi	9.852
Sayısal Olması	7.572

5.1.5 Varlık sözlüklerinin oluşturulması ve sözlük özeliklerin analizi.

Veri kümesindeki sözlük özelliklerin analiz işlemi için öncelikle kişi, organizasyon ve yer isimlerinden oluşan varlık sözlükleri oluşturulmuştur. Sözlüklerin oluşturma işleminde Şahin ve diğ. (2017) çalışmasında oluşturulan işaretli "Coarse Grain" veri kümesi kullanılmıştır. Her bir varlık tipi için veri kümesinin içerisinde işaretlenmiş varlıklar çıkartılarak sözlükler elde edilmiştir.

Tüm bunlarla birlikte yapılan ilk incelemede Şahin ve diğ. (2017) çalışmasında otomatik olarak oluşturulmuş bu işaretli veri kümesinde bazı hatalar tespit edilmiştir. Oluşturulan sözlüklere hatalı verilerin aktarımını engellemek adına elde edilen sözlük verisinin içerisindeki baş harfleri büyük olmayan sözcük içeren veriler ile 2 karakterden küçük olan veriler çıkartılarak sözlükler optimize edilmiştir.

Sözlük oluşturma işlemleri sonunda elde edilen sözlükler ve bu sözlüklerin içerisindeki varlıkların sayıları Çizelge 5.5'de verilmiştir.

Çizelge 5.5 : Sözlüklerdeki sözcük sayıları.

Sözlük tipi	Sözcük Sayısı
Kişi isimleri sözlüğü	62.833
Organizasyon isimleri sözlüğü	11.764
Yer isimleri sözlüğü	27.030

Sözlüklerin elde edilmesinin ardından Varlık İsmi Tanıma işlemlerinde kullanılacak veri kümesindeki her bir ğün sözlüklerde bulunup bulunmama durumu için GPER, GORG ve GLOC şeklinde 3 ayrı girdi özelliği hesaplanmıştır. Bu özellikler:

- **GPER:** Sözcüğün kişi sözlüğünde bulunması durumunda 1, bulunmaması durumunda 0

- **GLOC:** Sözcüğün kişi sözlüğünde bulunması durumunda 1, bulunmaması durumunda 0
- **GORG:** Sözcüğün kişi sözlüğünde bulunması durumunda 1, bulunmaması durumunda 0

şeklindedir.

5.2 Temsilin Oluşturulması

VİT sınıf etiketleri olarak ENAMEX veri tipi için tanımlanmış olan kişi (PER), organizasyon (ORG) ve yer ismi (LOC) etiketleri kullanılmıştır. Bu verilerin dışında kalanlar ise diğer (O) olarak sınıflandırılmıştır. İşaretlemede RAW etiket formatı kullanılmış, RAW etkiket formatı başarım ölçümü işlemleri öncesinde IOB2 formatına dönüştürülmüştür.

Son olarak yapay sinir ağı çalışmalarında kullanılmak üzere veri kümesi ve seçilen temsil özelliklerinin uygun bir şekilde sayısal temsili oluşturulmuştur. Sayısal temsil oluşturma çalışmalarında LSTM yapay sinir ağlarının gerçeklenmesinde girdi uzunluğunun sabitlenmesi gerektiğinden en uzun cümle boyu 50 sözcük olarak belirlenmiş, bundan uzun cümleler ise bölümlendirilmiştir.

5.3 Uygulanan Yapay Sinir Ağı Modelleri

Bölüm 4.4'de önerilen yapay sinir ağı modelleri farklı yapılandırmalarda LSTM, BLSTM ve DBLSTM yapay sinir ağı yapıları olarak gerçeklenmiştir. Deneyde gerçeklenen yapay sinir ağı modelleri LSTM katman sayıları ve unutma (Dropout) parametresine ait değerler Çizelge 5.6'da belirtilmiştir.

Çizelge 5.6 : Yapay sinir ağı modelleri.

Model	Katman Sayısı	Unutma
LSTM	1	0
BLSTM	1	0
DBLSTM (1)	2	0
DBLSTM (2)	3	0
DBLSTM (3)	5	0
DBLSTM (4)	3	0,4
DBLSTM (5)	5	0,4

5.4 Deneysel Sonuçlar

Deneysel çalışmaların çıktısı RAW etiket temsili ile elde edilmiştir. Sistemin başarımının ölçme değerlendirmesi yapılabilmesi için IOB2 yapısına dönüştürülmüştür. Ölçüm işlemlerinde CoNLL ölçüm metodolojisi kullanılmıştır. CoNLL ölçüm işlemleri CoNLL konferansları için hazırlanmış olan açık kaynaklı conlleval.pl[2] betiği kullanılarak gerçekleştirilmiştir.

Deneylerde 492K elemandan oluşan veri kümesi Şeker ve Eryiğit (2017) çalışmasındaki bölümlendirme ile kullanılmıştır. Kullanılan bölümlendirmede eğitim kümesinde 445K, test kümesinde ise 47K eleman yer almaktadır. Yapay sinir ağı eğitim işlemi eğitim kümesinin tamamı kullanılarak gerçekleştirilmiş, her bir iterasyon sonunda eğitilen sistemin test kümesi üzerindeki başarısı ölçülmüştür. Her bir veri modeli ve özellik kümesi için en başarılı ölçüm sonuçları sonuçları raporlanmıştır.

Öğrenme işlemlerine en basit modelden başlayarak veri kümesi kapsamında kullanılan özellikler arttırılarak ilerlenmiştir. Önerilen her model için:

- **A:** Temel (Sadece sözcük vektörleri kullanılarak)
- **B:** Temel + Yazım Özellikleri
- **C:** Temel + Yazım Özellikleri + Biçimbilimsel Özellikler
- **D:** Temel + Yazım Özellikleri + Biçimbilimsel Özellikler + Karakter Tabanlı Sözcük Vektörleri
- **E:** Temel + Yazım Özellikleri + Biçimbilimsel Özellikler + Karakter Tabanlı Sözcük Vektörleri + Sözlük Özellikleri

olmak üzere 4 farklı girdi kümesi ile öğrenme işlemleri yapılmış ve sonuçları Çizelge 5.7'de raporlanmıştır.

Çalışmanın sonuçlarında katman sayısının artmasının öğrenmeyi başarılı bir şekilde etkilediği gözlemlenmiştir. Özellikle tek katmanlı BLSTM modeline ikinci bir BLSTM katmanı eklendiğinde %10,28 F_1 puanı gibi kaydadeğer bir başarım artışı gözlemlenmiştir. Bununla birlikte BLSTM katman sayısının artması modelin

[2]https://github.com/Microsoft/CNTK/blob/master/Tests/EndToEndTests/Examples/Text/Miscellaneous/SLU/Config/conlleval.pl

Çizelge 5.7 : Yapay sinir ağı modellerinin geliştirme verisi başarımları.

Model	A	B	C	D	E
LSTM	72,00	75,26	75,75	82,54	82,50
BLSTM	78,74	82,25	82,67	82,42	82,14
DBLSTM (1)	89,02	92,39	92,49	90,83	91,86
DBLSTM (2)	90,03	92,90	92,65	93,59	91,83
DBLSTM (3)	91,05	93,22	93,58	91,77	92,73
DBLSTM (4)	90,58	92,60	93,69	92,65	93,06
DBLSTM (5)	91,59	93,33	93,64	93,52	93,03

başarısını arttırmakla birlikte modelin öğrenme süresini ve öğrenme işlemi esnasındaki donanım ihtiyacını da arttırdığı gözlemlenmiştir.

Bunun dışında yazım özelliklerinin modele eklenmesinin de başarıyı etkili bir şekilde değişirdiği sonuçlardan çıkarılmaktadır. Biçimbilim özelliklerinin modelin başarısına etkisi çok az olmakla birlikte DBLSTM(2) modelinde başarıyı olumsuz etkilediği gözlemlenmiştir. Biçimbilimsel özelliklerin sonuçları olumlu yönde çok iyileştirmemesini, zaten belirli bir başarıya ulaşmış olan bu model için biçimbilimsel çözümleme ve belirsizlik giderme araçlarının hata payının etkisinden kaynaklandığını tahmin etmekteyiz.

Karakter tabanlı sözcük vektörlerinin sisteme eklenmesi özellikle LSTM modelinin başarısını ciddi bir şekilde etkilediği, LSTM modelinin başarısını BLSTM modeline yakınlaştırdığını hatta daha başarılı sonuçlar vermesini sağladığı gözükmektedir. Daha karmaşık modellerde ise karakter tabanlı vektörlerin başarıyı beklenin aksine genellikle başarıyı olumsuz etkilediği gözlemlenmiştir. Bu durumu modelin karmaşıklaması ile birlikte girdi kümesine eklenen karakter tabanlı sözcük vektörlerinin önerdiğimiz yapay sinir ağı modeli ile başarılı bir şekilde işlenmediği şeklinde yorumlamaktayız. Verinin başarılı bir şekilde anlamlandırılabilmesi için yapay sinir ağı modelinin iyileştirmesi veya değiştirilmesi gerektiğini düşünüyoruz.

6. SONUÇ VE ÖNERİLER

Tez çalışması kapsamında LSTM, BLSTM ve DBLSTM yapay sinir ağı yapıları kullanılarak Türkçe'de ENAMEX varlık tipi için Varlık İsmi Tanıma Sistemi önerilmiş, önerilen modeller gerçeklenerek başarımları ölçülmüştür. Sistem başarımlarının ölçüldüğü ve raporlandığı Bölüm 5'te yapay sinir ağı tabanlı makine öğrenmesi sistemine ait sistem parametrelerinin ve seçilen girdi parametrelerinin başarımına etkileri detaylı bir şekilde raporlanmış ve yorumlanmıştır.

Türkçe VİT işlemleri için sözcük vektörleri, yazım özellikleri ve biçimbilimsel özellikler ile DBLSTM yapay sinir ağı kullanılarak başarılı sonuçlar elde edilmiştir. Oluşturulan DBLSTM (4) modeli ile CoNLL ölçümlerine göre ulaşılan %93.69 F_1 puanı, bilgimiz dahilinde olan çalışmalar arasında Türkçe VİT için ulaşılmış en iyi sonucu az bir farkla iyileştirmektedir.

Ayrıca yine karşılaştığımız kadarıyla önermiş olduğumuz DBLSTM (5) yapay sinir ağı modeli Temel Veri Kümesi ile %91.59 F_1 puanına ulaşmıştır. Erişilen bu sonuç sadece sözcük vektörleri kullanılarak ulaşılmış olması açısından başarılı bir sonuç olarak nitelendirilebilir.

İleriki çalışmalarda bu tez çalışmasına TIMEX, NUMEX gibi veri tipleri için de varlık ismi tanıma işlemi yapacak kural tabanlı bir sistem eklenmesi ve tez çalışması kapsamında oluşturulan ENAMEX varlık ismi tanıma sistemi de kullanılarak hibrit bir varlık ismi tanıma sistemi geliştirilmesi hedeflenmektedir.

KAYNAKLAR

[1] **Dale, R.**, **Moisl, H. ve Somers, H.** (2000). *Handbook of natural language processing*, CRC Press.

[2] **Rau, L.F.** (1991). Extracting company names from text, *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, cilt 1, IEEE, s.29–32.

[3] **Tjong Kim Sang, E.F. ve De Meulder, F.** (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, Association for Computational Linguistics, s.142–147.

[4] **Grishman, R.**, (1995), MUC-6 Named Entity Task Definition, `https://cs.nyu.edu/faculty/grishman/NEtask20.book_1.html`.

[5] **Güngör, O.**, **Yıldız, E.**, **Üsküdarlı, S. ve Güngör, T.** (2017). Morphological Embeddings for Named Entity Recognition in Morphologically Rich Languages, *arXiv preprint arXiv:1706.00506*.

[6] **Grishman, R. ve Sundheim, B.** (1996). Message understanding conference-6: A brief history, *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, cilt 1.

[7] **Nadeau, D. ve Sekine, S.** (2007). A survey of named entity recognition and classification, *Lingvisticae Investigationes*, *30*(1), 3–26.

[8] **Cucerzan, S. ve Yarowsky, D.** (1999). Language independent named entity recognition combining morphological and contextual evidence, *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC*, s.90–99.

[9] **Tür, G.**, **Hakkani-Tür, D. ve Oflazer, K.** (2003). A statistical information extraction system for Turkish, *Natural Language Engineering*, *9*(2), 181–210.

[10] **Bayraktar, O. ve Temizel, T.T.** (2008). Person name extraction from turkish financial news text using local grammar-based approach, *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on*, IEEE, s.1–4.

[11] **Küçük, D. ve Yazıcı, A.** (2009). Named entity recognition experiments on Turkish texts, *International Conference on Flexible Query Answering Systems*, Springer, s.524–535.

[12] **Yeniterzi, R.** (2011). Exploiting morphology in Turkish named entity recognition system, *Proceedings of the ACL 2011 Student Session*, Association for Computational Linguistics, s.105–110.

[13] **Tatar, S. ve Çiçekli,** (2011). Automatic rule learning exploiting morphological features for named entity recognition in Turkish, *Journal of Information Science*, *37*(2), 137–151.

[14] **Özkaya, S. ve Diri, B.** (2011). Named entity recognition by conditional random fields from Turkish informal texts, *Signal Processing and Communications Applications (SIU), 2011 IEEE 19th Conference on*, IEEE, s.662–665.

[15] **Şeker, G.A. ve Eryiğit, G.** (2012). Initial Explorations on using CRFs for Turkish Named Entity Recognition, *Coling*, s.2459–2474.

[16] **Şeker, G.A. ve Eryiğit, G.** (2017). Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content1, *Semantic Web*, *8*(5), 625–642.

[17] **Demir, H. ve Özgür, A.** (2014). Improving named entity recognition for morphologically rich languages using word embeddings, *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, IEEE, s.117–122.

[18] **Kuru, O.**, **Can, O.A. ve Yüret, D.** (2016). Charner: Character-level named entity recognition, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, s.911–921.

[19] **Sekine, S. ve Nobata, C.** (2004). Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy., *LREC*, Lisbon, Portugal, s.1977–1980.

[20] **Thielen, C.** (1995). An approach to proper name tagging for german, *arXiv preprint cmp-lg/9506024*.

[21] **Ravin, Y. ve Wacholder, N.** (1997). *Extracting names from natural-language text*, Citeseer.

[22] **Mikheev, A.** (1999). A knowledge-free method for capitalized word disambiguation, *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, Association for Computational Linguistics, s.159–166.

[23] **Da Silva, J.F.**, **Kozareva, Z. ve Lopes, J.G.P.** (2004). Cluster Analysis and Classification of Named Entities., *LREC*.

[24] **Sang, E.F. ve Veenstra, J.** (1999). Representing text chunks, *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, s.173–179.

[25] **McCulloch, W.S. ve Pitts, W.** (1943). A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics*, *5*(4), 115–133.

[26] **Olah, C.**, (2015), Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[27] **Hochreiter, S. ve Schmidhuber, J.** (1997). LSTM can solve hard long time lag problems, *Advances in neural information processing systems*, s.473–479.

[28] **Gers, F.A.**, **Schmidhuber, J. ve Cummins, F.** (1999). Learning to forget: Continual prediction with LSTM.

[29] **Gers, F.A.**, **Schraudolph, N.N. ve Schmidhuber, J.** (2002). Learning precise timing with LSTM recurrent networks, *Journal of machine learning research*, *3*(Aug), 115–143.

[30] **Bengio, Y.**, **Ducharme, R.**, **Vincent, P. ve Jauvin, C.** (2003). A neural probabilistic language model, *Journal of machine learning research*, *3*(Feb), 1137–1155.

[31] **Collobert, R. ve Weston, J.** (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning, *Proceedings of the 25th international conference on Machine learning*, ACM, s.160–167.

[32] **Turian, J.**, **Ratinov, L. ve Bengio, Y.** (2010). Word representations: a simple and general method for semi-supervised learning, *Proceedings of the 48th annual meeting of the association for computational linguistics*, Association for Computational Linguistics, s.384–394.

[33] **Mikolov, T.**, **Sutskever, I.**, **Chen, K.**, **Corrado, G.S. ve Dean, J.** (2013). Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems*, s.3111–3119.

[34] **Oflazer, K.**, **Say, B.**, **Hakkani-Tür, D.Z. ve Tür, G.**, (2003). Building a Turkish treebank, Treebanks, Springer, s.261–277.

[35] **Hankamer, J.** (1989). Morphological parsing and the lexicon, *Lexical representation and process*, MIT Press, s.392–408.

[36] **Sak, H.**, **Güngör, T. ve Saraçlar, M.**, (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus, Springer, s.417–427.

[37] **Oflazer, K.** (1994). Two-level description of Turkish morphology, *Literary and linguistic computing*, *9*(2), 137–148.

[38] **Şahin, H.B.**, **Tirkaz, C.**, **Yıldız, E.**, **Eren, M.T. ve Sönmez, O.O.** (2017). Automatically Annotated Turkish Corpus for Named Entity Recognition and Text Categorization using Large-Scale Gazetteers, *CoRR*, *abs/1702.02363*, http://arxiv.org/abs/1702.02363, 1702.02363.

ÖZGEÇMİŞ

Ad Soyad: Asım Güneş

Doğum Tarihi ve Yeri: 01.02.1985 / İstanbul

E-Posta: asim.gunes@itu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans:** 2008, İstanbul Teknik Üniversitesi, Telekomünikasyon Mühendisliği

MESLEKİ DENEYİMLER VE ÖDÜLLER:

- 2004-2008 yılları arasında İTÜ Bilgi İşlem Daire Başkanlığı Yazılım Geliştirme Grubunda yarı zamanlı, tam zamanlı ve grup başkanı pozisyonlarında görev almıştır.
- 2007-2008 yılları arasında İTÜ Bilgi İşlem Daire Başkanlığında Microsoft IT Academy Koordinatörlüğü görevini yürütmüştür.
- 2009-2010 yılları arasında Mikro Ödeme A.Ş.'de IT Yöneticiliği görevini yürütmüştür.
- 2010 yılından bu yana İTÜ Bilgi İşlem Daire Başkanlığında Yazılım Geliştirme Grubu Başkanlığı görevini yürütmektedir.

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- Güneş A., Tantuğ A.C., 2018. Derin Öğrenme ile Türkçe'de Varlık İsmi Tanıma. *26. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, May 2-5, 2018 Çeşme, Turkey.

DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- S. Cayir, A. Gunes, O. Buk, 2008. Türkiye'deki Kamu Kurumlarında Bilişim Teknolojileri Yönetişimi. *Akademik Bilisim 2008*, 30.01.2008 - 01.02.2008, Çanakkale, Turkey.
- A. Gunes, S. Cayir, 2007. Ninova e-Öğrenim Sistemi. *Akademik Bilisim 2007*, 31.01.2007 - 02.02.2007, Kütahya, Turkey.
- G. Akin, A. Gunes, 2007. Bir Worm'un Anatomisi. *Akademik Bilisim 2007*, 31.01.2007 - 02.02.2007, Kütahya, Turkey.
- Güneş, A., Akın, G., 2006. Kullanıcı Ağ Erişim Yönetimi. *Akademik Bilisim 2006*, 09.02.2006 - 11.02.2006, Denizli, Turkey.