T.C.
EGE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

DİLE ÖZGÜ ÖZNİTELİKLERİ KULLANAN BİR VARLIK İSMİ TANIMA SİSTEMİ GELİŞTİRİLMESİ

Ergin ALTINTAŞ

Danışman: Prof. Dr. Oğuz DİKENELLİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Doktora Programı

İzmir

2023

Ergin ALTINTAŞ tarafından Doktora tezi olarak sunulan "DİLE ÖZGÜ ÖZNİTELİKLERİ KULLANAN BİR VARLIK İSMİ TANIMA SİSTEMİ GELİŞTİRİLMESİ" başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 11 Temmuz 2023 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri üyeleri **İmza**

Jüri Başkanı : Prof.Dr.Oğuz DİKENELLİ

Raportör Üye : Dr.Öğr.Üyesi Özgün YILMAZ

Üye : Prof.Dr. Yalçın ÇEBİ

Üye : Prof.Dr. Murat Osman ÜNALIR

Üye : Doç.Dr. Selma TEKİR

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum "DİLE ÖZGÜ ÖZNİTELİKLERİ KULLANAN BİR VARLIK İSMİ TANIMA SİSTEMİ GELİŞTİRİLMESİ" başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

11/07/2023

Ergin ALTINTAŞ

ÖZET

DİLE ÖZGÜ ÖZNİTELİKLERİ KULLANAN BİR VARLIK İSMİ TANIMA SİSTEMİ GELİŞTİRİLMESİ

ALTINTAŞ, Ergin

Doktora Tezi, Bilgisayar Mühendisliği Anabilim Dalı
Tez Danışmanı: Prof. Dr. Oğuz DİKENELLİ
Tarih, 90 sayfa

Her bir doğal dil, pek çok farklı kurala ve bu kuralların bir araya gelmesi ile oluşan karakteristik yapılara sahiptir. Ancak son dönemlerde doğal dil işleme alanında büyük başarılar elde eden BERT gibi modeller, metin kodlama (tokenizasyon) işlemi dilden bağımsız olarak yapmaktadır. Dolayısıyla modelin başarılı olabilmesi için dilin karakteristik özelliklerine ait örüntülerin de modelin kendisi tarafından temsil edilmek üzere öğrenilmesi gerekmektedir.

Dilin kendine özgü özniteliklerini dikkate alan bir modelle Türkçe için daha yüksek bir başarım elde edilebileceği varsayımından yola çıktığımız bu çalışmada BERT modelinde kullanılan WordPiece isimli tokenizer bileşeni yerine Türkçe'nin temel ses dönüşüm özelliklerini dikkate alan yeni bir tokenizer geliştirilmiş ve bu tokenizer kullanılarak tamamen sıfırdan yeni dil modelleri eğitilmiştir. Yeni modelin başırmı orjinal modelin başarımı ile karşılaştırıldığında varlık ismi tanıma problemi özelinde, uygulanan yeni modelin eğitim hızını önemli ölçüde iyileştirdiği ve başarımda da hissedilir derecede gelişme sağlandığı gözlemlenmiştir.

Sonuç olarak, bu çalışmada Türkçe için özgün bir model geliştirilmiş ve üç alanda katkı sağlanmıştır: Türkçe'ye uygun bir tokenizasyon algoritması oluşturulmuş, bu algoritmayı kullanarak yeni bir dil modeli eğitilmiş ve elde edilen model Türkçe varlık ismi tanıma probleminde uygulanmıştır.

Anahtar sözcükler: Doğal Dil İşleme, Varlık İsmi Tanıma, Türkçe Dil Modelleri, Türkçe Tokenizasyon, Derin Sinir Ağları.

ABSTRACT

DEVELOPMENT OF NAMED ENTITY RECOGNITION SYSTEM EXPLOITING LANGUAGE SPECIFIC FEATURES

ALTINTAŞ, Ergin

Ph.D. in Department of Computer Engineering

Supervisor: Prof. Dr. Oğuz DİKENELLİ

Tarih, 90 pages

Each natural language has various rules and characteristic structures formed by the combination of these rules. However, recent models in natural language processing, such as BERT, perform text encoding (tokenization) independently of the language. Therefore, to achieve successful performance, the patterns of the language's characteristic features also need to be learned and represented by the model itself.

Based on the assumption that a model considering the unique attributes of the language could achieve higher performance for Turkish, in this study, we developed a new tokenizer that takes into account the fundamental sound transformation features of Turkish, instead of the WordPiece tokenizer used in BERT models. Using this new tokenizer, we trained entirely new language models from scratch. Comparing the performance of the new model with the original model specifically in the named entity recognition task, we observed significant improvement in training speed and noticeable enhancement in performance.

In conclusion, we developed an original model for Turkish in this study and made contributions in three areas: we created a tokenizer tailored for Turkish, trained new language models using this algorithm, and applied them to the named entity recognition problem in Turkish.

Keywords: Natural Language Processing, Named Entity Recognition, Turkish Language Models, Turkish Tokenization, Deep Neural Networks.

ÖNSÖZ

Yapay zeka ilintili konular lise yıllarımdan itibaren tutku ile araştırdığım konular olagelmiştir. Doktora çalışmamı da doğal dil işleme problem sahasında devam ettirmeye, bağlam ve anlam ilişkisinin önemini kavradığım yüksek lisans çalışmam sırasında karar vermiştim.

Vektör tabanlı kelime temsil modellerinin derin öğrenme teknikleri ile harmanlanması ile günümüzde artık kelime anlamlarının matematiksel modellemesine ilişkin oldukça etkili çözümlere ulaşılmıştır. Bu çözümleri temel alan ChatGPT gibi dil modellerinin elde ettiği görünürlük günlük haber yayınlarına kadar yansımış durumdadır.

Hem teorik hem de uygulama açısından beni tatmin eden bu çalışmanın olgunlaşması sürecinde özellikle derin sinir ağı modellerinin başarımlarını makul süreler içinde gözlemleyebilme ihtiyacı bana derin öğrenme tekniklerine yönelik mühendislik birikimimi de geliştirme fırsatı sağladı.

Bu çalışmadaki deneylerde uygulaması yapılan çok katmanlı transformer ağı mimarisindeki dil modellerinin, günümüz için doğal dil işlemeye yönelik en etkili derin öğrenme mimarisi olarak kendilerine yer edinmiş olduğu söylenebilir. Bununla birlikte Türkçe özelinde literatürde bu alandaki çalışmaların sayısı halen oldukça azdır. Şahsen önümüzdeki dönemlerde hem dil modelleri ile ilgili olarak hem de Türkçe doğal dil işleme çalışmaları özelinde heyecan verici gelişmelerin devamının geleceğine inanmaktayım.

İZMİR
11/07/2023

Ergin ALTINTAŞ

İÇİNDEKİLER

Sayfa

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler	Açıklama
BERT	Bidirectional Transformers
Bi-LSTM	Bidirectional Long Short Term Memory
BPE	Byte-pair Encoding
DDİ	Doğal Dil İşleme
IOB	Bir VİT etiketleme şeması
LMU	Linguistically Meaningful Units
LSTM	Long Short Term Memory
NER	Named Entity Recognition
RNN	Recurrent Neural Network
TRUBA	Türk Ulusal Bilim e-Altyapısı
VİT	Varlık İsmi Tanıma

1 GİRİŞ

Bu bölümde öncelikle doktora tezinin neden yazıldığı ve kapsanan araştırma konuları ile birlikte hangi problemlerin çözülmesinin amaçlandığı belirtilecektir. Daha sonra araştırma konularına olan yaklaşım ve önerilen çözüm tartışılarak tezin literatüre katkılarına değinilecektir. Son olarak, tezin yapısı hakkında genel bir fikir edinilmesini kolaylaştırmak maksadıyla takip eden bölümlerdeki konu başlıklarına kısaca değinilecektir.

1.1 Problem Tanımı

Varlık ismi tanıma (VİT, NER, Named Entity Recognition), doğal dil işleme alanındaki temel ve önemli problemlerden birisidir. VİT görevinde amaç metinlerde geçen varlık isimlerini (kişiler, kurumlar, yerler, ürünler vb.) otomatik olarak tanımak ve etiketlemektir. VİT probleminin zorlukları arasında metin içindeki isimlerin farklı varyasyonlarını ele almak, belirsizlikleri çözmek ve yanlış tanıma oranını en aza indirmek yer alır.

Diğer doğal dil işleme problemlerinde olduğu gibi, literatürde üzerinde yaygın şekilde çalışma yapılmış diller arasında yer alan İngilizce gibi dillerde, VİT görevi için de yaygın çalışmalar mevcuttur. Bununla birlikte Türkçe, Çekçe, Macarca, Fince gibi morfolojik açıdan da zengin olarak nitelendirilen bazı dillerde VİT görevine yönelik çalışmalar sınırlı seviyede kalmaktadır. Geleneksel VİT yaklaşımları arasında kural tabanlı, kelime tabanlı, istatistiksel, dil bilgisi tabanlı ve etiketlenmiş veriye dayanan makine öğrenmesi yaklaşımları sayılabilir. Bununla birlikte son zamanlardaki derin öğrenme tekniklerine dayalı Transformer tabanlı dil modelleri de kullanılan yeni ve oldukça başarılı yaklaşımlar arasında yerini almıştır.

Bu çalışmada, VİT probleminin çözümü için Transformer tabanlı modellerden birisi olan BERT modelinde uygulanan yaklaşımın Türkçe içerikli metinler için daha verimli ve başarılı bir şekilde çözüme ulaştırılabilmesi için Türkçe'ye özgü öznitelikleri dikkate alacak şekilde geliştirilmesi yöntemi seçilmiştir.

1.2 Tezin Kapsamı

Türkçe'nin literatürde üzerinde en çok çalışma yapılmış dillerden farklı karakteristiklere sahip olmasına neden olan özniteliklerini dikkate alan yaklaşımlar özellikle de son dönemde doğal dil işleme alanında önemli gelişmelere sebep olmuş transformer tabanlı dil modelleri ile bir arada ele alındıklarında sadece VİT probleminde değil, muhtemelen doğal dil işlemenin farklı problem sahalarında da Türkçe özelinde daha yüksek başarımlar elde edilmesine katkı sağlayabilir. Bununla birlikte tez çalışmasının sınırlı kaynaklarla ve sınırlı bir süre içinde tamamlanması zorunlu olduğundan bu tez kapsamındaki deneysel uygulamalar sadece VİT alanı ile sınırlı tutulmuştur.

1.3 Tezin Katkıları

Türkçe dilinin kendine özgü özniteliklerini dikkate alan bir modelle, VİT görevi özelinde Türkçe için daha yüksek bir başarım elde edilebileceği varsayımından yola çıktığımız bu çalışmada 3 farklı alanda literatüre katkı sağlanmıştır.

Bu kapsamda birinci katkı olarak, Google tarafından dilden bağımsız bir şekilde kullanılmak için tasarlanmış olan BERT modeli taban model olarak kullanılmış, BERT modelinde kullanılan WordPiece isimli tokenizer bileşeni yerine Türkçe'nin temel ses dönüşüm özelliklerini dikkate alan yeni bir tokenizer geliştirilmiştir. İkinci katkı olarak temeli BERT modeline dayanan ancak geliştirilen yeni tokenizer kullanılarak tamamen sıfırdan eğitilmiş olması dolayısıyla Türkçe'nin ses dönüşüm özelliklerini dikkate alan yeni bir dil modeli elde edilmiştir. Son olarak da, Türkçe için özelleştirilmiş tokenizer kullanılarak eğitilen yeni modelin başırmı, aynı veri seti üzerinde dil bağımsız olarak eğitilen orjinal modelin başarımı ile karşılaştırıldığında VİT problemi özelinde, uygulanan yeni modelin eğitim hızını önemli ölçüde iyileştirdiği ve başarımda da hissedilir derecede gelişme sağlandığı gözlemlenmiştir.

Böylece Türkçe'ye özgü bir tokenizer ile sıfırdan eğitilen bir model ile VİT problemi özelinde, görece daha yüksek başarımlara daha kısa eğitim süreleri ile ulaşılabildiği gösterilerek Türkçe doğal dil işleme literatürüne katkı

sağlanmıştır.

1.4 Tezin Bölümleri

Bu bölümde tezin maksadı, kapsamı, ele alınan problem sahası ve çözüm için önerilen yaklaşım ve tezin katkılarına kısaca değinilmiştir. Müteakip paragraflarda ise, tezin yapısı hakkında genel bir fikir edinilmesini kolaylaştırmak maksadıyla takip eden bölümlerdeki konu başlıklarına kısaca değinilecektir.

Tezin ikinci bölümünde doğal dil işleme, özellikle de Türkçe doğal dil işleme alanındaki temel zorluklara, bağlam ve anlam ilişkisine, metinlerin bilgisayarla işlenebilmesi için gerekli olan temel aşamalardan biri olan tokenizasyon algoritmalarına, doğal dil işlemede yapay sinir ağlarının ve derin öğrenme yöntemlerinin nasıl kullanılabileceğine değinilmiştir. Bu bağlamda özellikle doğal dil işleme alanında büyük ilgi odağı olmuş olan transfer öğrenmesine dayalı modellere ve bu alanda tarihsel süreçte ortaya çıkan önemli dönüm noktalarına ilişkin çalışmalara değinilmiştir. Ayrıca bu bölümde son olarak VİT probleminin ne olduğuna, çözüm için kullanılan bazı modellere ve Türkçe için şimdiye kadar uygulanan bazı yaklaşımlara yer verilmiştir.

Tezin üçüncü bölümünde tez çalışması kapsamı içinde yapılmış olan temel katkı olarak değerlendirilen ve WordPiece algortimasının Türkçe'nin özniteliklerine uygun bazı ses dönüşüm özelliklerne dayanan iyileştirmelerle geliştirmiş hali olarak özetlenebilecek WordPieceTR tokenizasyon algoritması ile BERT modelinin bu algoritma kullanılarak ön-eğitme aşamalarının yeniden uygulanmış hali olan BERT-TR Türkçe'ye özgü dil modelinin detayları ve nihai modele ulaşmak için uygulanan geliştirme safhaları sırasıyla anlatılmıştır.

Tezin dördüncü bölümünde hem tezin ilk aşamalarında mevcut modellerin sentezlenmesi ile gerçekleştirilmiş uygulamalar ve sonuçlarından hem de tez çalışması kapsamı içinde ortaya çıkarılmış olan BERT-TR modeli ile yapılan uygulamalı çalışmaların içeriği ve bu çalışmalardan elde edilen sonuçlar kıyaslamalı olarak ortaya konulmuştur.

Tezin sonuç bölümünde ise tez çalışmasında elde edilen sonuçlar de-

ğerlendirilmiş, dile özgün doğal dil işleme çalışmalarının günümüzdeki yerine ilişkin değerlendirmeler yapılmış, elde edilen yeni tokenizer ve dil model kullanılabileceği farklı doğal dil işleme görevlerinden bahsedilmiş ve ileride yapılabilecek çalışmalara yönelik önerilerden ve açık problem sahalarından bahsedilmiştir.

2 ARKAPLAN

Bu bölümde öncelikle doğal dillere ve doğal dil işlemenin temel zorluk alanlarına, sonrasında Türk dilinin doğal diller arasındaki yerine ve Türkçenin üzerinde yaygın şekilde araştırmalar yapılmış diğer dillerden farklarını ortaya çıkaran belli başlı özniteliklerine, metinlerin bilgisayarla işlenebilmesi için gerekli olan temel aşamalardan biri olan tokenizasyon algoritmalarına, doğal dil işlemede yapay sinir ağlarının ve derin öğrenme yöntemlerinin nasıl kullanıldığına değinilecek, sonrasında ve doğal dil işlemenin temel problem alanlarından birisi olan VİT konusunda şimdiye kadar yapılmış çalışmalara değinilerek metin işlemede bağlam ve anlam ilişkisinin önemini ortaya koyan örnekler verilecektir.

2.1 Doğal Diller ve Türk Dili

İnsan zihninde dilin nasıl işlendiği ve üretildiği konusunda dikkate değer teorileri bulunan ve dilbilimin alanında yaptığı çalışmalarla tanınan önemli bir bilim insanı olan Noam Chomsky gibi dilbilimciler insan beyninin ve içinde gerçekleşen bilişsel süreçlerin, doğanın bilinen ancak henüz tam olarak çözülememiş en karmaşık sistemlerden biri olduğunu savunmuşlardır (Chomsky, 2006). Chomsky, aynı zamanda dil yeteneğinin insan beyninde doğuştan gelen bir yapıya sahip olduğunu ve dilin karmaşıklığının, basit öğrenme mekanizmalarıyla açıklanamayacak kadar derin ve karmaşık olduğunu öne sürmüştür.

Bilişsel süreçlerin tam olarak nasıl işlediği ve bu süreçler içinde dilin nasıl bir süreç içinde işlenmekte olduğu gibi konular hala aktif bir şekilde araştırılan konular olması bu savın temelsiz bir sav olmadığına dair bir işaret olarak görülebilir. Bu alandaki araştırmalar, insan beyninin karmaşıklığını anlamak ve bilişsel süreçleri daha iyi açıklamak için şüphesiz önemli katkılarda bulunmaktadır.

Bununla birlikte yapay zeka ve doğal dil işleme alanında son yıllarda pek çok heyecan verici gelişme meydana gelmiştir. Bu gelişmelerden birisi de derin

öğrenme tekniklerinin kullanımının yaygınlaşması ve doğal dil işleme dahil pek çok farklı alanda başarılı görülen uygulamalarının sayısının artmasıdır.

Derin öğrenme yaklaşımı esasen çok sayıda katmandan oluşan yapay sinir ağlarına dayanan ve bu sinir ağlarını çok büyük veri kümeleri kullanarak ama olabildiğince verimli bir şekilde eğitmek için kullanan bir tür makine öğrenmesi yaklaşımıdır. Yapay sinir ağları ise insan beyninin sinir ağlarından ilham alan ve çeşitli eğitme algoritmaları ile eğitilebilen bir veri yapısıdır. Her iki yapıya ilişkin matematiksel hesaplamaların temelde büyük boyutlu matrislerin ardışık şekilde çarpımına dayandığı ifade edilebilir. Derin sinir ağlarındaki katman sayısı arttıkça hesaplanması gereken matris boyutları da gittikçe büyümekte ve bu ağların verimli şekilde eğitimi için ihtiyaç duyulan işlem gücü de sürekli olarak artmaktadır.

Günlük hayatın ayrılmaz bir parçası olan, canlılar arasında olay ve olgulara ilişkin bilgilerin, duyguların ve düşüncelerin paylaşımı için yoğun bir şekilde kullanılması yoluyla zekanın dışa vurumunun da yapıldığı önemli bir kanal olarak da görülebilecek olan doğal dil, temelde anlatma ve anlama işlevlerini gerçekleştirmek üzere kullanılan bir iletişim sistemi olarak da ele alınabilir. Bu şekilde bakıldığında dilin öncel olarak sözlü bir anlatım aracı olduğu hatta dilde iletişimin en temel yönteminin sesli iletişim olduğu da ifade edilebilir. Tüm bu yaklaşımlar temel olarak yanlış olmamakla birlikte, iletişimde mimik ve benzeri beden dili unsurlarının da yer aldığı yadsınmamalıdır. Dolayısıyla söz konusu anlatma ve anlama sürecini, iletilmek istenen mesajların anlatıcının beyni tarafından koordine edilen bir dizi zihnî, anatomik, fizyolojik işlemlerle fiziksel niceliklere yani sembollere dönüştürülerek karşı tarafın alabileceği şekilde aktarılması ve iletiyi taşıyan fiziksel niceliklerin ya da sembollerin karşı tarafın duyu organları tarafından algılanarak elektrokimyasal yollarla beyin kabuğunun (korteks) ilgili kısımlarına aktarılarak anlamlandırılması olarak tanımlayıp, dili de bütün bu süreçlerin birleşimi olarak nitelediğimizde oldukça kapsayıcı bir tanıma ulaşmış oluruz. Algılama, tanımlama, düşünme, hatırlama, kavrama gibi tüm zihinsel faaliyetlerde dilin de işlevsel bir rolü olduğu gibi, bireylerin zihinsel gelişim süreçleri de dil yeteneği ile paralel

süreçlerle iç içe gelişmektedir.

Bu nedenle doğal dili salt bir iletişim aracı olarak görmek doğru olmayacaktır, zira dil var olmadan da iletişim olabilmekte bazı durumlarda da iletişim mevcut olmasa da düşünce ile dil mevcut olabilmektedir. Örneğin resim, müzik gibi doğal dil dışında sayılabilecek araçlarla da insanlar arasında duygu ve düşüncelerin paylaşılması da bir iletişim yöntemi olarak görülebilir. İnsanlar genelde düşüncelerini yazılı veya sözlü olarak doğal dille ifade ederler ve düşüncelerini bu şekilde aktarıp yayarlar. Bununla birlikte doğal dil, insanların düşüncelerini daha iyi organize etmeleri için de bir araç olarak kullanılır. İç konuşmalar yaparak ya da yazarak düşünme pek çok kişinin kullandığı bir yöntemdir. Dolayısıyla doğal dil düşüncenin sadece aktarımında değil oluşum ve gelişiminde de önemli bir rol oynar. Bu yönüyle doğal dile insan zihninin önemli bir hareket alanı olarak da bakılabilir, dolayısıyla DDİ çalışmaları insan beyninin derin gizemlerine aralanın bir kapı olarak da görülebilir.

Doğal dilin kullanımında temel amacın iletişim olduğu kabulu ile ilerlediğimizi varsayalım. İletişim bir iletinin göndericisinden alıcısına ulaştırılması olayı olarak özetlenebilir. İletişim aracılığı ile gönderilen iletinin doğruluğu, açıklığı ve anlaşılırlığı, iletişimin başarısını etkileyen önemli faktörlerdir. Bununla birlikte iletişim süreci sadece iletinin gönderimi ile sınırlı değildir; aynı zamanda iletinin anlaşılması ve yorumlanması da önemlidir. Bu nedenle iletişim olgusu bütünsel bir yaklaşım gerektirir. Zira iletişimin başarıya ulaşması için iletinin alıcısında hedeflenen yorumlamayı oluşturması da bir gerekliliktir. Dolayısıyla aynı dili konuşan (veya konuşuyor gibi görünen) iki insanın kavrayış çerçeveleri, olgulara bakış açıları, zihinsel bağıntılılıkları çok farklı ise sağlıklı ve başarılı bir iletişim kurmaları zorlaşabilecektir. İşte bu nedenle iletişim hedefi ile sınırlarınmış bir kapsama alanı içinde dahi, salt sözcük anlamları ve tümce yapıları ile ilgilenmek doğal dilin anlaşılması için çoğunlukla yetersiz kalmaktadır.

Doğal dilin, düşünceleri ve duyguları ifade etme gücü, dilin yapısına ve kurallarına da dayalı olduğu gibi aynı zamanda dilin kelime hazinesinin büyüklüğüne ve anlam zenginliğine de bağlıdır. Doğal dil ile gerçekleştirilmeye

çalışılan iletişimin etkin ve verimli olmasını engelleyen ve sağlıklı bir iletişim kurulmasında güçlükler yaratan etmenlerin bütünü "dil bariyeri" olarak da ifade edilir. Dil bariyeri, sadece birbirlerinin kullandığı dilleri bilmeyen (yabancı dillere sahip olan) insanlar arasında değil, yukarıda örneklediğimiz gibi aynı dili konuşan fakat farklı kültürlerden olan insanlar arasında da ortaya çıkabilir.

Günümüzde halen insanlar arasında iletişimi ve anlaşmayı sağlayan en gelişmiş, en etkili ve en yaygın araç olan doğal dil yeteneğimiz bir yönüyle bizim için doğal bir miras olduğu gibi onun kullanıcıları olan bizler tarafından sürekli olarak yeni kavramlar ve kelimeler üretilmekte olması yönüyle de sürekli değişen ve gelişen canlı bir organizma gibidir. Dolayısıyla doğal dil yalnızca aynı tarihsel dönem içinde yaşayan insanlar arasında bir anlaşma aracı değil; aynı zamanda geçmişin birikimini geleceğe taşıyan ve bir anlamda insanlığın belleğininin yapı taşlarını oluşturan canlı (süreç içinde sürekli olarak gelişip değişen) bir varlık olarak görülebilir. Özetle dil insanların ihtiyaçlarına ve değişen koşullara göre yenilenip çağa uyum sağlayarak değişebilen ve böylece sanki yaşayan bir varlık gibi sürekli bir gelişim içinde olan bir olgudur.

Doğal diller, ses, biçim, sözdizimi ve anlam yükleri gibi bileşenlerden oluşur, bu bileşenler bir araya gelerek dilin karakteristik özelliklerini belirlerler. Doğal dilin en temel iletişim ortamı ve aynı zamanda en temel bileşenini oluşturan sesler, ağız, burun ve dil gibi organlar kullanarak üretilir. Doğal dillerin doğal gelişim süreci itibariyle de öncelikle sözlü bir anlatım aracı olduğu düşünülmesi de genel olarak doğru bir yaklaşımdır. Bununla birlikte, dil sistemi içerisinde mimikler ya da benzeri görsel beden dili unsurları da önemli bir rol oynar. İletişim sırasında sadece seslerle değil, aynı zamanda beden dilimiz ve mimiklerimiz de kullanılır. Bu görsel unsurlar, iletişim sırasında düşüncelerimizi, duygularımızı ve niyetlerimizi ifade etmemize yardımcı olur. Örneğin, bir kişinin gülümsemesi mutluluğunu, kaşlarının çatılması ise şüpheciliğini ifade edebilir. Bu nedenle doğal dilin, anlatma ve anlama işlevlerini gerçekleştirmek üzere birbirleriyle ilişkili ses, biçim, sözdizimi ile soyut veya somut anlam yükleri gibi bileşenlerin bir arada meydana getirdiği bir sistem

olarak düşünülmesi daha kapsayıcı ve daha doğru bir yaklaşım olacaktır.

Dil sisteminde önemli bir bileşeni olan, anlam yükleri, dilin kullanımı sırasında sözcüklerin anlamını değiştirerek ve belirginleştirerek dilin anlamsal zenginliğini artırırlar. Örneğin;

"Saat" kelimesi normalde zamanı gösteren bir aracı ifade ederken, "Saat kaç?" soru kalıbı içinde belirgin bir anlam yükü kazanır ve sormak istediğimiz şeyin zaman olduğunu net bir şekilde belirtir.

"Güzel" kelimesi, genellikle estetik bir güzelliği ifade ederken, "Hava ne kadar güzel." cümlesinde, "güzel" sözcüğü havanın iyi olduğunu ifade eden bir anlam kazanır.

"Yüksek" kelimesi, genellikle bir nesnenin dikey uzunluğunu ifade ederken, "Yüksek sesle konuşma." cümlesinde, "yüksek" sözcüğü sesin güçlü olduğunu ifade eder şekilde anlam kazanır.

"Küçük" kelimesi, genellikle bir nesnenin uzayda kapladığı hacimin büyüklüğünü ifade ederken, "Bu kadar küçük bir konu için kavga çıkarmaya değmez." cümlesinde, "küçük" sözcüğü konunun önemsiz olduğunu ifade eder.

"Yük" kelimesi, genellikle taşıt, araba, hayvan vb.nin taşıdığı şeyleri veya bu şeylerin miktarını ifade ederken "yükünü hafifletmek" ifadesinde farklı bir anlam yükü kazanarak kişinin sıkıntılarını ya da sorumluluklarını (azaltmayı) ifade eder.

Bu örneklerde görüldüğü gibi, sözcüklerin anlamları dilin kullanımı sırasında değişkenlik gösterebilir ve bu değişkenlikler sözcükler için farklı anlam yükleri meydana getirirler.

Özellikle sayısal iletişim ve bilgi üretme olanaklarının çok geliştiği günümüzde, özellikle sayısal ortamda ciddi miktarda yazılı bilgi kaynağı oluşturulduğu bir gerçektir. Sayısal ortamdaki yazılı bilgi kaynakları, basitçe internet üzerinden erişilebilen veya bilgisayarlarda saklanan elektronik dökümanlar olarak görülebilir. E-kitaplar, bloglar, wikiler, online web siteleri, online

gazete/dergiler gibi materyallerin arasında iletişim ağırlıklı kullanılan e-posta, sosyal medya platformları, iş veya kişisel maksatlı anlık mesajlaşma uygulamaları gibi ortamların da önemli bir kaynak olduğu dolayısıyla günümüzde yazılı iletişimin de doğal dil için çok geniş bir alan oluşturduğu göz ardı edilemez.

Bedenimizin en önemli öğesi olan beyin yüzyıllar boyunca doğa bilimleri kapsamında incelenegelmiştir. Bu öğe, canlı varlıkların yapı ve süreçleriyle ilgilenen temel biyolojik bilimlerin; madde ve enerji konularıyla ilgilenen fiziksel bilimler ile tıp gibi uygulamalı bilimlerin; canlıların davranışlarıyla ilgilenen davranış bilimleri ile psikolojinin uğraş konuları içindedir. Birbirinden oldukça uzak görünen bu alanların her birinin kendi içinde kapalı olarak çalışmasının beyin ve bilişsel süreçler arasındaki ilişkinin araştırılmasında verimsizliklere neden olduğu günümüzde bilim dünyası genelinde kabul edilen bir durumdur.

İnsana beyni olan diğer canlılardan büyük bir ayrıcalık sağladığı kabul edilen, gelişmiş doğal dil yeteneği; insan yavrusunun doğduğu andan itibaren hazır olarak bulunan bir yetenek değildir. Bu yetenek her bir insan yavrusunda en az 18 ay kadar süren fizyolojik, biyolojik ve sosyal bir sürecin sonucunda ortaya çıkmaktadır. Bu süreç içerisinde ilk 6 ay civarında öncelikle "ba-ba", "ma-ma", "de-de" şeklindeki kolay hecelerin oluşumu, sonrasında ilk 12 ay civarında basit kelimelerin tekil olarak ifade edilmesi görülür. Süreç, 18'inci aylardan sonra ise "anne gel" gibi az sayıda kelimeden oluşan basit cümlelerin söylenmesi şeklinde devam eder. Daha sonra insan yavrusu; anne, baba, diğer aile fertleri ve yakın çevresindeki diğer canlılar ile etkileşim içine girerek dil yeteneğini tam olarak kazanmaya başlar. Çocukta dil yeteneğinin gelişimi, özellikle 2-6 yaşlar arasında çok hızlanır. Bununla birlikte bu sürecin farklı iç veya dış etkenler nedeniyle daha uzun bir periyot içinde gerçekleşmesi de olasıdır. Çocukların okuma ve yazma yetenekleri ancak sözlü iletişim yeteneğinin gelişmesi sonrasında ortaya çıkabilmektedir. (Temizyurek, 2007) Yani doğal dilin temelinde sesli iletişimin yer aldığı yazılı doğal dil yeteneklerinin genellikle sonradan ortaya çıktığı gözlemlenmektedir (DemiRci, 2011).

Dil yeteneğinin beynin işleyişi ile olan bağlantısı bugüne kadar genellikle üç bölümde ele alınmıştır. Birincisi, bugüne kadar özellikle nöroloji ve fizyoloji

alanlarındaki çalışmalardan elde edilen bulgular yardımıyla dilin gerçekleşmesinde beynin hangi bölgelerinin daha etkili olduğunun belirlenmesi; ikincisi, dilbilimcilerin bakışıyla dil sisteminin kendi içindeki işleyişi; üçüncüsü de bireyin kullanımında ortaya çıkan somut eylemin, yani konuşmanın görünümü ve özelliklerinin gözlemlenmesidir. İnsan beyninin işleyişini çözmeye çalışan nörologlar ve fizyologlar, elde ettikleri bulgularla dil sisteminin işleyişindeki kimi yönlerin aydınlanmasına katkıda bulunurken, dilbilimciler de asıl inceleme konuları olan insan dilinin evrensel özelliklerinden yola çıkarak sistemin işleyişini belirlemeye çalışmaktadırlar. Disiplinler arası çalışmaların ve bu çalışmalardan elde edilen verilerin yoğunlaştırılması ile bilişsel süreçlere yönelik bilinmezlerin çözümünde ilerlemeler kat edilebilecektir (Ergenç, 2000).

Dil aynı zamanda sürekli değişen ve gelişen canlı bir organizma gibidir. Sürekli olarak yeni kavramlar ve kelimeler üretilmektedir. İnsan beyni, doğal dili anlamada ve hatta geliştirmede çok başarılıdır; doğru kelimeleri seçerek çok ayrıntılı ve farklı anlamları ifade edebilir, algılayabilir ve yorumlayabilir. Bu mükemmel organ sayesinde biz insanlar, dil yeteneklerimizi mükemmel olarak kullanabilir iken, kullandığımız dilin yapısını belirleyen kuralları (bir bilgisayar programlama dilinin yapısına benzer şekilde) kesin olarak tanımlayarak ortaya koyma konusunda zorlanırız. Dilde bu gibi belirsizlikler içeren ve dolayısıyla kesin kurallarla tarif edilemeyen durumların varlığı makinaların insan dilini işleyip anlayabilmesi konusunda çözülmesi gereken ve tek düze olmayan pek çok karmaşık problemin temelinde yatan önemli sebeplerin başında gelmektedir. Bu nedenle doğal dil anlama (Natural Language Understanding - NLU) probleminin doğru şekilde çözülmesi ile insan zekasının sırlarının ortaya çıkarılması arasında önemli bir ilişki olduğu düşünülebilir. Yapay zeka terimini ilk ortaya atan Alan Turing tarafından zekanın varlığının incelenmesi için ortaya atılan meşhur Turing Testi'nin (Turing, 1950) de doğal dil anlama ve üretme odaklı bir test olması bundan olsa gerektir.

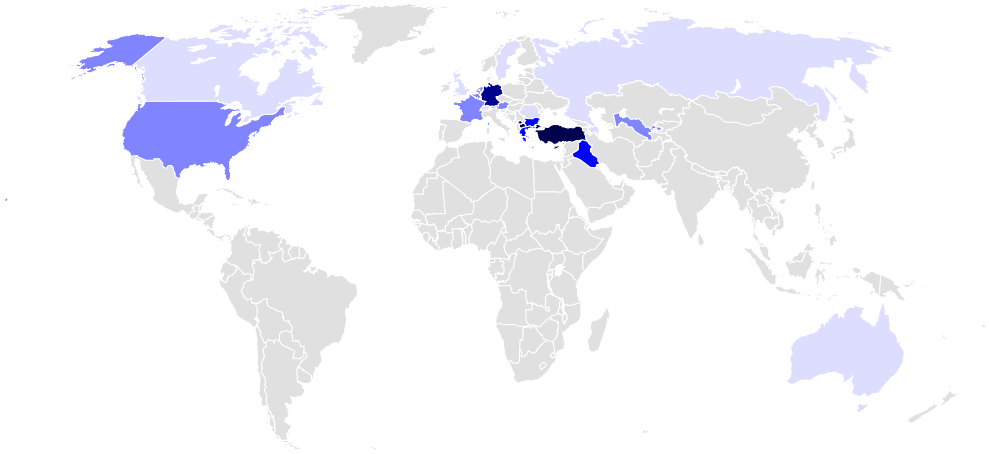
Doğal dil işleme (Natural Language Processing - NLP), doğal dilin (yani insanların konuştuğu ve yazdığı dilin) makine tarafından anlaşılıp işlenmesi olarak özetlenebilir. DDİ çalışmaları, makine öğrenimi, veri madenciliği,

veri analizi ve yapay zeka gibi alanlarda yapılan çalışmaları içerir. DDİ çalışmaları pratikte arama motorlarının daha verimli çalışmasına ve böylece insanların doğru bilgilere daha hızlı ulaşmalarına, makine çevirisinin daha düzgün yapılmasına ve böylece insanlar arasındaki dil bariyerlerinin ortadan kalkmasına imkan sağlar. Bu somut faydaları ile DDİ insan yaşamını gerçekten kolaylaştıran önemli bir alandır.

Tüm bu etmenler bir arada ele alındığında, hem dil bariyerlerinin aşılması gibi pratik uygulama alanları için büyük avantajlar sağlayan, hem de insan zihninin sırlarına kapı aralama potansiyeli olan doğal dil işleme alanında yapılan çalışmaların insanlık için gerçekten büyük önem taşıdığı ortaya çıkmaktadır.

2.2 Türkçenin Öznitellikleri

Türkçe Altay dil grubuna dahil olup bu grubun en büyük koludur. Türkçe, Orta Asya'dan Anadolu'ya yerleşen Türkler tarafından konuşulagelen bir dil olup dil tarihi açısından da çok zengin bir dildir. Türkçe, Azerbaycan, Kazakistan, Kırgızistan, Özbekistan ve Rusya gibi ülkelerde konuşulan dilleri (Başkurt ve Tatar dilleri) içeren Türk dil ailesinin bir üyesidir. Dünya genelinde yaklaşık 220 milyon kişi tarafından konuşulmaktadır (Akalın, 2009).



Şekil 2.1: Türkçe konuşanların dünya üzerinde yayılımı (Vikipedi, 2023)

Türkçe, Türkiye Cumhuriyeti ve Kuzey Kıbrıs Türk Cumhuriyeti'nin resmi dilidir. Ayrıca Almanya, Bulgaristan, Yunanistan, Romanya, Makedonya,

Bosna-Hersek, Kosova, Sırbistan, Hırvatistan, Avusturya, Kosova, Hollanda, Irak, İran, Suriye gibi pek çok farklı ülkede yaşamakta olan Türkler tarafından konuşulmaktadır. (Bkz. Şekil 2.1)

İslamiyet'in yayılması ve Osmanlı İmparatorluğu'nun oluşumu ile birlikte Türkçe tarihi süreç içinde Farsça, Arapça ve diğer diller ile gerçekleşen etkileşimler ile zenginleşmiştir. Tüm doğal diller bu tarz etkileşimlerle gelişip büyümekte ve hayatlarına devam etmektedirler.

Türkçe'de gramatik cinsiyet mefhumu mevcut olmadığından cümlelerde cinsiyet farkından kaynaklanan değişiklikler meydana gelmez. Yapısal açıdan incelendiğinde Türkçe eklemeli (agglutinative) diller grubunda yer alır. Türkçe ile birlikte Macarca, Fince, Moğolca gibi diller de bu grupta yer alır. Türkçe sondan eklemeli bir dil olup, eklerdeki zenginlik ve çeşitlilik dikkat çekicidir.

Bir çalışmada Türkçe dışındaki 50 farklı dilin kullanım sıklığı çok az olan bazı kelimelere ait istisnai durumlar dışında insanla ilişkili pek çok alanda geçerliliği gösterilmiş olan Zipf yasası bağlamında benzer örtüntüler sergilediği gösterilmiştir (Yu et al., 2018). Zipf yasası temel olarak bil dilde kullanılan kelimeler kullanım sıklığına gör sıralandığında, en sık kullanılan kelimeden en seyrek kullanılan kelimeye doğru kelime kullanımına sıklık değerlerinin kabaca doğrusal bir şekilde azalacağını ifade eder. Türk Dili'nin istatistiki nitellikleri bağlamında kelime frekanslarının dağılımını inceleyen bir çalışmada Türk dili'nin de insanla ilişkili pek çok alanda geçerliliği bulunan Zipf yasası'na genel olarak uyduğu gösterilmiştir (Dalkılıç and Çebi, 2005).

Doğal diller bu ve bezeri pek çok özellik açısından çeşitli benzerlikler gösterebilmektedirler ancak Türkçe'yi diğer dillerden ayrı kılan pek çok özellik de mevcuttur. Sonraki başlıklarda bu özelliklerden tez çalışması kapsamında ele alınmış önemli özelliklere değinilmeye çalışılacaktır.

2.2.1 Morfolojik Zenginlik

Dilbilimsel açıdan ele alındığında, morfoloji, dilin yapısal düzeyiyle ilgilenen, kelimelerin yapısal öğeleri olan morfemleri ve bunların hangi kurallara

göre bir araya geldiğini inceleyen bir disiplindir. Morfemler bir araya gelerek kelimeleri oluşturan dilin en küçük anlamlı birimleri olarak ele alınabilirler. Morfoloji ayrıca bir dilin içinde yer alan kelime türleri, çekim ekleri, köklerin yapısı çeşitli dilbilgisel yapılarla da ilgilenir. Bir dile ilişkin morfolojik kurallar, dildeki eklerin, köklerin, birleşik kelimelerin ve diğer morfemlerin nasıl bir araya geldiğini ortaya koyar. (Haspelmath and Sims, 2010)

Morfoloji sadece morfolojik açıdan zengin dillerde değil, neredeyse tüm dillerde kelimelerin anlamına etki eden önemli bir konudur. Önek veya soneklerle kelimelerin farklı anlamlara gelmesi İngilizce gibi dillerde de söz konusu olabilmektedir. Dolayısıyla tüm dillerdeki doğal dil işleme çalışmalarında morfolojiyi doğru şekilde ele alabilmek önemlidir. Ancak, bazı diller de vardır ki bu dillerde ekler ve eklerle ortaya çıkarılan yeni anlamlar dilin ana çatısını oluştururlar. Morfolojik açıdan zengin olarak nitelendirlen bu dillerde ad ve eylemlerin kök ya da gövdelerine getirilen eklerle farklı kavram ve anlamların türetilmesi çok daha yaygındır.

"Dilbilimde morfolojik açıdan zengin" ifadesi, bir dilin morfolojik yapısıyla ilgili olarak geniş ve karmaşık bir yapıya sahip olduğunu ifade eder. Morfolojik açıdan zengin bir dil, kelime oluşturma ve kelime yapısıyla ilgili olarak zengin bir varyasyona ve kompleksliğe sahiptir. Bu tür dillerde, sözcük dağarcığı geniş, kelime formları çeşitlidir ve dilbilgisel yapılar daha karmaşıktır. Morfolojik açıdan zengin bir dilde sözcüklerin oluşumunda morfem çeşitliliğinin fazla olması, dilde çok sayıda ve çeşitli ek yapılarının varlığı, kelime köklerinin farklı biçimlerde türetilebilmesi gibi özelliklere bulunur. Bu özellikler, dildeki kelime çeşitliliğini artırır ve anlamlı kelime formlarının sayıca fazla olabilmesine imkan sağlar. Bu nedenle eklemeli diller grubuna da giren Türkçe, Macarca ve Fince gibi morfolojik açıdan zengin dillerde morfolojinin anlama etkisi çok daha yüksektir (Virtanen et al., 2019) (Luoma et al., 2021). Tablo 2.1'de "yük" köküne getirilen iki ekle elde edilebilen farklı anlamlar örneklenmiştir.

Türkçe'nin literatürde üzerinde en çok çalışma yapılmış dillerden biri olan İngilizce ve benzeri dillere göre en önemli farklarından birisi aynı zamanda sondan eklemeli bir dil olmasıdır. Eklemeli dillerde yeni kelimeler

Kök+ek(ler)	**Anlam**
yük	Araba, hayvan vb.nin taşıdığı şeylerin hepsi.
yük-le(mek)	Bir yere, taşınması için belli ağırlıkta eşya veya araç gereç koymak.
yük-le-m	Cümlede oluş, iş ve hareket bildiren kelime veya kelime grubu.

Tablo 2.1: "Yük" köküne gelen eklerle oluşan bazı farklı anlamlar

ve terimler türetmek hem kolay hem de yaygın bir durumdur. Türkçe'nin de gelişmiş ve zengin bir ek sistemi mevcut olduğundan yeni kelimeler türetmeye elverişli bir dildir. Bu tür dillerde her bir ek, kelimenin anlamını tamamen değiştirebilecek bir işleve sahip olabilmektedir (Göksel and Kerslake, 2005). Örneğin Türkçe'de göz, gözlük, gözlükçü, gözlükçülük gibi kelimelerin hepsinin kökü ortak olmasına rağmen bu kelimelerin her biri tamamen farklı anlamlara sahiptirler.

Eklemeli diller istatistiki açıdan incelendiğinde de diğer dillere göre farklı özellikler sergilediklerinden (Dalkılıç and Çebi, 2005), Word2Vec ya da benzeri algoritmalarda kelime temsillerinin olası kelime altı parçalar dikkate alınmadan bir bütün olarak ele alınmasının (Mikolov et al., 2013a) başarımı olumsuz yönde etkilediği, Türkçe gibi morfolojik açıdan zengin olan eklemeli dillerde kelime altı parçaların ayrıştırılabiliriğini dikkate alarak gerçekleştirilen çalışmaların daha başarılı sonuçlar verdiği görülmektedir. (Cotterell and Schütze, 2015) (Zhang et al., 2015) (Chen et al., 2018) (Church, 2020)

Türkçe için eklerle türetilebilecek kelime çeşitliliğinin teoride sınırsız olduğu söylenebilir. Türkçedeki sözcüklerin neredeyse tamamı "kök+ek+ek+..." yapısına sahip olup, çoğunlukla türetilmiş kelimelerdir. Salt kök'ten ibaret olan kelime sayısı azdır. Eklerle türetilebilecek kelime çeşitliliği hakkında kaba bir fikir vermesi için geç fiil kökünden türetilen bazı kelimeleri Şekil 2.2'de listeledik.

Şekil 2.2'de örneklenen kelimelerin hepsi aynı köke (geç) sahip olmasına

geçebilecek, geçebileceği, geçebileceğine, geçebilen, geçebilme, geçecekken, geçeceğim, geçemediğimiz, geçememesinin, geçemeyenler, geçemeyince, geçemezken, geçen, geçende, geçenin, geçenler, geçenlerde, geçenleri, geçenlerle, geçer, geçercesine, geçerek, geçerken, geçerler, geçerli, geçerlik, geçerlilik, geçerliydi, geçerliyse, geçersek, geçersiz, geçersizlik, geçici, geçit, geçidinden, geçildi, geçilebilecek, geçilebilir, geçilebilmiş, geçilememesi, geçilemeyen, geçilemez, geçilen, geçilerek, geçilince, geçilmemesi, geçilmemiş, geçilmesi, geçilmez, geçilmiş, geçim, geçimli, geçimsiz, geçimsizlik, geçindirmek, geçinemeyen, geçinen, geçiniz, geçip, geçit, geçitlerindeki, geçitli, geçiş, geçişindeki, geçişkenlik, geçiştir, geçiştirdiler, geçiştirilebilecek, geçkin, geçmeyen, geçmiş, geçmişlerindeki, geçmiştekinden, geçmişten, geçtikçe...

Şekil 2.2: Türkçede "geç" köküne sahip bazı kelimeler

rağmen çok farklı anlamlara gelmektedirler, hatta aralarında "geçmiş (bugüne göre geride kalmış olan zaman, mazi)", "geçit (geçmeye yarayan yer, geçecek yer)", "geçim (anlaşma, uyum veya maişet)" vb. gibi anlam itibariyle bir birlerinden oldukça uzak anlama sahip olanlar da mevcuttur.

Türkçe gibi sondan eklemeli dillerde, her küçük bir ek, kelimenin anlamını hem de cümle içindeki fonksiyonunu (isim, sıfat, eylem vb.) tamamen değiştirebilecek bir işleve sahip olabilmektedir. Bazı durumlarda ise eklerin meydana getirdiği anlam değişiklikleri küçük olabilmektedir. Hatta bazı durumlarda aynı sesleniş/yazılışlara sahip olmasına rağmen esas itibariyle farklı işlevlere sahip ekler de kullanılabilmektedir. (Korkmaz, 2017)

$$\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{ruhsat}_{isim}+lan}_{eylem}+dır}_{eylem}+ıl+ama}_{eylem}+ma+sı+nda}_{isim}+ki}_{sıfat}$$

Şekil 2.3: Eklerin kelimenin fonksyionunu (ve anlamını) değiştirmesi (Oflazer, 2018)

Türkçenin sondan eklemeli yapısı sayesinde bu şekilde çok rahat yüzlerce kelime türetilebilmektedir. Bu nedenle Türkçe gibi sondan eklemeli ve morfolojik açıdan zengin dillerde kelimelerin bütününü tek bir token olarak

ele alan yöntemlerin başarımının sınırlı düzeyde kaldığı, bu karşın kök+ek ayrımını temsil edebilecek şekilde karakter tabanlı veya kelime altı parçaları token olarak ele alabilen temsil yöntemlerinin daha başarılı sonuçlar verdiği görülmektedir (Joulin et al., 2016) (Bojanowski et al., 2016) (Kuru et al., 2016) (Park et al., 2018).

Türkçe sondan eklemeli bir dil olduğu için, ekler esasen Türkçedeki en önemli unsurlardandır. Türkçede sözcük köklerine getirilen eklerle, yeni anlamda sözcükler türetilebildiği gibi ekler cümledeki sözcüklerin arasında anlam ilişkileri kurmak için de zorunludur. Türkçede ekler, yapım eki ve çekim eki olmak üzere ikiye ayrılır. Çekim ekleri her zaman yapım eklerinden sonra gelir. Yani yapım ekleri çekim eklerinden sonra gelemez. Yabancı kökenli bazı sözcükler hariç, Türkçede ön ek bulunmaz. Çekim ekleri, isim ve fiillerin kök ya da gövdelerine eklenerek kelimelerin cümle içinde zaman, yön, aitlik, tekil ya da çoğulluk gibi anlamları üstlenmelerini ve böylece cümle içindeki diğer kelimelerle anlam bağlantısı kurulmasını sağlayan eklerdir. Örneğin "ev-i-miz", "gör-dü-m" gibi. Yapım ekleri ise, aynı şekilde isim ya da fiil kök veya gövdelerine eklenerek, genellikle ana temanın çok dışına çıkmadan farklı bir anlam ortaya çıkararak yeni bir kelime oluşturan eklerdir.

Türkçe'de bazı anlamlar hem birleşik yazılan eklerle hem de ayrı yazılan kelimelerle sağlanabilmektedir. Bu kelimelerin, kullanım kolaylığı açısından zamanla ek durumuna geçtikleri düşünülebilir. Bu duruma örnek olarak "ile" sözcüğünün, hem ayrı olarak, hem de ek olarak kullanılması verilebilir: "el yordamı ile => el yordamıyla", "anahtar ile => anahtarla" vb.

Ekeylemlerin çekimleri de hem ayrı sözcük olarak hem de ek olarak kullanılabilmektedir. Örnekler: "söylemiş idi => söylemiş-ti" "her ne ise => her ne-y-se" vb.

Türkçe bazı anlamların burada örneklendiği şekilde hem ek hem de ayrı kelimeler ile sağlanabiliyor olması da, Türkçe gibi diller için eklerin ayrı anlamlar taşıyabilen dil birimleri olarak ele alınması gerektiği görüşünü destekler niteliktedir.

Türkçe	**İngilizce çeviri örnekleri**
Bilmediğimizden.	Because we don't know.
	It's one of the ones that we don't know
İlişkilendiremedik-	It's one of the ones that we couldn't relate
lerimizdendir.	It's probably because of the ones that we couldn't relate

Tablo 2.2: Farklı anlamlı çeviri örnekleri

Ünlü	**İnce Ünlü**		**Kalın Ünlü**	
	Düz	**Yuvarlak**	**Düz**	**Yuvarlak**
Dar	i	ü	ı	u
Geniş	e	ö	a	o

Tablo 2.3: Türkçe'de özelliklerine göre ünlü tipleri

Ayrıca Tablo 2.2'de yer alan örneklerde görüldüğü gibi, türkçede tek bir kelimeden oluşan cümleler kurulabilmekte ve bu cümlenin başka dillere çevrildiğinde karşılığı pek çok kelimeden oluşabilmektedir, hatta bazı durumlarda bu şekilde birden farklı çeviri olasılığı oluşabilmektedir:

Bununla birlikte, özellikle yabancı dillerin etkisi altında gerçekleşen morfolojik açıdan hatalı olarak da değerlendirilebilecek olan bazı dil durum/dönüşümleri dolayısıyla zaman zaman genel geçer kurallara uymayan kök-ek ilişkileri de görülebilmektedir. Örnekler: altın => altun-î , gidiş => gidiş-at, ayrı => ayrı-yeten, otlak => ot-la-k-iye vb.

Türkçe'ye benzer olarak Macarca, Fince ve Estonca gibi Ural-Altay dillerinde de değişen uzunluklarda farklı ekler ile gösterilebilen düzinelerce gramer durumu vardır.

2.2.2 Ses Dönüşümleri

Türkçe'de (ve diğer Türk dillerinde de) göze çarpan en temel kural olarak ünlü uyumu görülür. Ünlü uyumu kısaca bir kelimedeki tüm ünlülerin aynı tipte olması zorunluluğu olarak ifade edilebilir.

Özellik	**Öncesi**	**Sonrası**
Ünsüz sertleşmesi	seç(k)in	seç-gin
Ünsüz yumuşaması	aldık-ı	aldı(ğ)ı
Ünsüz düşmesi	ufa(k)-cık	ufacık
Ünsüz türemesi	af-etti	af(f)etti
Kaynaştırma	masa-ı	masa(y)ı
Ünlü düşmesi	em(i)r-etti	emretti
Ünlü türemesi	dar-cık	dar(a)cık
Ünlü daralması	bekle-yor	bekl(i)yor
Ünlü kalınlaşması	sen-e	s(a)n(a)

Tablo 2.4: Türkçe'de ses dönüşüm örnekleri

Türkçede özellikle eklerin köklere ve biribirilerine bağlanması sırasında meydana gelen ünlü uyumu, ünsüz uyumu, ünlü düşmesi ve ünsüz düşmesi gibi aşağıda detaylandırılan ses dönüşümleri söz konusudur. (Taş, 2021)

Büyük ünlü uyumu bağlamında Türkçede iki ünlü tipi vardır: kalın ünlüler (a, ı, o, u) ve ince ünlüler (e, i, ö, ü). Türkçe'de yan yana gelen iki hece içindeki ünlülerin tipleri genellikle aynı olur. Dolayısıyla bir kök ya da gövdeye gelen bir ek içindeki ünlü farklı tipte ise genellikle eklendiği gövdeye uyum sağlayacak şekilde diğer tipe dönüşür. Nadir bazı durumlarda ise kökteki ünlü veya ünlülerin eklenen kısma uyum sağlayacak şekilde dönüşmesi mümkündür.

Tablo 2.3

Türkçede ekler ses yapısı bakımından köklere uyum sağladıkları için eklerin kalın ve ince ünlüler içeren birden fazla biçimi bulunduğu da düşünülebilir. Buna en güzel örnek olarak çoğul eki "-lar, -ler" verilebilir (Zulfikar, 2013). Büyük-küçük ünlü uyumu ve diğer ünlü-ünsüz uyumları sebebiyle eklerin seslerinde değişim meydana gelebilmektedir.

Örnek: "çocuk-lar, öğrenci-ler"

Almanca ve Portekizce gibi bazı dillerde de, bir kelimeye bir son ek

eklediğinizde bazen kökteki sesli harfin telaffuzunun değişmesi gibi buna benzer durumlar oluşabilmektedir.

Küçük ünlü uyumunun iki temel kuralı vardır. Bir kelimede düz ünlüden sonra düz, yuvarlak ünlüden sonra yuvarlak dar veya düz geniş ünlüler bulunur:

Düz ünlü (a, e, ı, i) içeren heceden sonraki hecede de düz ünlü (a, e, ı, i) bulunmalıdır. (Dolayısıla o kelimedeki izleyen tüm heceler düz ünlü içerir.) Başka bir deyişle "a, e, ı, i" seslerini içeren hecelerden sonra ancak "a, e, ı, i" seslerini içeren heceler gelir.

Yuvarlak ünlü (o, ö, u, ü) içeren heceden sonraki hecede geniş düz (a, e) veya dar yuvarlak (u, ü) ünlü bulunmalıdır. Başka bir deyişle "o, ö, u, ü" seslerini içeren hecelerden sonra ancak "a, e, u, ü" seslerini içeren heceler gelir.

Örnekler: anlaşmak, anlaşmalı, kayıkçı, ısırmak, ılıklaşmak, seslenmek, yelek, bilek, çilek, yeşil; boyunduruk, börekçi, çocuk, güreşçi, ocakçı, odun, özlemek, sürmek, sormak, yoklamak, yorgunluk, yumurta, yüreksiz vb.

Bununla birlikte küçük ünlü uyumuna uymayan Türkçe kelimeler de vardır:

Örnekler: avuç, avurt, çamur, kabuk, yağmur vb.

Ünlü-ünsüz uyumları sebebiyle eklerin seslerinde değişim meydana gelebilmektedir.

Örnekler:

Bilinen geçmiş zaman eki: -dı, -di, -du, -dü, -tı, – ti, -tu, -tü

oku-du, yaz-dı, bil-di, koş-tu, geç-ti, git-ti, bit-ti.

Küçük ünlü uyumuna aykırı bazı kelimelere getirilen ekler, genellikle kelimenin son ünlüsüne uyar.

Örnekler: kavun-u, konsolos-luk, muzır-lık, müzik-çi, yağmur-luk vb.

Bununla birlikte bazı yabancı kökenli kelimelerde aykırı durumlar olu-

şabilir. Ayrıca “-ki” aitlik eki yalnızca birkaç örnekte (bugünkü, dünkü vb.) küçük ünlü uyumuna uysa da her zaman bu kurala uyum sağlamak zorunda değildir.

Örnekler: alkol-lü, kabul-ü, bandrol-lü, saat-lik, oduncunun-ki vb.

Burada genel şekilde ile alınan Türkçe dil özelliklerinin arasında ünsüz sertleşmesi, ünsüz yumuşaması, ünsüz düşmesi, ünsüz türemesi, ünlü düşmesi, ünlü türemesi, ünlü kalınlaşması, ünlü daralması, kaynaştırma ilave pek çok farklı ses olayı mevcuttur. Örneğin İki ünlü harfin yan yana gelmesi sonucunda iki ünlü arasına genellikle “y” bazen de “’n, s, ş” harfleri gelir, bu durumu kaynaşma denmektedir. Benzer şekilde “af” kelimesine “-etmek” eki eklendiğinde affetmek olur ve kelime ile ek arasında ilave bir ”f” harfi türer. Benzer şekilde “dar” kelimesine ”cık” eki eklendiğinde daracık olarak yazılır ve kelime ile ek arasında “’a” sesi türemiş olur.

2.2.3 Türkçe’de Hece Yapısı

Türkçe’de heceleme yapısı genel olarak aşağıdaki gibi basit kurallara bağlanabilir (Gonenc, 1973):

- Her hece içinde bir ünlü harf yer alır.

  Örnekler: a, e, ı, i, o, u, ö
- Ünsüz harfler yanlarındaki ünlü harflerle bir araya gelerek hece oluştururlar.

  Örnekler: al, az, et, kral, tren
- Kelime içinde iki ünlü harf arasında yer alan bir ünsüz harf, kendisinden sonraki ünlüyle aynı hece içinde yer alır.

  Örnekler: a-be-ce, ge-li-şi-ne, ka-ra-ca, ka-pı-cı, ku-yum-cu, kra-liçe, tre-loy-büs
- Kelime içinde iki ünlü harf arasında yan yana olacak şekilde yer alan iki ünsüzden ilki kendisinden önceki ünlüyle, ikincisi kendisinden sonraki ünlüyle aynı hece içinde yer alır.

  Örnekler: kar-ta-ca, kap-lı-ca, kuy-ruk-suz, sev-gi, duy-gu, kor-kunç
- Kelime içinde iki ünlü harf arasında bir araya gelen üç ünsüz harften ilk

ikisi kendisinden önceki ünlüyle, üçüncüsü kendisinden sonraki ünlüyle aynı hece içinde yer alır.

Örnekler: Türk-çe, üst-len-di, kurt-lan-mış, sırt-lan, yo-ğurt-çu, borç-lan-dı, trenç-kot

2.3 Varlık İsmi Tanıma

VİT doğal dil işleme alanındaki temel ve önemli problemlerden birisi olup genellikle bilgi bilgi çıkarım sisteminin en temel bileşeni olarak kabul edilir. Bu nedenle geniş çapta ilgi görmüş ve 1990'lı yıllardan günümüze pek çok araştırmaya konu olmuştur. VİT alanını geliştirmek için yarışmalar şeklinde etkinlikler düzenlenmiştir. Her bir etkinlik farklı bir alanı hedeflemiş ve görevin farklı bir tanımını gerektirmiştir. VİT görevi ilk kez 1995 ABD Ulusal Standartlar ve Teknoloji Enstitüsü ile ABD Savunma İleri Araştırma Projeleri Ajansı tarafından düzenlenen bir dizi etkinlik olan Mesaj Anlama Konferanslarından 6'ıncısı olan MUC-6 konferasında tanıtılarak sonraki etkinliklere de dahil edilmiştir. (Chinchor, 1998) Bu etkinlikler, karşılaşılan zorluklar ve ihtiyaçlar doğrultusunda gerçekleştirilmiştir. Haber ajansı metinleri gibi genel alandan başlayarak, daha sonra askeri raporlara, terörizm odaklı bilgi çıkarım içeriklerine, bloglara ve en son olarak sosyal medya içeriklerine yönelen ve tek dilde başlayıp çok dilli bir çerçeveye geçen bir süreç izlenmiştir (Leaman and Gonzalez, 2008).

VİT görevinde amaç metinlerde geçen varlık isimlerini otomatik olarak tanımak ve etiketlemektir. VİT çalışmalarında hedeflenen varlık isimleri ağırlıklı olarak kişi, kurum ve yer isimlerinden oluşmaktadır. Bu hedef küme problem sahasına göre zaman, ürün gibi ek isimlendirmeleri içerecek şekilde de genişletilebilir. Örneğin, bir gazete makalesinde geçen varlık isimlerini otomatik olarak tanıyıp etiketlediğimizde, ilgili kişilerin, yerlerin veya olayların analizini çok daha rahat bir şekilde yapabilir hale geliriz. VİT çözümlerinden bilgi çıkarma, metin sınıflandırma veya diğer metin tabanlı görevlerin çözümünde faydalı bilgiler sağlayan yardımcı bir unsur olarak faydalanmak da mümkündür.

VİT yaklaşımları için literatürde sunulan çeşitli sınıflandırma şekilleri mevcuttur. Borthwick, VİT yaklaşımlarını gerekli bilgiyi sistemlere sağlama mekanizmasına göre; kural tabanlı, otomatik ve hibrit tabanlı olarak sınıflandırır. (Borthwick, 1996) (Küçük and Yazici, 2010) Otomatik yaklaşım, makine öğrenimi algoritmalarının etiketli verilerden sınıflandırma modelini oluşturmak için kullanıldığı modelleri simgeler. Literatürde VİT çalışmalarını (Chitnis and DeNero, 2015a)dil, tür, öğrenme yöntemi ve özellik uzayı gibi faktörlere bağlı olarak nasıl farklılık gösterdiklerine göre sınıflandıran çalışmalar da mevcuttur. (Nadeau and Sekine, 2007) (Yadav and Bethard, 2018) (Li et al., 2022)

Derin öğrenme tekniklerinin ortaya çıkmasından önce kullanılmakta olan VİT yöntemlerini, geleneksel VİT yaklaşımları olarak isimlendirebiliriz. Bunlar arasında kural tabanlı, kelime tabanlı, istatistiksel, dil bilgisi tabanlı ve etiketlenmiş veriye dayanan makine öğrenmesi yaklaşımları sayılabilir:

1. Kural ve kelime tabanlı yaklaşımlarda önceden tanımlanmış kurallar, kelime listeleri ve önermeler kullanarak metin içerisindeki varlık isimleri tanımlanmaya çalışılır. Örneğin bir metinde büyük harfle başlayan bir kelime sonrasında ilk harfi büyük olan bir kelime daha varsa bu kelimelerin kişi isimleri olma olasılığı yüksektir. Benzer olarak "Bay" veya "Bayan" ile başlayan bir kelime grubu bir kişi adı olarak kabul edilebilir. Bu yaklaşım genellikle el ile tanımlanan kurallar kümesine dayanır ve genellikle dilbilgisi veya belirli bir alandaki bilgilerle desteklenebilir.
2. İstatistiksel Yaklaşımlar: Bu yaklaşımlar, metinleri istatistiksel özelliklerine göre analiz eder. Metinlerdeki kelime sıklıklarını ve olasılıklarını kullanarak varlık isimlerini belirlemeye çalışır. İstatistiksel VİT yaklaşımında metinde yer alan kelime dizilerindeki gizli durumları (varlık ismi olma durumu) tahmin etmeye çalışan Hidden Markov Modellerinin (HMM) kullanımı yaygındır.
3. Dilbilgisi tabanlı yaklaşımlar metinlerdeki dilbilgisi yapılarını ve aralarındaki ilişkileri kullanarak varlık isimlerini tanımlamaya çalışır. Örneğin, bir cümlenin öznesi veya nesnesi olarak işaretlenen kelimelerin kişi veya yer isimleri olma olasılığı yüksektir. Dilbilgisi tabanlı yaklaşımlar genel-

likle dilbilgisi kaynaklarını (örneğin, dilbilgisi kuralları, kelime dağarcığı vb.) kullanır.

4. Klasik makine öğrenmesi yaklaşımları genellikle etiketlenmiş veriye dayanır ve bir veri setinde el ile etiketlenmiş örneklerin kullanıldığı makine öğrenmesi tekniklerini içerir. Önceden etiketlenmiş veri setleri ve sınıflandırma algoritmaları kullanılarak, metin içerisindeki varlıkları tanımlamak için çeşitli makine öğrenmesi teknikleri uygulanır. Etiketli veri setleriyle eğitilen modeller modele daha önce gösterilmemiş farklı metinlerde varlık isimlerini tahmin etmek için kullanılır. Bu maksatla kullanılabilecek algoritmalar arasında Karar Ağaçları, Destek Vektör Makineleri (SVM) ve Naive Bayes sayılabilir.

Bu geleneksel VİT yaklaşımları, eldeki veri setlerinin sınırlı olduğu, belirgin hesaplama gücü kısıtlarının mevcut olduğu, özel bir dil veya belirli bir alan için özel bir VİT modeli oluşturulmasının gerekli olması gibi geleneksel yöntemlerin hala daha uygun bir seçenek olarak hala tercih edilebileceği özel durumlar söz konusu olabilir. (Alfonseca and Manandhar, 2002) Ancak, derin öğrenme tabanlı VİT yöntemleri, son yıllarda geleneksel yöntemlere kıyasla büyük bir ilerleme kaydetmiştir. (Goyal et al., 2018)

Güncel derin öğrenme modelleri olarak VİT alanında yaygın olarak kullanılan modeller arasında uzun-kısa süreli bellek ağı (BiLSTM) ve koşullu rastgele alan (CRF) kombinasyonu olan BiLSTM-CRF modeli (Huang et al., 2015) ile pek çok farklı dil işleme görevi için oldukça etkili sonuçlar elde etmiş bir derin öğrenme modeli olan Transformer tabanlı modellerden BERT (Bidirectional Encoder Representations from Transformers), RoBERTa ve ALBERT gibi modeller (Devlin et al., 2018) ile XLM-R ve mBERT (multilingual BERT) gibi modeller (Conneau et al., 2019) özellikle VİT problemi özelinde yüksek performans gösteren örnekler olarak sayılabilir. (Chiu and Nichols, 2016) (Bolucu et al., 2019)

Derin öğrenme tabanlı yaklaşımlar, büyük miktarda veriyle önceden eğitilerler ve bu nedenle yüksek hesaplama gücüne ihtiyaç duyarlar ancak bu sayede dile ilişkin daha kapsayıcı bir model edilmiş olur ve bu modeller çoğu

durumda klasik VİT yaklaşımlarına göre çok daha yüksek doğruluk oranlarına sahip olurlar. (Gungor et al., 2018) (Şeker and Köse, 2020) (Türkoğlu and Iscan, 2021)

VİT probleminin zorlukları arasında metin içindeki isimlerin farklı varyasyonlarını ele almak, belirsizlikleri çözmek ve yanlış tanıma oranını en aza indirmek yer alır. Örneğin, "Amazon" kelimesi hem ormanlara hem de e-ticaret devine atıfta bulunmak için kullnılmaktadır. Bu nedenle bir metin içerisinde bu kelime ile karşılaşıldığında, doğru etiketlemeyi yapabilmek için kelimenin geçtiği bağlama uygun anlamın da doğru bir şekilde çıkarılabilmesi gereklidir.

Bu çalışmada, VİT probleminin çözümü için Transformer tabanlı modellerden birisi olan BERT modelinde uygulanan yaklaşımın Türkçe içerikli metinler için daha verimli ve başarılı bir şekilde çözüme ulaştırılabilmesi için Türkçe'ye özgü öznitelikleri dikkate alacak şekilde geliştirilmesi yöntemi seçilmiştir.

Diğer doğal dil işleme problemlerinde olduğu gibi, literatürde üzerinde yaygın şekilde çalışma yapılmış diller arasında yer alan İngilizce gibi dillerde, VİT görevi için de yaygın çalışmalar mevcuttur. Bununla birlikte Türkçe, Çekçe, Macarca, Fince gibi morfolojik açıdan da zengin olarak nitelendirilen dillerde VİT görevine yönelik çalışmalar sınırlı seviyede kalmaktadır. Türkçe'de hem VİT'e konu olan kelime türleri hem de bu kelimelerin içinde geçtiği bağlamda yer alan diğer kelimeler genellikle eklerle birlikte kullanılmakta ve kullanılan eklerin tipine göre çeşitli ses dönüşümleri ile kelimelerin biçimlerinde çeşitli değişiklikler (ulama, ünsüz yumuşaması/sertleşmesi, ünsüz düşmesi, ünlü daralması vb.) meydana gelebilmektedir. Sonuç olarak VİT görevi Türkçe gibi sondan eklemeli ve morfolojik açıdan zengin diller için hala açık bir araştırma alanıdır. (Çelikkaya et al., 2013) (Demir and Ozgur, 2014) (Eken and Tantug, 2015) (Eryiğit, 2016) (Küçük et al., 2017)

2.3.1 VİT için kullanılan etiketleme şemaları

Daha önce de belirttiğimiz gibi VİT çalışmalarında amaç varlık isimlerini etiketlemektir. Bu kapsamda kullanılan temel etiketler 'PER' (kişi) ve 'LOC' (konum) gibi tür etiketleridir. Bununla birlikte bir varlık isminin birden fazla kelimeden oluşması durumu oldukça yaygın karşılaşılan bir durumdur. Bu durumlarda etiketlenecek kelimelerin daha kapsamlı bir bilgi oluşturacak şekilde etiketlenmesi için kelimenin varlık isminin neresinde yer aldığı bilgisinin de temsil edilmesi önem taşıyabilmektedir. Hem bu ihtiyacı karşılamak hem de VİT çalışmalarının başarımlarının daha kolay bir şekilde karşılaştırılabilmesi ve herkes tarafından kullanılabilen standart veri kümelerinin oluşturulabilmesi için çeşitli standartlaşmış etiketleme şemalarının kullanımı zaman içinde yaygın hale gelmiştir. (Tjong Kim Sang, 2002) (Tjong Kim Sang and De Meulder, 2003)

BIO ve BILOU kodlama şemaları, Named Entity Recognition (NER) gibi görevlerde kullanılan iki farklı etiketleme yöntemidir. Bu şema isimlerindeki B,I,O,L,U harfleri "başlangıç" (B-Beginning), "içinde" (I-Inside), "dışında" (O-Outside), "sonuncu" (L-Last) ve "birim" (U-Unit) kelimelerinin ingilizcedeki karşılıklarının ilk harflerinden oluşur. Bu harfler etiketlenen bir metin parçasının bir varlık isminin içinde (I), dışında (O), başlangıcında (B) vb. olduğunu belirtmek için önekler olarak kullanırlar. Kullanılan etiket şemaları, VİT çözümlerinin geliştirilmesinde ve sonuçlarının değerlendirilmesinde önemli bir rol oynar.

BILOU Kodlaması:

BILOU kodlaması, varlık isimlerini daha ayrıntılı bir şekilde belirlemeye olanak tanır. Özellikle birden fazla kelime içeren varlık isimlerinin tam olarak belirlenmesi için "L" etiketi kullanılır. Ayrıca, tek kelime içeren varlık isimlerini ayırt etmek için "U" etiketi kullanılır.

B (Beginning): Bir varlık isminin başladığını gösterir. İlk kelimeyi belirtir.

I (Inside): Varlık isminin devam ettiğini gösterir. İlk ve son kelime hariç, varlık içindeki diğer kelimeleri belirtir.

L (Last): Varlık isminin son kelimesi olduğunu gösterir. Varlık ismi birden fazla kelime içeriyorsa son kelimeyi belirtir.

O (Outside): Herhangi bir varlık içermeyen kelimeler için kullanılır.

U (Unit): Tek kelimelik varlık isimlerini belirtir.

BIO Kodlaması:

Bu kapsamda kullanılan ve BILOU kodlamasına göre daha az detay barındıran bir şema da BIO kodlama şemasıdır. BIO kodlaması temel olarak varlık isimlerini belirlemek için başlangıç ve devam etme bilgisini verir. Ancak, tek kelime içeren varlık isimlerini ayırt etmek ya da varlık isimlerindeki son kelimeyi ayırt etmek için özel bir etiket içermez.

B (Beginning): Bir varlık isminin başladığını gösterir. İlk kelimeyi belirtir.

I (Inside): Varlık isminin devam ettiğini gösterir. İlk kelimeyi hariç, varlık içindeki diğer kelimeleri belirtir.

O (Outside): Herhangi bir varlık içermeyen kelimeler için kullanılır.

Örneğin, "Ahmet Necdet SEZER" adlı kişinin adının etiketlenmesinde "Ahmet" kelimesi için "B-PER" (başlangıç), "Necdet" kelimesi için "I-PER" (içinde) etiketleri kullanılabilir. Bu varlık ismi içinde geçmeyen "konuştu" gibi herhangi bir kelimenin etiketlenmesin de ise "O" (dışında) etiketi kullanılır.

```
BILOU  => BIO
"L"    => "I"
"U"    => "B"
```

Şekil 2.4: BILOU kodlamasından BIO kodlamasına dönüşüm kuralları

BILOU kodlamasından BIO kodlamasına dönüşüm yapmak için diğer

etiketlerde herhangi bir işlem yapmadan sadece derlemdeki "L" etiketlerinin "I" olarak, "U" etiketi "B" olarak dönüştürülmesi yeterli olmaktadır. Biz de bu çalışmada ana referans olarak BIO şemasını kullandık ve gereken VİT etiketleri için Şekil 2.4'de kuralları gösterilmiş olan dönüşümü uyguladık.

2.4 Bağlam ve Anlam İlişkisi

Doğal dil anlama konusunun da içinde ele alındığı disiplin olan doğal dil işleme alanı, tam anlamıyla yapısal olmayan bir veri olan doğal dil verisini girdi veya çıktı olarak üreten yöntem ve algoritmaların tasarlandağı tek düze çözümleri mevcut olmayan pek çok probleme ev sahipliği yapan bir çalışma alanıdır. Doğal dilin günlük yaşamımızda çoğunlukla pek farkında olmadığımız ama onu bilgi işlemsel açısından oldukça zorlu bir çalışma alanı olmasında büyük etkisi olan önemli özelliklerinden birisi de, doğal dilin oldukça belirsiz (ambigious) olmasıdır. Bu durumu bir cümlenin farklı bir dile çevrilmesi sırasında çokça alternatifin mevcut olduğu durumlarda somut olarak görebiliriz. Örnek TR: "Yapabilirsin" İNG-1:"You can you do it." İNG-2:"You may do it."

Belirsizliğin temel kaynağı bir kelimenin ya da ifadenin farklı bağlamlar içinde farklı anlamları ifade edecek şekilde kullanılıyor olmasıdır. Bununla birlikte gramer kurallarından veya dillerin mimarisinden kaynaklı belirsizlikler de olabilir. Metin tabanlı DDİ çalışmalarında bağlam genellikle bir doğal dil sembolünün önünde ve ardında bulunan yazılı diğer doğal dil sembollerinden oluşacak şekilde seçilmektedir. Ancak çalışma alanına göre bağlamın ses, vurgu, görüntü veya benzeri ilave gerçek dünya bilgilerini içerecek şekilde genişletilmesi de mümkün olabilir. DDİ problemlerinde, bağlamsal kapsamın ne kadar zengin ve geniş bir çerçevede ele alınacağı hem problemimizin karmaşıklık seviyesini hem de çözüm önerimizin başarımını etkileyecek en önemli konulardan birisidir. (Altintas et al., 2005) (İlgen et al., 2013)

Az önce tanıştığınız bir kişi ile vedalaşmadan hemen önce "Bana cebini verir misin?" şeklinde bir cümle kullandığınızı düşünelim. Bu cümlede geçen "cep" kelimesinden kastedilen şeyin aslında "cep telefonu numarası" olduğunu

günümüzde yaşayan her hangi bir kişi rahatça anlayacaktır. Bu cümleyi ingilizceye tercüme ettiğimizi düşünelim. Doğru tercüme şekli "Can you give me your pocket?" değil "muhtemelen Could you give me your mobile?" veya "Could you give me your mobile number?" şeklinde olmalıdır.

Türk Dil Kurumunun Güncel Türkçe Sözlüğünde “cep” kelimesinin olası karşılıklarına baktığımızda aşağıdaki açıklamaları görmekteyiz:

1. Genellikle bir şey koymaya yarayan, giysinin belli bir yeri açılarak içine yerleştirilen astardan yapılmış parça: “Bayramın her günü gelirler, ellerini ceplerine sokarak dolaşırlardı.” - Ayla Kutlu

2. Trafiği kolaylaştırmak, araçların durabilmesine olanak sağlamak için yaya kaldırımları veya şehirler arası yolların kenarlarında bulunan taşıt yanaşma yeri.

3. Cep telefonu: “Seninle yarın cepten konuşuruz.”

4. Savaş alanının bir yerinde düşmanın geriletilmesiyle ortaya çıkan taktik durum, çökertme.

5. Otomobil yarışlarında arabalarının yarışa başladıkları nokta.

Sözlükteki olası anlam karşılıklarını incelediğimizde bizim örnek cümlemizde ifade edilmek istenen anlama en yakın karşılığın 3’üncü sırada yer aldığı ancak yine de “Cep telefonu” anlamının “Cep telefonu numarası” anlamını tam olarak karşılamadığı görülecektir. Dolayısıyla aslında dil günlük yaşamda pek çok durumda sözlük anlamlarının yetersiz kalabildiği bir mecradır. İşte bu örnekte olduğu gibi, cümlede “cep” kelimesi ile ifade edilmek istenen gerçek anlamın ne olduğunun bilgisayarlar ile otomatik olarak tespit edilmesi ve bu örnekte olduğu gibi başka bir dile doğru anlamıyla otomatik olarak tercüme edilebilmesi günümüzde tekdüze algoritmalarla çözülemeyen karmaşık problemlere güzel bir örnektir.

Bir dil unsurunun anlamına etki eden diğer çevre unsurlara bağlam denilmektedir.

Aşağıdaki örneklerde kalın yazılan kelimeler farklı bağlamlarda kullanılmıştır:

- İşte bu içten tavırları sayesinde herkesçe kısa sürede sevilmişti.

- İçten yanmalı motorun icadı modern araba üretiminin önünü açtı.

Bu iki örnekte "içten" kelimesi ile ifade edilmek istenen farklı anlamları (1: Samimi, yürekten, candan, 2: Oyuk şeylerin boşluğu.) kelimenin içinde kullanıldığı bağlam sayesinde kolaylıkla anlarız.

Hem bu örnekte hem de bir önceki örneğimizdeki "cep" kelimesin görüldüğü üzere, dil verisi içerisinde yer alan ifade unsurlarının, kullanıldıkları yere ve zamana göre, çevrelerinde bulunan diğer unsurlar nedeniyle farklı anlamlar kazanması durumu söz konusudur.

Örnek olarak kullandığımız aynı cümlenin ("Bana cebini verir misin?") daha cep telefonunun icat edilmemiş olduğu 1900'lü yıllarda kullanıldığını hayal edin. Günümüzde hiç yadırgamadığımız bu cümleyi, o çağlarda yaşayan herhangi bir kişi çok anlamsız bulacaktır. Bu durum doğal dilin önemli özelliklerinden birinin de aynı canlı bir organizma gibi sürekli olarak değişen ve gelişen bir yapıya sahip olması olduğunun açık bir göstergesidir. İnsan zekasının kıvrak örneklerini içeren, teşbih, nükte, metafor ve benzeri söz sanatlarında ise daha büyük zorluklarla karşılaşılacağı ortadadır. Aşağıda metafor kullanımına çeşitli örnekler verilmiştir:

- Konuşurken boğazım düğümleniyordu.

- Gözlerimin içi gülüyordu.

- Sözleri insanın yüreğini yakardı.

- Onu görünce, yüreğim yerinden hopladı.

Literatürde bağlam ve anlam ilişkisinin önemini vurgulayan pek çok çalışma mevcuttur. Örneğin Mikolov kelime temsillerinin vektör uzayında dağıtılmış olarak nasıl temsil edileceği ve bu temsillerin nasıl bir araya

getirilebileceği üzerine çalışmıştır. (Mikolov et al., 2013b) Kelimelerin vektörler halinde temsil edilmeye başlanması, bağlam ve anlam ilişkisini temsil eden bilgileri daha etkin bir şekilde yakalama ve bu bilgileri yapay sinir ağlarına girdi olarak kullanabilme konusunda bir dönüm noktası oluşturmuş önemli bir çalışma niteliğindedir. (Bojanowski et al., 2016) Bu maksatla dilin sentaks analizi ile ve temsil tabanlı modelleri bir araya getiren birleşik model çerçeveleri de oluşturulmuştur. Bu modellerde sentaktik ağaç yapıları ve dağıtımsal semantik temsilleri 2 ayrı aşamada ele alıp birleştirerek kullanan dilin yapısını ve anlamını daha iyi bir şekilde birleştirerek daha kapsamlı bir dil anlama yeteneği sağlanabilmektedir. (Bowman et al., 2015)

Bağlamsal kelime temsil modellerinin derin öğrenme teknikleri ile birlikte ele alan çalışmalar arasında belirli bir bağlamı dikkate alan kelime temsilleri oluşturmaya odaklanılan çalışmalar da mevcttur. Aynı kelime farklı bağlamlarda farklı anlamlara gelebildiğinden, derin sinir ağlarına dayalı dil modellerini kullanan çalışmalarda kelime temsillerini dilin yapısını ve örüntülerini de kapsayabilecek seviyede zenginleştirebilmek için metin tabanlı büyük derlemlere ihtiyaç duyulur. (Collobert et al., 2011) Böylece kelime anlamlarını temsil edebilmek için daha geniş bir bağlam uzayı dikkate alınabilir hale gelir. Sonuç olarak elde edilen derin bağlamsal kelime temsillerine dayalı dil modelleri ile doğal dil işleme görevlerindeki performansın genel olarak artırıldığı gösterilmiştir. (Peters et al., 2018) Derin bağlamsal kelime temsillerini kullanarak doğal dil işleme alanında bağlam ve anlam ilişkisi konusunda önemli katkılar sağlamış, birçok dil anlama görevinde başarıyla kullanılmış ve konuyla ilgili araştırmacılar arasında geniş çapta kabul görmüş pek çok çalışma mevcuttur. (McCann et al., 2017) (Peters et al., 2018) (Akbik et al., 2018) (Devlin et al., 2018).

2.5 Doğal Dil İşlemede Yapay Sinir Ağları ve Derin Öğrenme Modelleri

Doğal dil işleme, dilbilgisel yapıyı anlamayı ve doğal dil metinlerini işlemeyi hedefleyen bir disiplindir. Yüzeysel bakıldığında belli kural ve örüntü-

lerin göze çarptığı ancak detaylar ve derinlere inildiğinde genel geçer kurallar bulunamayan çok fazla alana sahip karmaşık bir problem olan doğal dil işleme alanında yapay sinir ağlarının kullanılması yeni bir yaklaşım değildir. Bu yaklaşım, metin sınıflandırması, kelime tahmini, anlamsal benzerlik ölçümü gibi doğal dil işleme görevlerinde başarıyla uygulanmıştır. (Goldberg, 2017) Bununla birlikte geçmişte bu yaklaşım ile yapılan çalışmalar çok yaygın bir etkiye sahip olmamıştır.

Yapay sinir ağlarının temelde veri setindeki özellikleri öğrenerek, veri setine ait bir sınıflandırma veya tahmin yapmayı sağlayan modeller olduğu söylenebilir. Doğal dil işleme çalışmalarında da yapay sinir ağları ile doğal dil verilerinde bulunan anlamlı ilişkilerin öğrenilmesi amaçlanabilir. Örneğin, bir doğal dil işleme çalışmasında, yapay sinir ağları kullanılarak, metinlerdeki anahtar kelimelerin belirlenmesi ve metinlerin sınıflandırılması gibi görevler gerçekleştirilebilir. (Kim, 2014) Ancak, yapay sinir ağları uygulamaları içeren diğer alanlarda olduğu gibi, doğal dil işleme çalışmalarında da yapay sinir ağlarının ve derin öğrenme yöntemlerinin etkin bir şekilde kullanılabilmesi için, yeterli miktarda ve nitelikli veri setine sahip olunması çok önemlidir. (Yang et al., 2016)

Derin öğrenme, makine öğrenmesinin güncel ve popüler bir dalı olmakla birlikte aslında yıllar önce ortaya çıkmış olan yapay sinir ağları yaklaşımının gelişen teknoloji sayesinde artan işlemci gücü ve veri hacmi imkanlarından daha fazla yarar sağlayacak şekilde yeniden şekillendirildiği ve klasik yapay sinir ağlarından farklı olarak katman çeşitliliği ve derinliğinin artırıldığı böylece çok daha derin ve karmaşık sinir ağı modelleri oluşturulabilen bir modelleme yaklaşımıdır (LeCun et al., 2015). Bu nedenle bu tarz modeller zaman zaman derin sinir ağları olarak da adlandırılmaktadırlar. Zaten derin öğrenme adı, farklı işlevlere sahip birçok yapay sinir ağı katmanının arka arkaya eklenmesi ile eskiye göre çok daha derin yapay sinir ağları oluşturulmuş olmasından kaynaklanmaktadır. (Schmidhuber, 2015)

Derin öğrenme tabanlı doğal dil işleme yöntemlerinin hızla geliştirilmesi ile birlikte büyük veri setlerindeki dilbilgisel özelliklerin etkin bir şekilde

temsil edilmesi imkanı ortaya çıkmıştır ve metin sınıflandırma, duygu analizi, dil çevirisi, konuşma tanıma gibi pek doğal dil işleme probleminde büyük sıçramalar elde edilmiş ve bu sonuçlara bağlı olarak da doğal dil işleme çalışmalarında yapay sinir ağı ve derin öğrenme modelleri kullanım oranı çok büyük oranda artmıştır. (Mikolov et al., 2013b) (Kim, 2014) (Vaswani et al., 2017) (Young et al., 2018) (Devlin et al., 2018)

Diğer yapay sinir ağı yaklaşımlarında olduğu gibi derin öğrenme yaklaşımında da giriş ve çıkış katmanlarındaki verilerin doğru şekilde temsil edilmesi büyük önem taşımaktadır (Bengio et al., 2013). Her iki yaklaşımda da doğal dil sembollerinin yapay sinir ağının giriş katmanını verimli bir şekilde besleyecek uygun şekilde temsil edilmesi için çeşitli dönüşümlerden geçmesi gerekmektedir.

Derin veya normal bir yapay sinir ağı modelini eğitirken, bu dönüşüm ihtiyacı karşılamak üzere, giriş ve çıkışlarda her bir kelimeye (veya dil sembolüne) karşılık gelen ayrı bir giriş/çıkış düğümü belirleyip bu düğümün değerini '1', diğer tüm düğümlerin değerini de '0' yapabiliriz. Farklı sınıfları bu şekilde temsil etmeye bire-bir kodlama (one-hot encoding) denilmektedir. Fakat herhangi bir doğal dilde on binlerce kelime olduğundan bu yöntem geniş bir sözlük dağarcığımız olması gerektiğinde çok verimli olmamaktadır. Yine de eğer sınıflandırma işlemi sonucunda elde edilecek farklı etiketlerin sayısı çok fazla değilse çıkış katmanında bire-bir kodlama kullanılabilir. Bu durumda yapay sinir ağı modelinden bir tahmin alırken çıkışta en büyük değer hangi rakama karşılık geliyorsa cevabı o doğru cevap olarak kabul edebiliriz.

Yapay sinir ağlarının giriş düğümlerinde dil sembollerini (kelime, harf vb.) girdi olarak kodlarken, bire-bir kodlama kullanılması basit bir yaklaşım olmakla birlikte çoğunlukla verimli olmadığından, genellikle dil sembollerine karşılık gelen temsil (embedding) modelleri kullanılır.

Kelime temsil (word embedding) modelleri, kelimenin kendisini bir bütün olarak ele alıp, kelimelerin bağlamları, metin içerisinde kelimelerin birbirine göre konumları, kelimelerin aynı bağlam içinde bir arada geçme sıklıkları

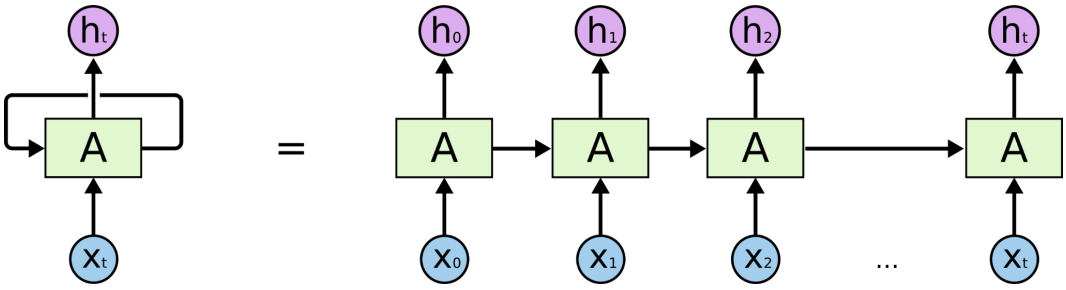
gibi bilgileri temsil edecek şekilde eğitilebileceği gibi kelimeleri kelime altı bileşenlerine indirgeyerek ve bu bileşenlerin bir arada geçmesini baz alacak şekilde tasarlanabilirler.

2.5.1 Recurrent Neural Network (RNN)

Temel sinir ağlarında tüm girdilerin (ve çıktıların) birbirinden bağımsız olduğu varsayılabilir durumda ise de, doğal dil işlemenin pek çok alanı için bu yaklaşım uygun değildir. Zira bir cümlede bilgiler ardışık olarak sıralanmış kelimeler şeklinde geçer ve kelime sırasının değişimi anlam değişimine de neden olabilir.

İnsanlar olarak bizler de bir konu hakkında düşünmeye hiç bir zaman sıfırdan başlamayız. Örneğin siz bu makaleyi okurken ve her bir kelimeyi anlamlandırırken daha önceki kelimeleri ve cümleleri de dikkate alıyorsunuz. Hatta bir cümleyi okuduktan sonra okuyacağınız bir cümle önceki cümleden anladığınız şeyi pekiştirebilir veya değiştirebilir.

RNN'lerin arkasındaki temel fikir de, gerçek dünyada ardışık (zaman akışı gibi) olarak bulunan veriyi verimli bir şekilde işleyebilmektir.

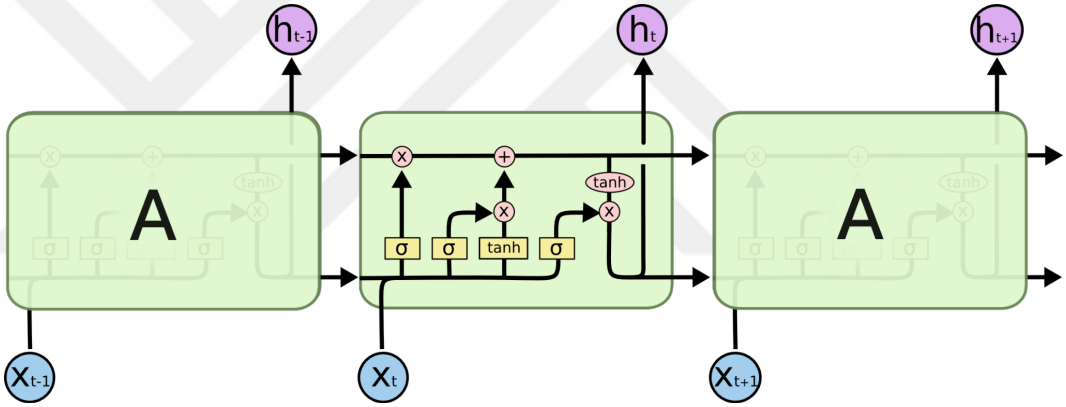


Şekil 2.5: RNN Gösterimi

RNN'lerin "yinelemeli" (recurrent) olarak isimlendirilmesinin nedeni, her düğümün kendi içinde bir geri dönüş içeriyor olmasıdır. Başka bir deyişle her bir düğümün çıkışının bir sonraki iterasyonda aynı düğüm için ayrı bir girdi olarak kullanılmaktadır. Böylece her iterasyondaki sonucun daha önceki hesaplamalara da bağlı olması sağlanır. RNN düğümleri bu sayde o zamanda kadar hesaplamış oldukları tüm çıktıları belli oranda depolayabilen kısmen kalıcı bir bir "hafıza"ya sahiptirler.

2.5.2 Long Short Term Memory (LSTM)

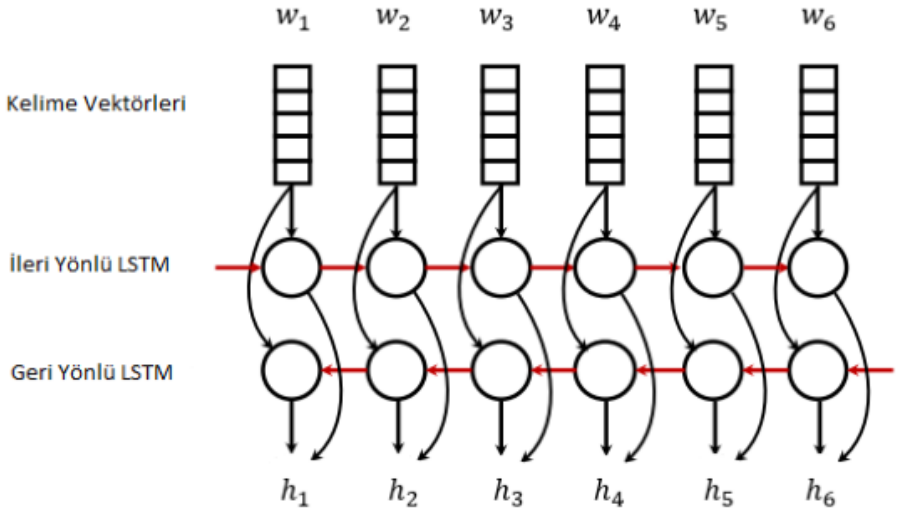
RNN'lerin yukarıda anlatılan yapısı teorik bir sınır içermemekle birlikte, pratikte onların sadece birkaç iterasyon öncesini hesaba katmasına imkan sağlayabilmektedir. Örneğin bir bağlamdaki kelimeler (yani iterasyonlar) arasındaki mesafe uzadıkça RNN'lerin kalıcılık özelliği yetersizleşmektedir. Oysa uzun bir cümlede yüklem ile özne arasında onlarca kelime olabilir. Bu problemi çözebilmek için geliştirilmiş bir RNN tipi olarak LSMT'ler ortaya çıkmıştır. Yapılan çalışmalar, LSTM modelinin metinlerdeki uzun bağımlılıkları modelleme ve diğer görevlerde başarılı olduğunu göstermektedir (Hochreiter and Schmidhuber, 1997). Doğal dil işleme problemlerinde de etkili bir şekilde kullanılmış olan LSTM mimarisi derin öğrenme alanında uzun bağımlılıkların modellenmesi için önemli bir yapı taşı olarak tanımlanabilir.



Şekil 2.6: LSTM Gösterimi

LSTM'ler, RNN'lere göre çok daha uzun vadeli bağıntıları öğrenebilen özel bir RNN türevidir ve bu nedenle, uzun kelime dizileri içeren doğal dil işleme görevleri için uygundurlar. Yukarıdaki sinir ağı düğümünü bir yapay sinir ağı hücresi olarak ele alalım. Uzun vadeli bağıntıların öğrenilebilmesini kolaylaştıran temel özellik LSTM hücresinin, durumunun korunmasını sağlayan bir bileşene sahip olmasıdır. Yukarıdaki gösterimde üst tarafta görülen yatay çizgi bu hücre durumunu simgeler. Hücre durumu her hücre için ayrı ayrı tutulan ve hücrenin girdileri (hücrenin bir önceki iterasyondaki çıktısı da bu girdilere dahildir) tarafından belli operatörlerle değiştirilebilen kalıcı bir değerdir.

Bir LSTM hücresinin çalışması sırasında adım adım olarak; önceki hücre durumunun ne kadarının korunacağı hesaplanır, hücre durumu değerinde yapılacak değişimin hesaplanır, eski hücre durumunun korunan kısmı ile meydana getirilecek değişiklik birleştirilerek yeni hücre durumu değeri hesaplanır, güncel hücre çıkış değerinin (bir önceki hücre çıktısı dahil hücre girdilerine ve hücre durumu değerine bağlı olarak) hesaplanması şeklinde gerçekleşir.



Şekil 2.7: Bi-LSTM Gösterimi

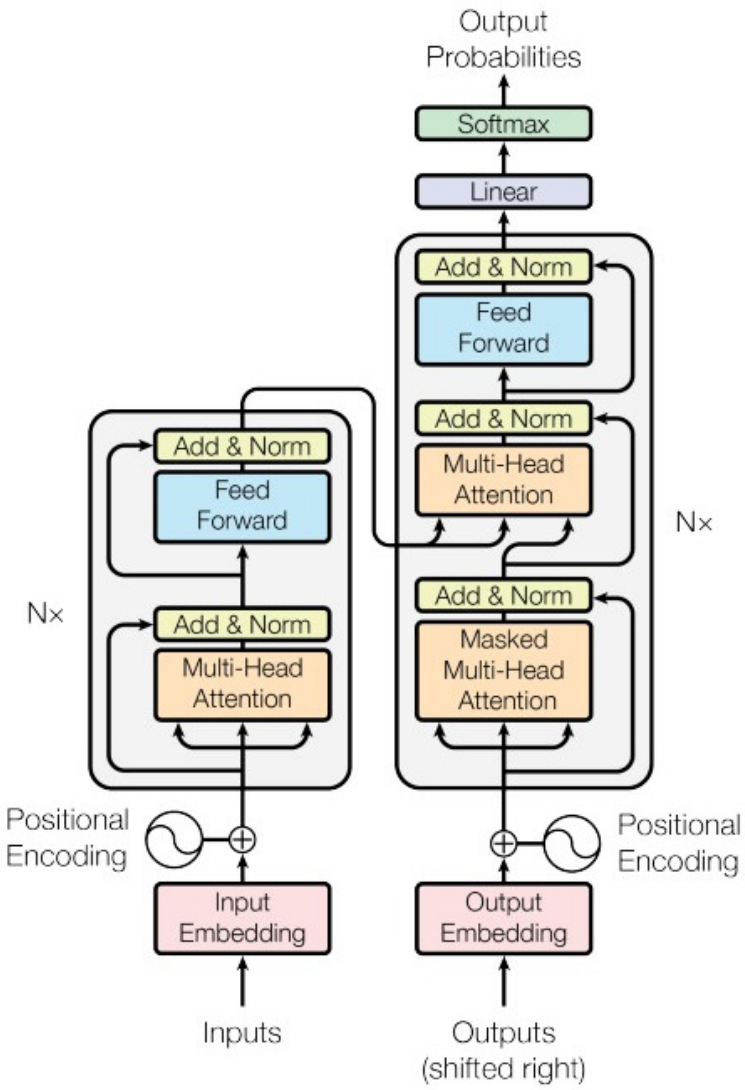
Son zamanlarda LSTM'ler kullanılarak özellikle konuşma tanıma, dil modelleme, çeviri vb. doğal dil işleme alanlarında başarılı sonuçlar elde edilmiştir. Hücre durumuna ilişkin bilgi akışının sadece tek yönlü (ileriye doğru) değil, çift yönlü (hem ileriye hem de geriye doğru) olduğu LSTM modellerine de Bi-LSTM denilmektedir. Böyle bir modele örnek aşağıda gösterilmiştir. Doğal dil uygulamalarında Bi-LSTM modelleri genel olarak LSTM modellerinden daha başarılı sonuçlar vermektedirler.

2.5.3 Transformer Modeli

Transformer modeli 2017 yılında ortaya çıkmış bir Derin Öğrenme modeli olup doğal dil işleme görevlerinde önemli rol oynamış bir modeldir. Transformer yaklaşımı geleneksel dil işleme modellerinde yaygın olarak kullanılan RNN ve LSTM gibi yapılar yerine tamamen dikkat mekanizmasına dayalı bir model kullanılmasını önerir. Transformer modeli genel olarak metin içerisindeki ilişkileri daha iyi yakalamak için farklı pozisyonlardaki kelimeler arasındaki

dikkat ağırlıklarını hesaplayarak çalışır. (Vaswani et al., 2017)

Modelin ana fikri öz-dikkat (self-attention) mekanizmasınının derin sinir ağı modeli içindeki katmanlarda kullanılmasına dayanır. Öz-dikkat mekanizması girdi veri dizisi içindeki farklı elemanlarının işlemin sonucuna etkisini farklı seviyelerde ağırlıklandırırak işleme alan bir mekanizmadır.



Şekil 2.8: Transformer modeli gösterimi (Vaswani et al., 2017)

Öz-dikkat mekanizması temelde girdinin sadece uygun ve gerekli kısımlarına odaklanılmasını sağlar böylece kelime temsilleri arasındaki uzun vadeli bağımlılıkların önemini kaybetmesinin (vanishing gradient) önüne geçebilir. Dikkat (attention) katmanlarındaki ağırlık değerleri de diğer sinir ağı modellerinde olduğu gibi back propagation yöntemi ile hesaplanır. Dikkat mekanizma-

sının temelde her bir kelimenin diğer kelimelerle etkileşimini hesaplayarak ve metnin farklı bölgeleri arasındaki bağıntıları yakaladığı düşünülebilir. Dikkat mekanizması paralel işlemeye uygun yapısı sayesinde uygun donanımlarla eğitim sürecini hızlandırmaya böylece eğitim kalitesini de yükseltmeye imkan sağlamıştır. (Chen et al., 2018)

Transformer modeli içerisinde öz-dikkat mekanizmasına ilave olarak iki ana temel bileşen olarak kodlayıcı (encoder) ve çözücü (decoder) bileşenleri de yer almaktadır. Kodlayıcı, metni temsil eden vektörleri oluştururken, çözücü ise bu vektörleri hedeflenen görevi çözüme kavuşturmak için kullanır. Dikkat mekanizması, her bir kelimenin diğer kelimelerle etkileşimini hesaplamak için kullanılır ve metnin farklı bölgeleri arasındaki bağlantıları yakalar. Transformer modeli ayrıca hesaplama açısından etkin bir şekilde paralel işlemeye uygun bir modeldir. Yeterli sayıda ve uygun donanımlar (GPU, TPU vb.) kullanıldığıda bu özelliği ile toplam eğitim süresini ciddi şekilde kısaltmaktadır.

BERT dahil transformer modeline dayalı olarak geliştirilmiş benzer pek çok farklı model doğal dil işleme ve anlama görevlerinde kullanılmış ve yüksek performanslar göstermişlerdir. (Radford et al., 2018) (Devlin et al., 2018) (Liu et al., 2019b) (Yang et al., 2019).

2.5.4 BERT Modeli

BERT (Bidirectional Encoder Representations from Transformers) Google tarafından geliştirilip 2018 yılında duyurusu yapılan, Transformer modelini baz alan güçlü bir dil modeli modelidir. BERT metin girdilerini hem ileri (soldan-sağa) hem de geri (sağdan-sola) yönde kodlayarak metine dair bağlam modellemesini çift yönlü olarak oluşturur. Transformer modelinin Encoder parçasından faydalanan BERT modelinin sıfırdan eğitilmesi için büyük ölçekli metin derlemlerine ve yüksek hesaplama gücüne ihtiyaç duyulur. Bu nedenle önceden eğitilip kaydedilmiş modellerin ince ayar eğitimleri ile farklı görevlerde kullanılması şeklinde bir yaygın kullanım şekli oluşmuştur. BERT, genel olarak pek çok doğal dil anlama görevlerindeki gelişme seviyesini (State-of-the-Art) bir üst basamağa taşıyan sonuçlar elde etmiş ve önceden eğitilmiş bir

model olarak da çeşitli uygulamalarda başarıyla kullanılmıştır. (Devlin et al., 2018) (Koroteev, 2021) BERT ortaya koyduğu başarılı sonuçlar nedeniyle çok popüler hale gelmiş ve üzerine yapılan eklenti ve geliştirmelerle RoBERTa (Liu et al., 2019b), ALBERT (Lan et al., 2020), DistilBERT (Sanh et al., 2019), ELECTRA (Clark et al., 2020), XLNet (Yang et al., 2019) ve ERNIE (Sun et al., 2019) gibi pek çok türev model/yaklaşım ortaya çıkmıştır (Liu et al., 2019a).

BERT'in sıfırdan eğitim sürecinde iki ana görevi mevcuttur:

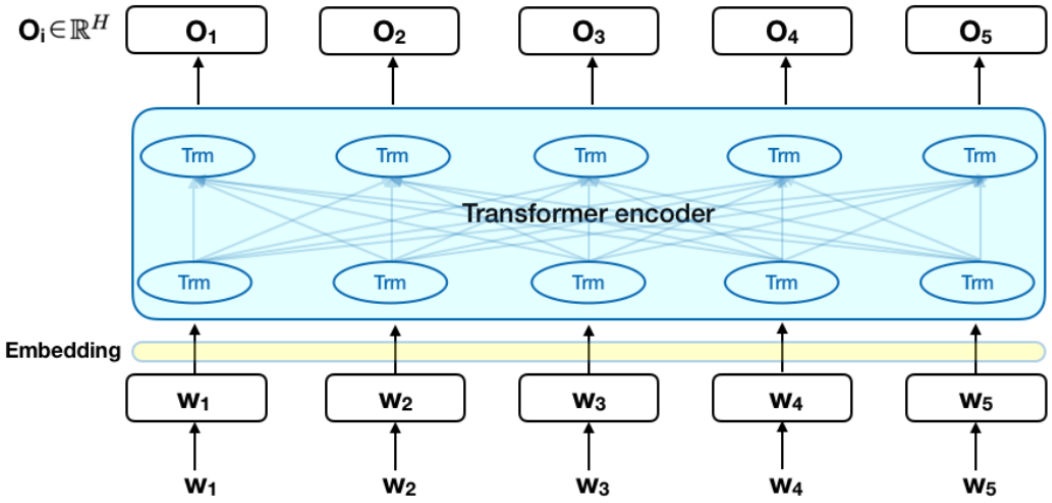
1. Maskeli Dil Modeli (MLM): Bu görevde metin girdisinde rastgele seçilen kelimeler maskelenir ve eğitilen modelin amacı, bu maskelenen kelimelerin ne olduğunu, maskelenmiş kelimeleri çevreleyen bağlamı temel alınarak tahmin etmektir.

2. Sonraki Cümle Tahmini (NSP): Bu görevde model, iki cümlenin orijinal metinde ardışık olarak görünüp görünmediğini tahmin etmek üzere eğitilir.

Google tarafından 800M kelime hazinesine sahip olan BookCorpus ve 2.5B kelime hazinesine sahip olan Wikipedia corpus kullanılarak farklı derinliklere sahip 2 tip BERT modeli (bert_large, bert_base) yayınlanmıştır.

Sıfırdan eğitim aşaması tamamlanmış bir BERT modeli, varlık ismi tanıma, metin sınıflandırma, soru-cevap, duygu analizi gibi pek çok farklı doğal dil işleme görevi için sonradan ince ayar eğitimine tabi tutulabilir. İnce ayar eğitimleri, önceden eğitilen BERT modelinin üzerine seçilen göreve özgü ilave bir katman ekleyerek ve göreve uygun şekilde etiketlenmiş veri seti kullanılarak BERT modelinin tekrardan eğitilmesi ve böylece modelin belirli bir göreve uygun çalışacak hale getirilmesi şeklinde uygulanır.

BERT'in bağlamsal bilgiyi başarılı bir şekilde yakalama yeteneği sayesinde pek doğal dil işleme görevlerinde önemli gelişmeler meydana gelmiş ve BERT modeli farklı uygulamalarda kullanılabilen temel bir model haline gelmiştir. Son yıllarda doğal dil işleme alanında önceden eğitilmiş (pretrained) BERT modellerinin daha spesifik doğal dil işleme görevleri için sonradan



Şekil 2.9: BERT modeli gösterimi (Devlin et al., 2018)

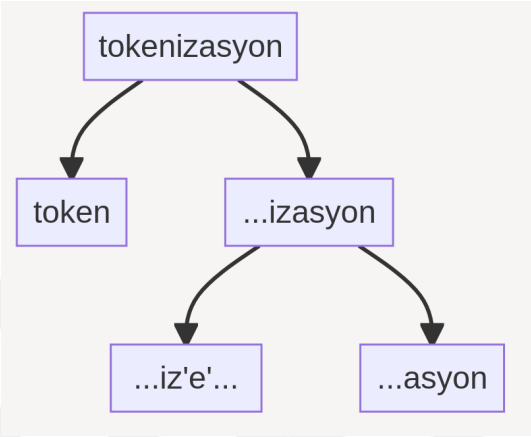
yapılan ince ayar uygulama (fune-tuning) eğitmelerinin başarımı artırmada önemli bir ivme sağladığını gösteren çok sayıda çalışma mevcuttur (Devlin et al., 2018) (Sun et al., 2021) (Liu et al., 2019b) (Aksakallı and Türk, 2019). BERT modellerinin ortaya koyduğu yüksek başarım, dünya genelindeki pek çok doğal dil işleme araştırmacısının, belirli bir dil veya belirli bir doğal dil işleme görevi üzerinde eğitilerek özelleştirilmiş BERT modelileri üzerinde araştırma yapmaya yöneltmiştir. (**?**) (Taher et al., 2020)

Bu çalışmada ise, BERT modelinin VİT başarımını artırmak üzere model girdilerinin dile özgü ilave özellikler ile geliştirilmesi maksadıyla temel alınan BERT modeli özelinde modelin iç yapısının ve çalışma mantığının dile özgü özellikleri daha iyi modelleyecek şekilde revize edilebilirliği kapsamında iyileştirme sağlanabilecek önemli bileşenlerden birisinin tokenizasyon aşaması olduğu kabul edilmiş ve temel model olarak evrensel bir dil modeli olarak hizmet edebilecek şekilde tasarlanan ve 104 dilden oluşan bir derlem üzerinde eğitilmiş olan "BERT-Base, Multilingual Cased" modelinden (Devlin et al., 2018) faydalanılmıştır.

2.6 Tokenizasyon

Tokenizasyon, bir metni içindeki kelime, kalıp veya örüntülerden oluşacak şekilde parçalara ayırma işlemine verilen isimdir. Tokenizasyon işlemi sonunda

bir metin içerisinde kelime, cümle ya da paragraf bazında olmak üzere doğal dil için anlamlı olan birimlerin (token) belirlenmesi ve bunların bir liste haline getirilebilmesi hedeflenir. Tokenizasyon, metinlerin daha kolay işlenebilir hale getirilmesini sağlayarak, metinlerin çözümlenmesi ve anlamlandırılması için gerekli olan bir adımdır. Bu adım çoğu zaman metinlerin makine öğrenmesi, derin öğrenme ve yaveri madenciliği gibi alanlarda işleme alınmasından önce yapılan bir ön işleme adımıdır.



Şekil 2.10: Tokenizasyon gösterimi

Herhangi bir Yapay Sinir Ağı veya DDİ Derin Öğrenme modeli üzerinde çalışma gerçekleştirebilmek için de öncelikle ham metin girdilerinin tokenizasyon işleminden geçirilerek, eğitilecek model tarafından anlaşılabilir bir temsile (token) çevrilmesi (eşleştirilmesi) gerekmektedir. Tokenizasyon sonrasında yapay sinir ağı açısından olabildiğince iyi şekilde işlenebilir ve anlamlandırılabilir birimlere (token) bölünmüş olan metinler sinir ağı modeline giriş olarak verilir.

Geleneksel bir yaklaşım olarak, metinler kelime düzeyinde işlenebilir ve kelime seviyesinde tokenizasyon yapılabilir. Ancak, dilden dile bu durumun ağırlığı değişmekle birlikte doğal dillerde kelime sınırları genellikle yeterince belirgin değildir veya kelime düzeyindeki ayrıştırma yeterli seviyede bir detay sağlayamıyor olabilir. Bu durumda, dile özgü özelliklerin daha iyi modellemesi için tokenizasyon aşamasının revize edilmesi fayda sağlayabilir (Domingo et al., 2018) (**?**).

Oxford sözlüğüne göre, token (Türkçe: jeton veya belirteç) “Bir gerçekliğin, ölçümün veya algının vb. somut ve görünür bir temsili olarak kullanılan herhangi bir şeydir”. Bu haliyle bile kulağa yeterince karmaşık gelen bu tanıma Doğal Dil İşleme perspektifinden yaklaştığımızda işler biraz daha karmaşık bir hal alır.

Genel bir yaklaşım olarak "token nedir" sorusunun yanıtını şöyle verebiliriz: Token, bir metin (harf dizisi veya katar) içinde tanımlanabilen (ya da tanınabilen) bir dil birimidir. Bir metnin token parçalarına ayrılması işlemine de tokenizasyon denilir. Çok daha genel bir yaklaşımla bir karakter dizisini (metin) parçalara bölme işlemine de tokenizasyon denilebilir. Ancak tokenizasyon işlemi sonucunda elde edilen parçaların dilbilimsel açıdan (veya en azından sonrasında işleme alınacağı süreçler açısından) olabildiğince anlamlı birimler (İng: Linguistically Meaningful Units - LMU) olması hedeflenmelidir.

Tokenizasyon süreciyle ilgili temel sorunlardan biraz bahsetmek gerekirse; tokenizasyon ilk bakışta çok basit bir konu gibi görünebilir, ancak doğal dilin belirsizliği ve dillerin kendine özgü nitelikleri nedeniyle aslında henüz kapanmamış bir çalışma alanı olduğunu söylemek kesinlikle yanlış olmayacaktır. (Dridan and Oepen, 2012) (Park et al., 2020) (Friedman, 2023)

Tokenizasyon konusunda temel olarak da kabul edilebilecek en basit yaklaşım, daha önce de örneklediğimiz gibi tüm metin girdilerini boşluklardan bölmek ve her kelimeye ayrı bir tanımlayıcı (token) atamak olacaktır. Ancak bu yaklaşım, kaynak metninizde bulunan her kelimeyi ayrı ayrı temsil ederek depolayacağı için kelime dağarcığımız büyüdükçe sorunlar artacaktır. Ayrıca "insan" ve "insanlar" gibi kelimeler, anlamları yakın anlama sahip olsa bile tamamen farklı tanımlayıcılar ile temsil edilecektir.

Kelimelerin zaten boşluklarla ayrılmış olması nedeniyle boşlukların ve noktalama işaretlerinin doğal bir ayraç görevi gördüğünü ve boşluk/noktalama işaretleri ile ayrılmış her bir kelimenin bir token olabileceğini düşünmek latin alfabesinin kullanıldığı çoğu dil için genel olarak yeterli bir yaklaşım gibi görünebilir. Bazi nadir durumlarda bir tokenin bir kelimeden oluşmasının

verimli sonuçlar verebileceği durumlar da mevcuttur ancak her kelimenin ayrı bir token olarak görülmesi genellikle çok verimli bir sonuç vermemektedir (Domingo et al., 2018).

Örneğin, basit yaklaşımla "Dün buraya geldi." cümlesinde tokenlar "dün", "buraya", "geldi" olarak hesaplanabilir. Ama aslında buraya kelimesi "bura" ve "-ya" parçalarından oluşacak, "geldi" kelimesi de "gel", "-di" parçalarından oluşacak şekilde de tokenlara ayrılabilir. Dolayısıyla her durum ve şartta tek bir doğru tokenizasyon yönteminin var olduğunu düşünmek hatalıdır. Hatta bir de bu cümlede gizli özne olarak "O" öznesinin bulunduğunu hatırladığımızda dildeki belirsizliklerin tokenizasyon dahil doğal dil işleme problemlerini nasıl zorlaştırabildiği daha rahat anlaşılacaktır. İlave olarak, birçok dilde bileşik isimler aralarında boşluk bulunmayacak şekilde birleşik yazılır. Birleşik kelimelerin birden fazla token ile ifade edilmesinin avantajlı olabileceği durumlar da mevcuttur (Hirschmann et al., 2016) (Nivre and Hall, 2008). Ayrıca Çince, Japonca gibi bazı dillerde de, kelimeler arasında boşluk bırakmak zorunlu değildir (Liu et al., 2022).

Tokenizasyon aşamasındaki zorlukları aşmak için literatürde dile özgü tokenizasyon yöntemleri geliştirilmesi, tokenizasyon aşamasında dilin özelliklerini daha iyi yansıtmayı hedefleyen, dilbilimsel kurallar veya dil özelliklerine dayalı ilişkilerin kullanılması ve kelime düzeyinden daha küçük birimlerden oluşan token'lara ayrıştırma yapılması ve kelime altı optimum parçaların belirlenmeye çalışılması gibi yaklaşımlar uygulanmıştır (Kazakov and Manandhar, 2000) (Sennrich et al., 2016) (Chitnis and DeNero, 2015b) (Kudo, 2018).

2.6.1 Kelime Altı Tokenizasyon

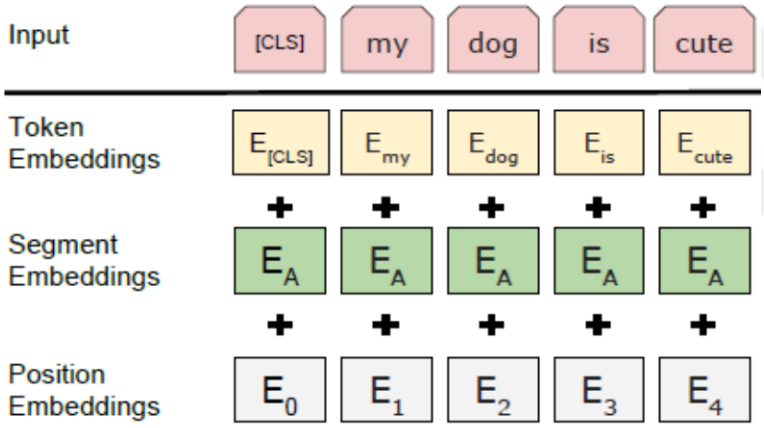
Önceki bölümde belirtilen sorunların üstesinden gelmek için geçmiş dönemlerde daha çok kelime bütünü seviyesinde kullanılan tokenizasyon teknikleri son dönemlerde daha çok temsil modellerinde kelime altı (ing: sub-word) bileşenlere de yer verilmesine yönelmiş ve bu şekilde başarımın da artırılabileceği görülmüştür. Çünkü bu tür eğitim algoritmaları, örneğin İngilizce derlemlerde sıkça yer alan"-ing", "-ed" gibi kelime altı tanımlayıcıları

rahatlıkla ortaya çıkabilmektedir.

Bizim çalışmamızın odak noktası Transformer modelleri arasında başarımda ciddi sıçramalara vesile olmuş olan BERT modeli olup, Transformer modellerinde yaygın olarak kullanılan üç temel tokenizasyon algoritması mevcuttur.

2.6.2 WordPiece Algoritması

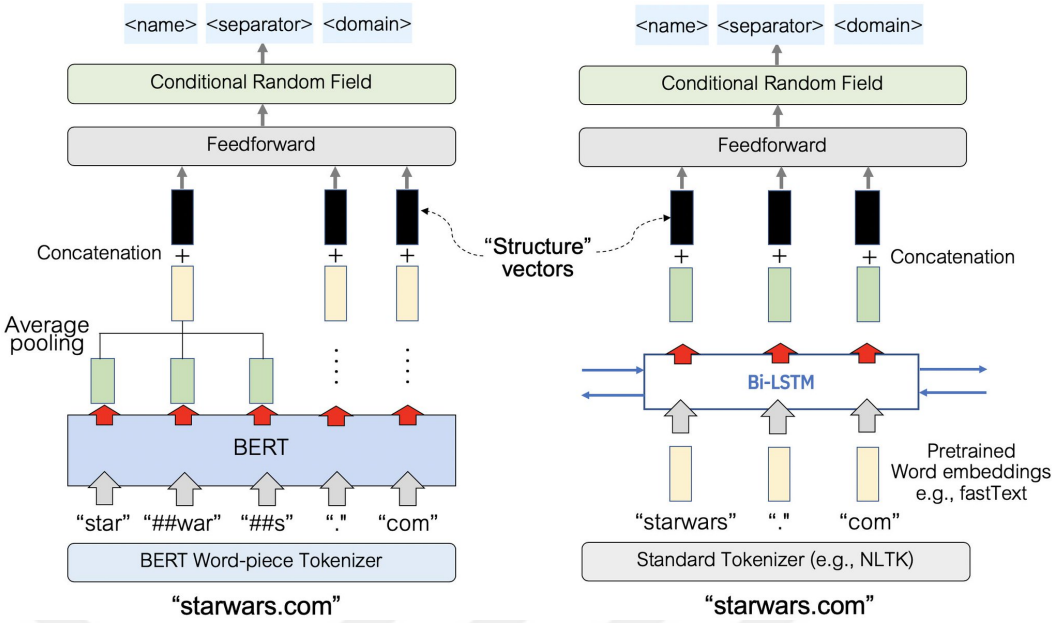
WordPiece algoritması ilk olarak, Google'da çalışan araştırmacılar tarafından Japonca ve Korece'ye uygulanmak üzere başarılı bir sesli arama sistemi oluşturma kapsamında yapılan AR-GE faaliyetleri sırasında yaşanan ve temelde bu diller için sınırsız olan kelime dağarcığıyla başa çıkmak üzere kullanılan tekniklerden birisi olarak ortaya çıkmıştır.



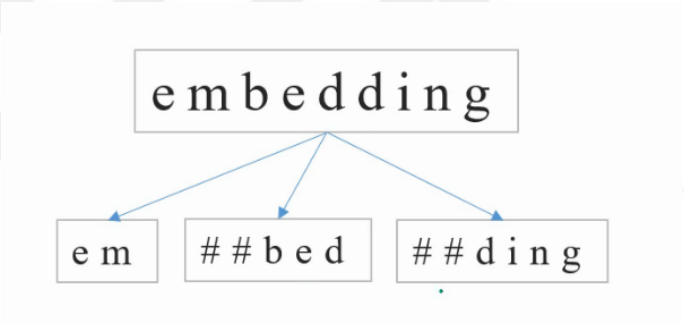
Şekil 2.11: BERT Modeline beslenen WordPiece gösterim katmanları

Yukarıdaki görselde orjinal makaleden alıntı yapılarak açıklanan algoritma temelde oldukça basittir (Wu et al., 2016). Ancak algoritmanın daha performanslı çalışması için daha sonradan geliştirilen implementasyonlarda (Devlin et al., 2018) çeşitli optimizasyonlar yapılmıştır.

Algoritma temelde bir dilde yer alan en temel bileşen olarak karakterleri baz almakta ve eğitim datası içinde bir arada geçme olasılığı en yüksek olan karakterleri bir araya getirilmiş kelime altı bileşenler (token) olarak belirleyerek token envanterini sürekli olarak artırmaktadır. (Sennrich et al., 2016) Belirli bir



Şekil 2.12: BERT ve Bi-LSTM modellerini beslenen token girdilerinin kıyaslanması



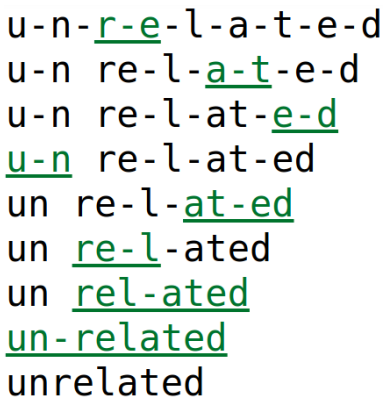
Şekil 2.13: WordPiece algortiması, örnek tokenizasyon

sayıda token'a ulaşıldığında veya envantere eklenecek yeni bir token'ın eğitim verisi içinde geçme olasılığı belirli bir eşik değerinin altına indiğinde algoritma sonlanmaktadır.

Böylece dilden bağımsız olarak kelimelerin yukarıdaki örnekte görülen şekilde anlamlı parçalara ayrılması mümkün olabilmektedir. (Görseldeki "##" karakterleri o kelime bileşenin kelime başındaki bir bileşen olmadığını ifade etmektedir.)

WordPiece algoritmasının bilinirliği başarımda yüksek bir sıçrama yakalayan BERT modeli içinde dilden bağımsız bir tokenizer olarak kullanılması ile birlikte çok artmıştır.

2.6.3 Byte-pair Encoding Algoritması



Şekil 2.14: BPE tokenizasyon algoritması

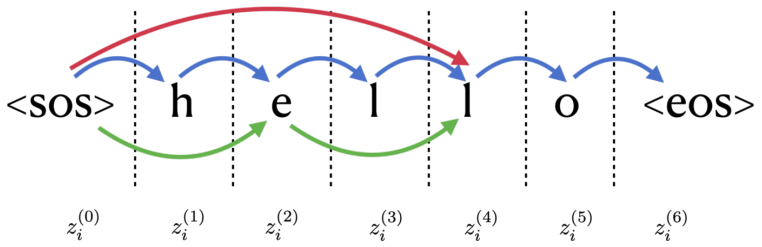
BPE'nin ilk çıkış maksadı esas itibariyle bir sıkıştırma algoritması olmasıdır. Çalışma mantığı gereksiz yer kaplayan fazlalık bilginin ortadan kaldırılmasına dayanmaktadır. (Sennrich et al., 2016) (Bostrom and Durrett, 2020) Bunu yapmak için girdide en sık geçen bayt ikilisini (girdide geçmeyen) tek bir bayt gibi kodlar ve bu süreci belirli bir eşiğe ulaşana kadar devam ettirir. Yan tarafta orjinal makaleden alıntılanan algoritmayı adım adım açıklamak gerekirse:

Derlemdeki tüm tekil karakterleri sözlüğe birer token olarak ekle. Derlemde bir arada en sık geçen iki token'ı birleştir ve sözlüğe yeni bir token olarak işle. Derlemdeki token'ları yeni token ile değiştir. Sözlük boyutu belirli bir eşiğe ulaşana kadar adım 2'den devam et.

2.6.4 Unigram (SentencePiece) Algoritması

BPE veya WordPiece'in aksine, Unigram temel kelime dağarcığını çok sayıda sembolle başlatır ve daha küçük bir kelime hazinesi elde etmek için her bir sembolü kademeli olarak atarak küçültür. Böylece temel sözcük dağarcığı, örneğin önceden belirtilmiş tüm sözcükleri ve ayrıca en yaygın olarak geçen kelime altı dizilere karşılık gelecek şekilde oluşur. Unigram modeli SentencePiece algoritması ile birlikte kullanılır. SentencePiece, kelime alt modellerini doğrudan ham cümlelerden eğitmekte olduğundan hem dilden bağımsız hem de açık kaynak kodlu bir çözüm sağlamaktadır (Kudo and Richardson, 2018).

İlk uygulamaları İngilizce-Japonca makine çevirisi üzerinde yapılmış olan SentencePiece, kelime alt modellerini doğrudan ham cümlelerden eğitmekte olduğundan hem dilden bağımsız hem de açık kaynak kodlu bir çözüm sağlamaktadır.



Şekil 2.15: Unigram (SentencePiece) tokenizasyon örneği

WordPiece önceden ayrıştırılmış olan sözcüklerin içindeki kelime altı bileşenleri (token) bulmaya odaklanmışken, SentencePiece modeli ham metin üzerinde çalışır. Bu sırada kelimelerin ayrıştırılması yerine boşluklar özel bir simge (_) ile temsil edilir ve diğer karakterler gibi normal olarak işlem görür. Bu nedenle öncelikle kelimeler arasında boşluk bulunmayan Çince ve Japonca gibi dillere yönelik olarak kullanılmıştır.

2.6.5 Tokenizasyonun Önemi ve Etkisi

Tokenizasyonun dil modeli üzerindeki etkisi, üzerinde çalışma yapılan metnin dili ve kullanılan dil modeline göre değişebilmektedir. (Goldsmith, 2001) (Taylor et al., 2022) Türkçe için de tokenizasyonun dil modeli performansı üzerinde önemli bir etkiye sahip olabileceğini düşünmeizin nedeni Türkçenin karmaşık morfolojik yapıları ve çok sayıda ekler kullanmasıdır. Bu nedenle Türkçe için doğru token sınırlarını belirlemek kolay bir problem değildir. Türkçe için verimli olmayan tokenizasyon şemaları kullanmak verimli olmayan kelime temsillerinin oluşmasına ve model doğruluğunun azalmasına neden olabilecektir.

Bu nedenle Türkçe'ye özgü bir dil modelinin performansını iyileştirmek için, Türkçenin morfolojik karmaşıklığını doğru bir şekilde yakalayan uygun tokenizasyon yöntemlerini kullanmak önemli olacaktır. Daha önce incelenen üç algoritmanın da bir arada geçme olasılıkları yüksek olan kelime altı bileşenleri

(harf, karakter vb.) gruplayarak ayrı tokenlar olarak belirlemeye çalıştığını ancak bunu yaparken farklı yöntemler izlediklerini söylemek mümkündür. Bu yöntemin Türkçe için ne kadar verimli bir yöntem olduğu ise tartışmaya açık bir konudur.

Derin dil modellerinin eğitiminde kullanılan toplam benzersiz token sayısı (vocabulary size) da önemli bir parametredir. Tüm bu tokenler bir araya gelerek aslında modelimizin öğrendiği yani işleyebileceği kelime dağarcığını belirler. Model tarafından işleme alınacak herhangi bir metin verisi, bu token sayısına ve seçilen tokenizasyon algoritmasına göre tokenlara bölünür. Zaman zaman işlenen metinlerde öğrenilmiş token sayısında bulunmayan bazı tokenlar gözlemlenebilir. Bunları bilinmeyen tokenlar ya da kelime dağarcığında yer almayan (out of vocabulary) kelimeler olarak ifade ederiz. Tokenizasyon algortiması da bu metinleri "bilinmeyen token" olarak işaretler. Bilinmeyen tokenların sayısının çok olması, anlamsal bağlamın doğru bir şekilde kapsanmasının önünde bir engel oluşturur ve genellikle modelde performans kaybına yol açar.

Kelime dağarcığındaki token sayısının yüksek olması bu bakımdan iyidir, çünkü token sayısı ne kadar fazla ise o kadar çok token işaretleme işlemi yapılabilir ve bilinmeyen tokenların oluşma olasılığı azalır. (Zheng et al., 2021) Fakat kelime dağarcığındaki token sayısının yüksek olmasının dezavantajları da vardır. Token sayısının fazla olması model açısından öğrenilmesi gereken daha fazla bilgi anlamına gelir ve modelin boyutunu büyüterek hesaplama açısından daha az verimli bir model olmasına neden olur, yani modelin bellek gereksinimi artar ve modelin eğitimi daha zor ve maliyetli hale gelir. Dolayısıyla kelime dağarcığının büyütülmesi modelin toplam performansında giderek azalan bir olumlu etki oluşturur hatta belirli bir noktadan sonra pratikte olumsuz bir etki oluşturmaya başlar. Sonuç olarak kelime dağarcığı boyutu açısından model boyutu ve performansı arasında bir denge sağlanması gerekmektedir (Gowda and May, 2020).

Dolayısıyla bir modelin işleyebildiği token sayısı (kelime dağarcığı büyüklüğü) model boyutu ve başarımı açısından optimum dengesi bulunması gereken

oldukça önemli bir parametredir.

Literatürdeki çalışmalar derin öğrenme odaklı dil modellerinin eğitimi aşamasında metin girdilerinin kodlanmasında ve tokenizasyonunda seçilen yöntemlerin elde edilen modelin doğal dil işleme görevlerindeki başarımında önemli farklar meydana getirebildiğini göstermektedir. (Bostrom and Durrett, 2020)

Eğer bir sinir ağı modelinde kullanılan tokenizer bileşeninde yapısal değişiklikler uygulanırsa, bu değişikliklerin model üzerinde aşağıdaki gibi etkileri olabilir:

1. Girdi boyutu ve biçimi değişebilir: Yeni tokenizer, metni farklı sayıda tokena bölebilir veya bazı metinler için farklı token türleri oluşturabilir. Bu durumda, modelin girdi yapısı ve boyutu da değişecektir.Tokenizer, metni daha küçük parçalara böldüğünde, girdi boyutu azalır ve modelin daha hızlı çalışmasını sağlayabilir. Aynı zamanda, metni daha büyük parçalara böldüğünde, girdi boyutu artar ve modelin daha karmaşık bağlantıları ve uzun vadeli bağlamsal ilişkileri daha iyi anlamasına yardımcı olabilir. Yeni bir tokenizer, farklı dil yapılarını temsil etmek için farklı token türlerini kullanabilir ve bu da modelin metni farklı şekillerde yorumlamasına ve işlemesine yol açar. Tokenizerde yapılan bu tür değişiklikler, doğal olarak modelin de performansını etkileyecebilecektir. Örneğin girdi boyutundaki değişiklikler, modelin bellek kullanımını ve hesaplama maliyetini etkileyebilir ve modelin eğitim sürecini etkileyebilir. Girdi biçimindeki değişiklikler, modelin metni nasıl yorumladığını ve anladığını değiştirebilir ve bu da modelin başarı ve etkinliğini etkiler.

2. Anlam kaybı veya kazancı: Farklı tokenizerlar, metni farklı şekillerde tokenlara böler. Bu da modelin metnin anlamını daha iyi veya daha kötü temsil edebilmesine yönelik bir etki oluşturabilir. Bazı tokenizerlar metni daha doğru bir şekilde bölerek, dilin yapısını ve anlamını daha iyi yakalayabilir. Bu durumda, model daha zengin bir bağlam anlayışına sahip olabilir ve metindeki ilişkileri daha iyi kavrayabilir. Eğer yeni tokenizer, metindeki nüansları ve önemli bağlamsal bilgileri koruyabiliyorsa, modelin performansı artabilir.

3. Performans değişiklikleri: Tokenizer değişiklikleri, modelin performansını etkileyebilir. Eğer yeni tokenizer, metni daha doğru bir şekilde işlemeyi sağlıyorsa, modelin performansı artabilir. Ama tersi durumda, performans düşebilir. Bazı tokenizerlar, dilin yapısını ve özelliklerini daha iyi yakalayabilir ve modelin dil öğrenme yeteneğini artırabilir.

Bu nedenle, tokenizerde yapılan herhangi bir değişikliğin, modelin performansı ve yetenekleri üzerinde önemli etkileri olabilir. Eğer tokenizerde değişiklik yapılmışsa, modelin yeniden eğitilmesi veya bazı durumlarda en azından ince ayar eğitimleri yapılması gerekecektir. Bu sayede, yeni tokenizerın getirdiği değişikliklerin uygun bir şekilde yönetilebilmesi ve modelin hedef görevlerde en iyi performansı göstermesi sağlanabilir.

Sonuç olarak, daha iyi bir tokenizasyon, dilin yapısını ve ilişkilerini daha iyi yansıtan bir girdi sağlayarak, besleyeceği modelin performansını artırabilir ve o dile özgü görevlerde daha iyi sonuçlar elde edilmesini sağlayabilir (Kudo, 2018). Bu çalışmada da BERT modelinin Türkçe diline özgü özelliklerini daha iyi yakalayabilmesi için tokenizasyon aşamasında (WordPiece) yapılabilecek bazı iyileştirmeler ortaya koyulmuştur.

3 YÖNTEM

Tez çalışmaları boyunca genel bir yöntem olarak literatür taraması sırasında karşılaşılan modellerden seçilenlerin Türkçe derlemler üzerinde denenmesi, hangi modellerin Türkçe için daha başarılı ve anlamlı sonuçlar verdiğinin incelenerek yorumlanması, farklı yöntemlerin bir araya getirilerek Türkçe için daha iyi performans verebilmesi için alternatif yaklaşımlar geliştirilmesi yöntemleri uygulanmıştır.

Bu analiz çalışmaları sırasında yapılan inceleme ve denemelerin sonuçları da müteakip başlıklarda yer almaktadır. Daha sonraki çalışmalarda, bu analiz çalışmalarından elde edilen sonuçlara dayanarak, Türkçe gibi sondan eklemeli ve morfolojik açıdan zengin dillerin doğasında yer alan ve önceki bölümde açıklanan öznitelikleri nedeniyle literatürde yaygın olarak kullanılan ve kelimeleri bütün olarak ele alan yöntemlerin başarımının sınırlı düzeyde kalacağı, bu yöntemlerin kök, ek, hece, karakter gibi daha küçük dil birimlerini ve ayrıca ses dönüşüm kuralları gibi dile özgü gibi özellikleri dikkate alan temsil yöntemleri ile güçlendirilmesinin ise bu tarz diler için daha başarılı sonuçlar vereceği fikri bir önkabul olarak ele alınmıştır.

Tezin müteakip bölümlerde detaylı açıklamaları yer alan uygulamalı araştırmalardan elde edilecek sonuçlar ile bu hipotezin doğruluğu test edilerek incelenmeye çalışılmıştır. Bu kapsamda da temel yöntem olarak Türkçe diline özgü temel bazı kuralların tokenizer içinde uygulanması ve VİT başarımında ortaya çıkacak farkların incelenmesi yöntemi seçilmiştir.

Sonuç olarak VİT başarımında iyileşme sağlamak için BERT modelinin girdilerinin dile özgü ilave özellikler ile geliştirilmesi kapsamında iyileştirme sağlanabilecek temel bir aşama olarak tokenizasyon aşaması olduğu görülmüş ve nihai olarak WordPieceTR tokenizasyon algoritması ile birlikte bu algoritmayı kullanan BERT-TR modeli tez çalışmasının nihai çıktısı olarak ortaya konulmuştur.

3.1 WordPieceTR Tokenizasyon Algoritması

WordPieceTR, Türkçe metinlerin daha verimli ve etkili bir şekilde işlenmesi için özel olarak geliştirilmiş bir dil modeli tokenizasyon algoritmasıdır. Bu algoritma da WordPiece algoritması gibi metni parçalara, yani tokenlara böler ve bu tokenleri dil modeli tarafından daha iyi işlenebilir hale getirir.

Çalışmamızın odak noktası olarak Transformer modelleri arasında başarımda ciddi sıçramalara vesile olmuş olan BERT modeli seçilmiştir. BERT modeli içinde kullanılan WordPiece algoritması bir kelime altı tokenizasyon algoritmasıdır. Bu özelliği itibariyle WordPiece aslında genel olarak derlem içinde geçme olasılığı çok olan karakter öbeklerinin etkin bir şekilde temsil edilmesine imkan sağlamaktadır. Dolayısıyla bu özelliği ile WordPiece'in morfolojik açıdan zengin dillerde önemli yere sahip olan ek yapılarını da belirli bir verim ile işleyebilmektedir. Literatürdeki çalışmalar incelendiğinde de WordPiece algortimasındaki modellemelin Word2Vec gibi modellerle kıyaslandığında morfolojik açıdan zengin dillerde daha başarılı sonuçlar ortaya çıkardığı görülmektedir (Joulin et al., 2016) (Bojanowski et al., 2016). Bu durum bizim arkplan çalışmaları içinde yaptığımız deneysel uygulamalarımızda da gözlemlenmiştir.

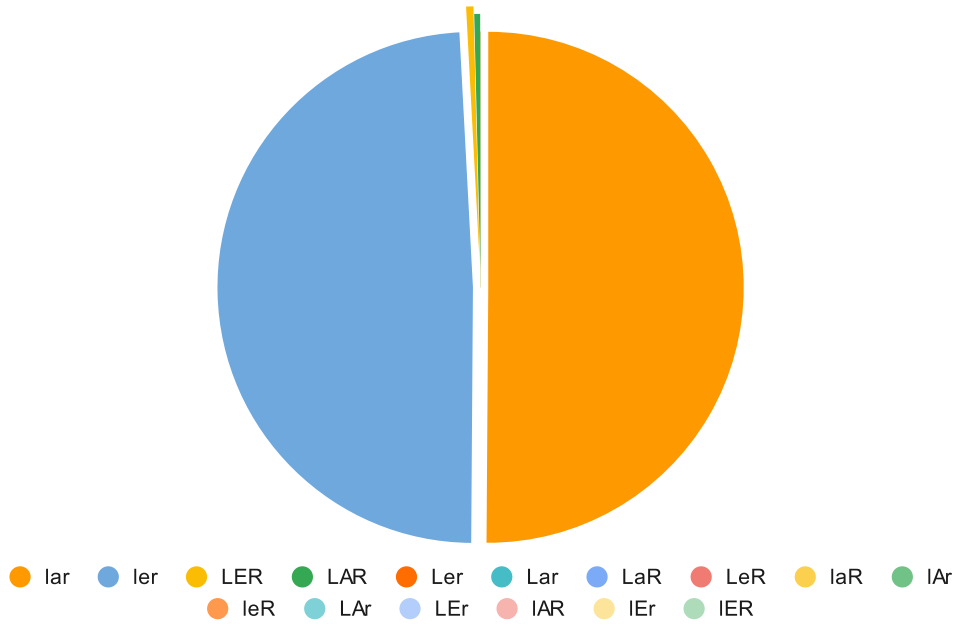
Ancak, WordPiece algoritması tamamen dilden bağımsız bir algoritmadır ve kaynak dilin özelliğinden kaynaklı olan ses dönüşümlerini ya da benzer herhangi bir dil bazlı özniteliği dikkate alan bir yetenek bu algoritma içinde mevcut değildir. Böyle bir özelliği olmadığından, türkçedeki ses dönüşümleri nedeniyle özellikle ek yapılarını oluşturan karakter öbekleri gibi aslında aynı anlama gelen kelime altı parçaların metin içinde ince ünlüler ve kalın ünlüler içerecek şekilde farklı biçimlerde geçtiği durumlarda (örn: ler, lar) WordPiece bunları farklı şekilde tokenize edilebilecek karakter öbekleri olarak ele almaktadır. Bu çalışmada bu durumu Türkçe açısından WordPiece algoritmasında iyileştirme yapılabilecek bir nokta olarak ele alınmış ve bu kapsamda WordPiece algoritması üzerinde Türkçe'nin özelliklerini daha iyi yansıtacak şekilde iyileştirmeler yapılması hedeflenmiştir.

Tokenizer bileşeninin değiştirilmesi ile dil modeli üzerinde meydana gelebilecek değişikliklere daha önce "Tokenizasyonun Önemi ve Etkisi" başlığında değinilmişti. Bu çalışmada genel bir yaklaşım olarak WordPiece algoritmasının Türkçe'nin özelliklerini daha iyi yansıtacak şekilde geliştirilmesi ile, WordPiece algoritmasının tokenizasyon aşaması içinde kullanıldığı BERT modelinde de Türkçe özelinde başarımının artacağı kabulu ile WordPieceTR algoritması geliştirilmiştir. WordPieceTR algoritması, WordPiece algoritmasının Türkçe'ye özel bazı durumları ele alacak şekilde uyarlanmış bir versiyonu olarak görülebilir. WordPiece algoritması, metni tekrarlayan karakterler ve alt kümeler temel alınarak parçalara böler ve böylece dil modelinin daha öğrenilebilir bir dil yapısı elde etmesine yardımcı olur. WordPieceTR algoritması, Türkçe dil özelliklerini dikkate alarak daha iyi bir tokenizasyon sağlamak için çeşitli ilave adımlar içerecek şekilde geliştirilmiştir.

Bir tokenizer'ın çalışma mantığı gereği, encode edilen verinin decode edilebilmesi ve decode edildiğinde orjinal veriye deterministik bir şekilde dönüş yapılabilmesi gerekmektedir. Türkçe'de yukarıda örneklenen pek çok özellik mevcut olmasına rağmen tüm bu özelliklerin geliştirilen algoritmaya tek bir adımda yansıtılması mümkün olmamış, geliştirmeler 2 aşamalı ve kontrollü olarak yapılmıştır.

Bu kapsamda, başarımda ortaya çıkacak farkın gözlemlenmesi maksadıyla ilk aşama olarak OSCAR derlemi genelinde toplamda 16.620.927 kere olmak üzere, şekil 3.1'de belirtilen farklı biçim dağılımı çerçevesinde geçen, genellikle birden fazla varlığı ifade etmek için isimlerden sonra gelen veya bazı durumlarda saygı ifadesi olarak kullanılan ve biçimsel olarak "-ler, -lar" şeklinde iki farklı ses biçiminde ortaya çıkan çoğul durum ekleri üzerinde çalışma yapılmış, bir sonraki aşamada ise farklı ek ve özelliklerin de implemente edilerek algoritmanın daha da geliştirilmesi planlanmıştır.

Orjinal WordPiece algoritmasının implementasyonu baz alınarak birinci safha geliştirmelerine başlanan WordPieceTR algoritması tokenize edilen kelime içinde "ler/lar" ifadeleri geçiyorsa bu ifadeleri "##l{a|e}r" şeklinde kodlayarak tek bir token olarak türetmektedir. Decode aşamasında ise "##l{a|e}r"



Şekil 3.1: "-ler, -lar" eklerinin derlemde farklı biçimlerde geçme sıklığı

olarak encode edilmiş verinin orjinal haline geri dönüştürülmesi için bir önceki token içinde yer alan en son sesli harfin ince mi kalın mı olduğuna göre "lar" veya "ler" çıktısı üretmektedir. Böylece decode işlemi deterministik bir şekilde yapılmış olmaktadır.

BERT modelinin girdilerinin Türkçe'ye özgü özniteliklerin yorumlanması ile geliştirilmesi kapsamda önceki dönemde BERT modelinin tokenizasyon aşamasında kullanılmakta olan WordPiece algoritması Türkçe'de çok yaygın şekilde kullanılan ve biçimsel olarak "-ler, -lar" şeklinde iki farklı biçimde ortaya çıkan çoğul durum eklerini tek bir token ile temsil edecek şekilde düzenlenmiştir.

Çalışmanın ikinci aşamasında "-ler, -lar" eklerine yönelik yapılan çalışma derlem içerisinde geçme sıklıklarına göre seçilen ve Tablo 3.1'de yer alan 28 farklı ek biçimini kapsayacak şekilde genişletilmiştir. Böylece model içerisinde yer alan Tokenizer bileşeni Türk dilinin özelliklerine göre daha iyi genelleştirme yapılabilecek özelliklerle geliştirilmiştir.

WordPieceTR algoritması tokenize edilen kelime içinde üzerinde çalışma yapılan eklerin paternine uygun olan ek ifadeleri (örnek: -sak/-sek) geçiyorsa

Harf Örüntüsü Örneği	**Farklı Kullanım Örnekleri**	**Token Örüntüsü**
-maz	paslanmaz, bükülmez	##m{a,e}z
-cak	evcek, çabucak	##c{a,e}k
-lar	kitaplar, kalemler	##l{a,e}r
-sal	duygusal, sezgisel, bölgesel	##s{a,e}l
-cı	avcı, oyuncu, simitçi, sütçü	##{c,ç}{u,ı,ü,i}
-msi	ekşimsi, acımsı	##ms{u,ı,ü,i}

Tablo 3.1: WordPieceTR algoritması kapsamında özel olarak tokenize edilen harf örüntülerine örnekler

bu ifadeler uzman sistem yaklaşımı ile Tablo 3.1'de karşılık olarak örnekleri yer aldığı şekilde Türkçe'ye özel bir token örüntüsü ile temsil edilip genelleştirilir ve bu şekilde kodlanıp her bir olası ek varyantı tek bir token olarak türetilir.

Tablo 3.1'de sadece bazı kurallara ait örnekler yer almakta olup Türkçe'nin öznitelikleri bağlamında WordPieceTR Tokenizasyon Algoritması içinde implemente edilmiş her bir farklı ek tipi için geçerli olan ve tüm kuralları kapsayan tablo ayrı bir ek olarak Tablo A-1'de sunulmuştur.

Bazı örnek Türkçe kelimeler için WordPieceTR ve WordPiece algoritmaları ile ile tokenize edilmiş karşılıkları Tablo 3.2'de sunulmuştur. Bu tablodan da görüleceği üzere WordPieceTR tarafından karşılanan özel kurallara uyan kelimleler için token sayısı genellikle daha düşük olmaktadır.

Böylece tokenizasyon aşamasında kullanılan ve kelime altı parçalardan oluşan güncel token sözlüğünün içeriği Türkçe'nin ses dönüşüm özelliklerini daha iyi yansıtacak şekilde güncellenir. WordPieceTR içinde uygulaması yapılmış kurallara uymayan durumlarda ise WordPieceTR algoritması yöntem olarak WordPiece algoritması ile aynı şekilde çalışır.

Decode aşamasında ise kodlanmış olarak (Örnek: "##la|er") olarak encode edilmiş verinin orjinal haline geri dönüştürülmesi için bir önceki token içinde yer alan harf ve seslerin Türkçe ses uyumu kurallarına uygunluğuna göre olması gereken dönüşüm uzman sistem yaklaşımı ile deterministik bir

Kelime	**WordPiece**		**WordPieceTR**	
	Token	**Adet**	**Token**	**Adet**
anlatarak	an, ##lat, ##ara, ##k	4	an, ##lat, ##[y]{a|e}r{a|e}k	3
gülerek	gü, ##ler, ##ek	3	gül, ##[y]{a|e}r{a|e}k	2
dinlemeden	din, ##lem, ##ede, ##n	4	din, ##le, ##m{a|e}d{a|e}n	3
gecikmeksizin	ge, ##ci, ##km, ##ek, ##si, ##zin	6	ge, ##cik, ##m{a|e}ks{ı|i}z{ı|i}n	3
yorulmaksızın	yo, ##ru, ##lma, ##ks, ##ı, ##z, ##ı, ##n	8	yor , ##ul, ##m{a|e}ks{ı|i}z{ı|i}n	3
beklemeksizin	be, ##kle, ##me, ##ks, ##iz, ##in	6	be, ##kle, ##m{a|e}ks{ı|i}z{ı|i}n	3
paslanmaz	pas, ##lan, ##ma, ##z	4	pas, ##lan, ##m{a|e}z	3
unutulmaz	un, ##ut, ##ul, ##ma, ##z	5	un, ##ut, ##ul, ##m{a|e}z	4

Tablo 3.2: WordPieceTR ve WordPiece algoritmaları ile tokenize edilmiş Türkçe ifadelerin kıyaslaması

şekilde ve otomatik olarak yapılarak orjinal metinde yer alan (Örnek: “-lar” veya “-ler”) metin girdisi üretilebilmektedir. Böylece WordPieceTR algoritması, Türkçe metinleri daha iyi bir şekilde tokenize ederken aynı zamanda metinlerin dil özelliklerini de korumuş olur.

Sonuç olarak WordPieceTR algoritması ile, özellikle Türkçe diline odaklı dil modellerinde hem daha yüksek başarımlı hem de tutarlı sonuçlar elde edilmesine yardımcı olunması hedeflenmiştir.

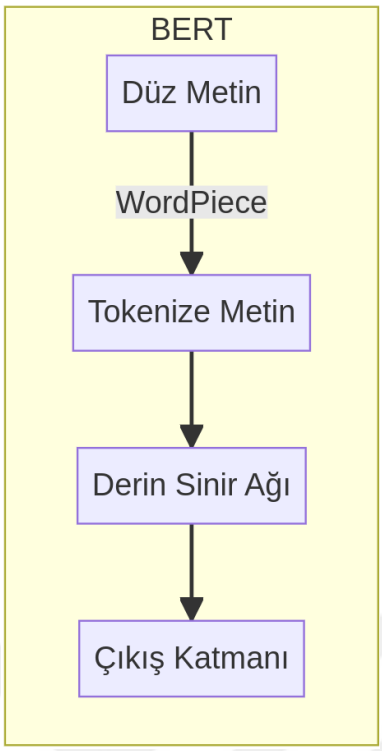
3.2 BERT-TR Modeli

Türkçe VİT çalışmalarında daha yüksek bir başarım elde edebilmek için Türkçe metinleri daha iyi bölümleyebilen ve kelime altı seviyesinde ses dönüşüm kurallarını dikkate alabilecek şekilde özel olarak tasarlanmış WordPieceTR tokenizasyon algoritmasının kullanılması gerekli olmuştur. Tokenizer’ın değiştirilmesi metin verisinin işleme alındığı en temel katmanda değişiklik gerektirdiğinden mevcut bir BERT modelinin ince ayar yöntemi ile eğitilmesi mümkün olmayacaktır. Bu nedenle oluşturulacak yeni modelin yeni tokenizer ile sıfırdan eğitilmesi zorunlu olmaktadır. Böylece yeni geliştirilmiş WordPieceTR tokenizer kullanılarak BERT modelini baz alan yeni bir model olan BERT-TR modeli oluşturulmuştur.

Şekil 3.2 ve şekil 3.3’den de açıkça görüleceği üzere BERT-TR modelinin BERT modelinden temel farkı giriş metni üzerinde Türkçe’nin karakteristiğine uygun bir tokenizer olan WordPieceTR tokenizer kullanmasıdır.

Seçilen yeni tokenizer ile birlikte, sıfırdan eğitilmiş bir BERT modeli oluşturulmalıdır. Böylece Türkçe dilinin özelliklerin daha uygun bir kelime dağarcığına ve dil yapısına sahip olacak şekilde önceden eğitilmiş BERT tabanlı yeni bir model elde edilmiş olacaktır. Elde edilen bu yeni modelin, Türkçe VİT veri kümesi üzerinde ayrıca eğitilmesi (ince ayar) gerekli olacaktır.

İnce ayar eğitim sürecinden sonra, elde edilen nihai model Türkçe VİT görevine özgü bir hale gelmiş olacaktır. Böylece modelimizin çıktı katmanı ve etiketleme mekanizması VİT görevine uygun olarak düzenlenmiş olacaktır.



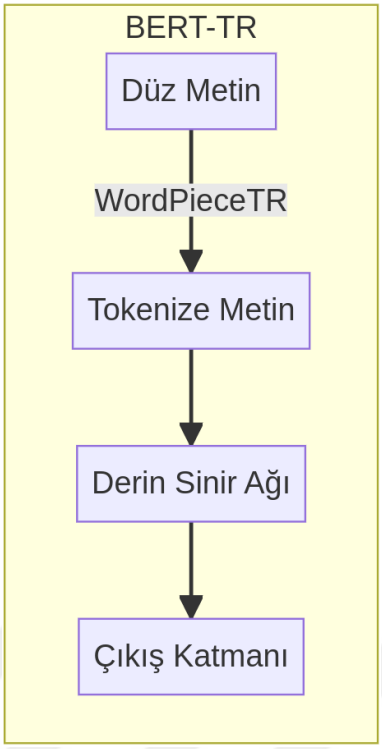
Şekil 3.2: BERT modelinde kullanılan tokenizer

Böylece modelimiz artık Türkçe varlık isimlerini daha doğru bir şekilde tanımlayabilir hale gelecektir.

Özellikle ilk adım olarak ele aldığımız sıfırdan eğitme aşamasında yüksek işlem gücüne ve gerçekten büyük veri kümelerine ihtiyaç duyulacaktır. Bu veri kümesinin etiketlenmiş olması gerekmemektedir.

İkinci adım olarak ele aldığımız ince ayar eğitimi aşamasında ise Türkçe VİT görevi için uygun bir veri kümesi kullanılması gerekmektedir. Bu veri kümesi ise etiketlenmiş metin örneklerinden oluşmalıdır. Metin örneklerindeki varlık isimleri (örneğin, kişi adları, organizasyon adları, yer adları) etiketlenmiş olarak bulunmalıdır. Genellikle böyle veri kümesi el ile etiketleme yöntemi ile oluşturulabilir. Ancak bu aşamada kullanılacak veri setinin birinci aşamadaki kadar büyük bir veri seti olması zorunlu değildir.

Eğitimler tamamlandıktan sonra, modelin performansını değerlendirmek için ince ayar eğitim aşamasında kullanılmamış verilerden oluşan ayrı bir test veri kümesi kullanılmalıdır. Bu test veri kümesi de el ile etiketlenmiş varlık



Şekil 3.3: BERT-TR modelinde kullanılan tokenizer

isimlerini içeren Türkçe metin örneklerinden oluşmalıdır. Elimizdeki mevcut tek VİT veri setinin bir kısmı eğitim bir kısmı test için ayrılarak kullanılabilir.

Türkçeye özgü nitelikleri dikkate alacak şekilde eklemeler yapılarak oluşturulan yeni modelin önceki modelle doğru bir şekilde kıyaslanabilmesi için burada belirtilen tüm eğitim aşamaları eklemeler yapılmamış model ile de uygulanarak iki modelin başarım seviyeleri kıyaslanmıştır.

Derlem Türü	**Kelime sayısı**	**Boyut**
Orjinal	7,577,388,700	60 GB
Tekilleştirilmiş	3,365,734,289	27 GB

Tablo 4.1: Öneğitim aşamasında kullanılan "OSCAR 2019 Turkish" derlem bilgileri

4 DENEYLER

Tez çalışmalarının ilk döneminde öncelikle mevcut modellerin yapısında hiç değişiklik yapılmadan veya birden fazla modelin bir araya getirilerek Türkçe veri setlerinde nasıl davrandıkları incelenmiştir. Bu çalışmalar daha çok analiz maksatlı çalışmalar olup daha sonra yapılacak özgün çalışmalara temel oluşturmak ve sonuçların kıyaslanması maksadıyla gerçekleştirilmiştir. Tez içerisinde bu çalışmalar analiz/erken dönem çalışmaları olarak da ifade edilmektedir.

4.1 Kullanılan Veri Setleri

Tez çalışması kapsamında BERT modelinin sıfırdan eğitilmesi (öneğitim) aşamasında kullanmak için etikelenmiş olması gerekmeyen büyük boyutlu Türkçe derlem ihtiyacını karşılamak üzere OSCAR 2019 Turkish derleminin tekilleştirilmiş (tekrarlayan veriler temizlenmiş) halinden faydalanılmıştır. (Ortiz Suárez et al., 2020) Orjinal OSCAR 2019 Turkish derlemdeki toplam kelime sayısı 7,577,388,700 (boyut: 60GB) iken derlemin tekilleştirilmiş halindeki toplam kelime sayısı 3,365,734,289 (boyut: 27GB) dir. (Bakınız Tablo 4.1)

İnce ayar eğitimi (finetuning) aşamasında ihtiyaç duyulan etiketlenmiş Türkçe VİT veri seti olarak ise BOUN Web Corpus (ver. 1.0) kaynak olarak faydalanılmıştır. Bu derlem içerisinde 492.340 kelime, 27.564 cümle mevcuttur. Verinin sınıflandırılmasında arkaplan bölümünde detayları verilen ve BIO kodlamasına göre daha detay bilgi içeren BILOU kodlamasına uygun şekilde belirlenmiş 13 farklı etiket (B-LOC, B-ORG, B-PER, I-LOC, I-ORG, I-PER, L-LOC, L-ORG, L-PER, U-LOC, U-ORG, U-PER, O) ile VİT etiketlemesi

Etiket	Sıklık
O	438.701
B-PER	16.292
B-LOC	11.716
B-ORG	9.183
I-PER	7.746
I-ORG	6.880
I-LOC	1.823

Tablo 4.2: BIO şemasına göre VİT etiketi sıklıkları

yapılmış durumdadır.

Bu çalışamada BIO kodlama standardı tercih edilmiş olduğundan kaynak veri seti etiketlerinde dönüşüm yapmak gerekli olmuştur. Bunun için arkaplan bölümünde detayları verilen basit dönüşüm işlemi uygulanmıştır. BILOU kodlamasından BIO kodlamasına dönüşüm yapılmasına ilişkin bir örnek Tablo A.2'de, dönüşüm işlemi öncesinde orjinal veri setindeki etiket sıklıklarına ilişkin bilgileri Tablo A.3'de dönüşüm işlemi sonrasında elde edilen ve ince ayar eğitimi aşamasında kullanılan veri setindeki etiket sıklıklarına ilişkin bilgiler ise Tablo 4.2'de yer almaktadır.

Elde edilen yeni veri setli üzerinde manuel analizler gerçekleştirilmiş ve herhangi bir anomali olup olmadığı kontrol edilmiştir. Kullanılan veri setinde karakter kodlamaları ile ilgili tespit edilen sorunlar manuel yöntemlerle düzeltilmiş ve bu konularda veri kaynağına geri dönüş sağlanmıştır. Veri seti etiketlerindeki dönüşümün NER başarımına etkisine yönelik sonuçlara müteakip bölümde değinilecektir.

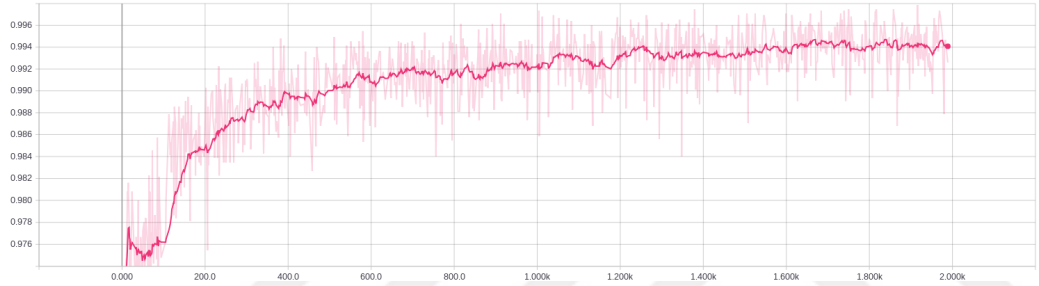
4.2 Veri Seti Etiketlerindeki Dönüşümü

Elde edilmiş olan yeni veri setleri üzerinde, hem kelime vektörlerini hem de karakter temsillerini girdi olarak bir arada bir alan VİT maksatlı çift yönlü uzun kısa süreli bellek ağ (Bi-LSTM) modelleri gibi farklı ağ modelleri ve farklı

Çalışma:	**F1-score:**
Şeker and Eryiğit (2012)	91.94%
Demir and Özgür (2014)	91.85%
Kuru et al. (2016)	91.30%
Güngör et al. (2017)	92.93%
DBMDZ (2020) (IOB)	**95.23%**
DBMDZ (2021) (IOB)	**93.49%**

Tablo 4.3: Diğer sonuçların kıyaslanması

ağ parametreleri ile toplam 92 adet VİT uygulaması gerçekleştirilmiştir.



Şekil 4.1: En başarılı modelin eğitim başarım (accuracy) grafiği

Tablo 4.2'den de görüleceği üzere elde edilen en yüksek başarım sadece 5 epoch ile elde edilmiş olup bir erken dönem çalışmalarına nazaran %2,6 puan yüksek başarım elde edilmiştir.

Etiket sayısının indirgenmesi ile başarımda ne kadar değişim olacağının gözlenmesi maksadıyla BERTurk (Schweter, 2020) çalışmasında kullanılan model ile özdeş bir model oluşturularak BIO ve BILOU etiket şema yapısındaki farklı veri setleri ile deneyler yapılmıştır. Bu deneylerde elde edilmiş olan başarım durumu Tablo 4.2'de sunulmuştur.

Hem başarım sonuçlarında küçük bir iyileşme oluşmaması hem de daha önce yapılan çalışmalarla daha doğru bir şekilde kıyaslama yapılabilmesi maksadıyla BIO şemasının kullanılması tercih edilmiştir. Müteakip başlıklarda değinilen BERT modelinin girdilerinin dile özgü ilave özellikler ile geliştirilmesinin başarıma olan etkisinin incelenmesi kapsamındaki çalışmalarda BIO

Model	Süre (dk.)	Epoch	F1	Precision	Recall
1	93	2	%93,24	%92,72	%93,77
2	127	3	%93,64	%93,01	%94,28
3	181	4	%93,77	%93,18	%94,36
4	217	5	**%93,9**	%93,41	%94,39
5	414	10	%93,84	%93,48	%94,21

Tablo 4.4: BERT transfer learning başarımı (13 etiket)

Model	Süre (dk.)	Epoch	F1	Precision	Recall
1	97	2	%93,54	%93,02	%94,07
2	122	3	%93,87	%93,24	%94,51
3	183	4	%94,46	%93,85	%95,07
4	215	5	%94,65	%94,12	%95,19
5	273	6	**%94,92**	%94,43	%95,41

Tablo 4.5: BERT transfer learning başarımı (7 etiket)

şemasına uygun olarak 7 etikete indirgenmiş veri seti kullanılmıştır.

4.3 BERT-TR Modeli

Mevcut modellere yönelik analiz çalışmalarında genel olarak mevcut modeller olduğu gibi kullanılmış bazı durumlarda hibrit modellerle deneyler yapılmıştı, BERT modeli ile gerçekleştirilen analiz uygulamalarında ise ön-eğitim aşaması uygulanmadan doğrudan google tarafından yayınlanmış olan "BERT-Base, Multilingual Cased" modelinden faydalanılarak, bu model üzerine "Transfer Learning" tekniğine uygun olarak mevcut tokenizerler ile ince ayar eğitimleri yapılarak VİT modelleri eğitilmişti. Bu analiz çalışmalarından elde edilen bir netice olarak, dile özgü iyileştirme yapılabilecek bir alan olarak tokenizer bileşeni seçilmiştir.

Eğer bir sinir ağı modelinin tokenizer bileşeni üzerinde yapısal değişiklik yapılırsa, bu değişiklik modelin tüm girdilerini etkiler. Çünkü tokenizer, metni işlemeye uygun parçalara (tokenlara) bölen bileşendir ve model bu tokenları

işlemek üzere eğitilir. Dolayısıyla, tokenizerde yapılan değişiklikler, metni nasıl parçaladığını ve hangi tokenlara dönüştürdüğünü değiştirir ve bu da modelin aldığı girdilerin tamamını değiştirir. Bu nedenle önceden eğitilmiş dil modelleri kullanırken ve bu modelleri NLP görevleri için ince ayar eğitimleri ile uyarlamaya çalışırken aynı Tokenizer'ın kullanılması önemlidir. Bu nedenle tokenizer bileşeni üzerinde yapısal değişiklik yapıldığında derin sinir ağı modelleri için oldukça maliyetli bir süreç olan modelin eğitim sürecinin ön-eğitim dahil en baştan yeniden işletilmesi gerekmektedir.

Bu kapsamda Bölüm 3.1'de detaylı tarifi yapılan ve 2 aşamalı olarak geliştirilen WordPieceTR Tokenizer'ı ile deneyler gerçekleştirilmiştir. Tokenizer bileşeninin değiştirilmiş olması daha önce farklı bir tokenizer ile eğitilmiş bir modelin temel model olarak kullanılabilmesi imkanını ortadan kaldırdığından, VİT görevinde kullanılacak yeni modelin sıfırdan eğitilmesi ve dolayısıyla yüksek işlem gücü ve büyük bir derlem gerektiren ön-eğitim işlemlerinin de deneyler kapsamında uygulanmasını zorunlu kılmıştır.

Ön-eğitim aşamasında genel olarak toplam boyutu 27 GB olan tekilleştirilmiş "OSCAR 2019 Turkish" derleminden faydalanılmıştır, ancak Ön-eğitim çok fazla işlemci gücü gerektiren bir çalışma olduğundan makul sürelerde anlamlı bir sonuca ulaşabilmek için, sıfırdan eğitim gerektiren deneylere başlamadan önce bir ön analiz çalışması yapılmıştır. Ön analiz safhasında kaynak bu veri seti içinden rastlantısal olarak 1.000.000 cümle seçilerek boyutu 404MB'lık daraltılmış bir derlem oluşturulmuştur.

Daraltılmış derlem üzerinde yapılan en analiz deneylerinde dile özgü nitelikleri tokenizer seviyesinde işlemek amacıyla sadece "-ler, -lar" eklerini "##la|er" şeklinde kodlayarak tek bir token olarak genelleştirecek bir uygulama yapılmıştır. Bu uygulama kapsamında daraltılmış derlem içinde "ler" ifadesinin 42.403 cümlede "lar" ifadesinin ise 43.864 cümlede geçtiği görülmüştür.

Ön analiz kapsamında daraltılmış derlem üzerinde orjinal BERT modeli ve yukarıda tarif edilen kurallarla çalışan yeni tokenizer ile düzenlenmiş BERT-

Model	**F1**	**Kesinlik**	**Duyarlılık**
BERT	%83,54	%83,02	%84,07
BERT-TR*	%84,32	%84,43	%84,21

Tablo 4.6: BERT ve BERT-TR (ön analiz çalışması) modellerinin VİT başarımı

TR modeli kullanılarak ön-eğitimler yapılmış ve 2 ayrı model elde edilmiştir. Elde edilen yeni modeller kullanılarak BIO şemasına (7 etiket) göre yapılan VİT ince ayar eğitimleri tekrarlanmış ve Tablo 4.3 sonuçlara ulaşılmıştır.

Elde sonuçlar incelendiğinde; ön analiz maksadıyla Türkçe için kısmi bir iyileştirme sağlayacak şekilde tokenizer ile eğitilmiş olan modelin başarımının (%84,32) orjinal WordPiece tokenizer'ı kullanılan BERT modelinin başarımından (%83,54) belirgin bir şekilde (%83,54) daha iyi olduğu tespit edilmiştir. Bu durum ön analiz çalışmasının maksadına ulaştığı ve tokenizer üzerinde dile özgü olarak yapılacak iyileştirmelerin modelin başarımına da olumlu yönde etki edebileceği varsayımımızı doğrulamıştır.

Elde edilen Türkçe için iyileştirilmiş BERT-TR modelinin, WordPiece algoritması kullanan orijinal BERT modeline göre VİT başarımına yönelik +%0,78 olumlu etkisi olduğu görülmüştür. Bu sonuçlara dayanarak, hem ön-eğitim için kullanılan derlem boyutunun büyütülmesi hem de modelde Türkçe'ye özgü yapılabilecek ilave iyileştirmelerin uygulanması ile daha da yüksek bir başarı elde edilebileceği değerlendirilerek bu yöndeki çalışmalara aşağıdaki şekilde devam edilmiştir. Bununla birlikte her iki modelin genel başarımının diğer çalışmalarla kıyaslandığında düşük kalmasının sebebinin ön-analiz kapsamında bu çalışmaların oldukça sınırlı bir derlem üzerinde yapılmış olmasından kaynaklandığı değerlendirilmiştir.

Bu kapsamda, ön-analiz aşamasında sadece "ler/lar" ifadelerinin genelleştirilme yeteneği eklenerek ana çatısı oluşturulan WordPieceTR algoritması ve implementasyonu Tablo A-1'de tam listesi ve örnekleri yer alan 28 farklı örüntüye ilişkin kural setini kapsayacak şekilde geliştirilerek uzman sistem yaklaşımı ile implemente edilmiştir. Böylece model içerisinde yer alan Tokenizer

Model	**F1**	**Kesinlik**	**Duyarlılık**
BERT	92,786%	92,751%	92,821%
BERT-TR	93,025%	92,929%	93,121%

Tablo 4.7: BERT ve BERT-TR modellerinin VİT başarımı

bileşeni Türk dilinin özelliklerine göre daha iyi genelleştirme yapılabilecek bir seviyeye gelmiştir.

Tokenizer kurallarının değiştirilmiş olması nedeniyle yüksek işlemci kaynağı gerektiren ön-eğitim işlemlerinin de yeniden tekrarlanması gerekli olmuştur. Bu aşamada ön analiz safhasında sınırlı veri üzerinde yapılan ön-eğitim uygulaması, eldeki derlemin tamamı ve WordPieceTR tokenizer'ı kullanılarak yeniden gerçekleştirilmiştir.

WordPieceTR algoritması kullanılarak modelin tokenizer bileşeni üzerinde değişiklik yapılmış olduğundan modelin tüm girdileri değişmektedir. Bu nedenle modele yönelik ön-eğitim işlemlerinin de en baştan tekrarlanması gerekmiş ve dönem için de en çok zaman alan çalışmalar bu çalışmalar olmuştur. Bundan sonraki süreçte ön-eğitim aşamalarında toplam boyutu 27 GB olan tekilleştirilmiş "OSCAR 2019 Turkish" derleminin (Ortiz Suárez et al., 2020) tamamı kullanılmıştır. Ön-eğitim aşaması çok fazla işlemci kaynağı gerektiren bir çalışma olduğundan makul sürelerde anlamlı sonuçlara ulaşabilmek için nümerik hesaplamalarda TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezi'nce işletilen TRUBA GPU (CUDA) kaynakları (akya, barbun, palamut) ile Google ve Kaggle tarafından sağlanan TPU ve GPU kaynaklarından faydalanılmıştır.

Eğitimler sonucunda biri orjinal BERT modeline dayanan ve diğeri WordPieceTR algoritması ile Türkçe için geliştirilmiş bir model olan BERT-TR olarak ön-eğitimi tamamlanmış olan 2 ayrı model elde edilmiştir. Elde edilen bu 2 model kullanılarak daha önceki dönemde 7 etikete indirgenmiş BIO şemasına göre yapılan VİT ince ayar eğitim çalışması tekrarlanmış ve Tablo 4.3 sonuçlara ulaşılmıştır.

Elde edilen Türkçe için iyileştirilmiş BERT-TR modelinin, WordPiece algoritması kullanan orijinal BERT modeline göre VİT başarımına yönelik +%0,78 olumlu etkisi olduğu görülmüştür.

Elde edilen ümit verici sonuçlardan sonra BERT-TR modelinin sağlayabildiği avantajların daha iyi analiz edilebilmesi amacıyla, elde edilen sonuçların gözden geçirilerek, varsa geliştirilen çözümlerdeki olası yazılımsal hata ve bugların ayıklanarak temizlenmesi, aynı süreç ve uygulamaların tekrarlanması ile aynı sonuçların tutarlı şekilde üretilebildiğinin görülerek sonuçların doğrulanması ve böylece sonuçların sonuçların validasyonu sağlanmıştır.

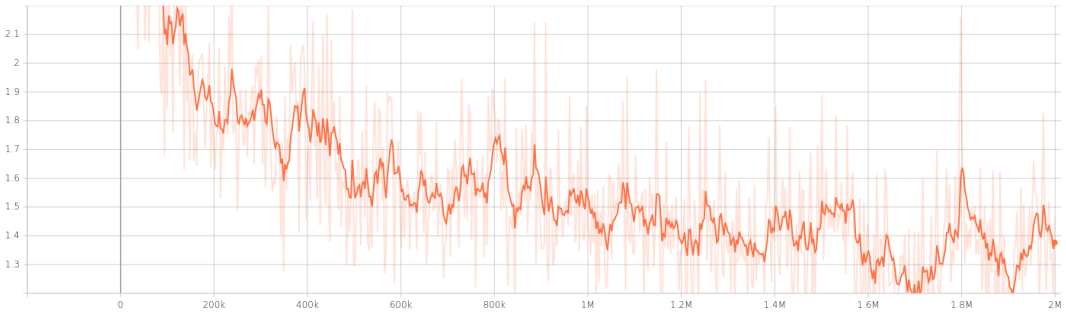
Bir önceki dönemde yapılan çalışmalardaki başarımın ile güncel çalışmalarının sonuçlarının genel olarak tutarlı olduğunun görülmesi sonrasında elde edilen sonuçların yorumlanarak kalitatif analiz yapılabilmesi maksadıyla önceki dönemde kullanılan aynı veri seti üzerinde orjinal BERT ve daha önce implementasyonu tamamlanmış olan aynı algoritmalar kullanılarak farklı epoch sayıları ile eğitmeler gerçekleştirilmiş ve her bir epoch seviyesinde modellerin başarımı yeniden ölçülerek kıyaslanmıştır.

Orjinal WordPiece (BERT) ve genişletilmiş WordPieceTR (BERT-TR) algoritması ile eğitilmiş yeni modeller kullanılarak daha önceki dönemde yapılan VİT çalışmasına benzer çalışmalar farklı epoch seviyelerinde yeniden tekrarlanmıştır.

Bu çalışmalarda elde edilmiş olan modelin tutarlı bir şekilde eğitim sağladığı, öğretim verisine aşırı uyum sağlamadığı kayıp ve validasyon kontrolü ile sürekli kontrol edilmiş (Bkz. Şekil 4.2) ve 7 etikete indirgenmiş BIO şemasına göre yapılan testlerde aşağıdaki sonuçlara ulaşılmıştır.

Elde edilmiş olan başarım durumu ve bu durumdaki değişimler incelendiğinde, elde edilen iyileştirilmiş modelin, WordPiece algoritması kullanan orijinal BERT modeline göre VİT başarımına yönelik +%0,23 - +%3,91 oranında değişen seviyede olumlu etkisi olabildiği görülmüştür.

Bununla birlikte yukarıdaki grafikte de açıkça görüleceği üzere BERT-TR



Şekil 4.2: BERT-TR modeli eğitim sürecindeki kayıp grafiği

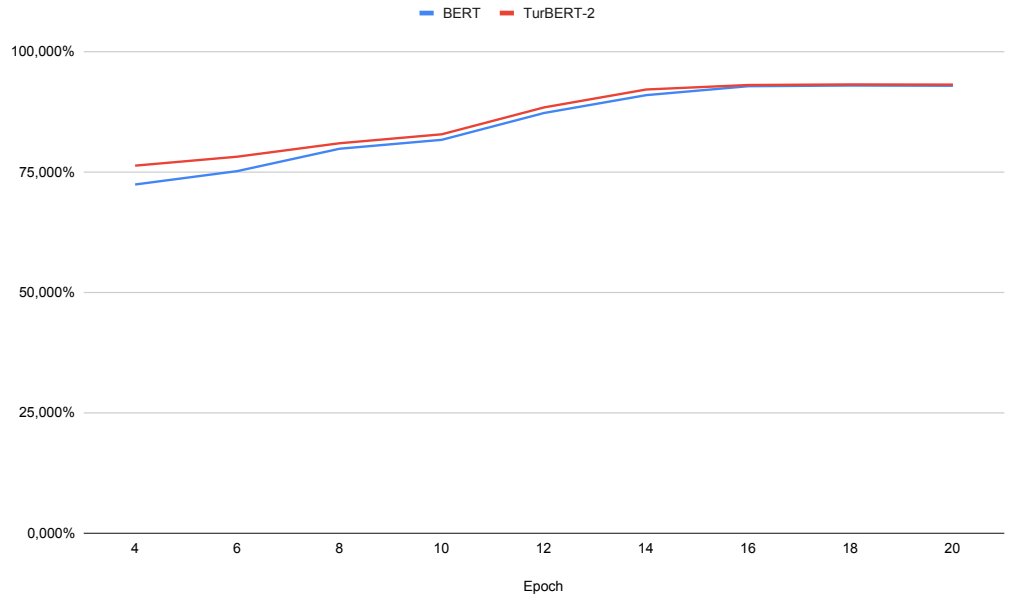
Epoch	**BERT (F1)**	**BERT-TR (F1)**	**Delta**
4	72,368%	76,282%	3,91%
6	75,151%	78,143%	2,99%
8	79,790%	80,934%	1,14%
10	81,646%	82,794%	1,15%
12	87,213%	88,376%	1,16%
14	90,924%	92,097%	1,17%
16	92,779%	93,027%	0,25%
18	92,919%	93,148%	0,23%
20	92,872%	93,111%	0,24%

Tablo 4.8: BERT ve BERT-TR modelleri ile 7 etikete indirgenmiş BIO şemasına göre farklı epoch seviyerlerinde yapılan VİT çalışması

modelinin standart BERT modelinin başarımına göre farkı epoch sayılarının düşük olduğu dönemlerde daha yüksekken epoch sayıları yükseldikçe aradaki fark kapanıyor görünmektedir. Bu durum WordPieceTR tokenizerının daha kolay veya daha hızlı bir şekilde ince ayar (finetune) kabul eden bir model üretilmesine imkan sağladığını göstermektedir.

4.4 BERT-TR-2 Modeli

Çalışmanın son safhasında BERT-TR modelinde el yordamı ile seçilen kurallar yerine daha kapsayıcı bir çözüm elde edilmesi hedeflenmiştir. Bu kapsamda mevcut WordPieceTR tokenizer'ı kelime sonlarında geçen ve büyük ünlü uyumu nedeniyle dönüşüme uğrayan tüm heceleri dikkate alacak şekilde



Şekil 4.3: Farklı epoch sayılarında eğitilmiş BERT ve BERT-TR modellerinin başarım grafiği

revize edilmiştir. Tokenizer'ın revize edilmiş olması token örüntüsünde değişime neden olacağından ön-eğitim işlemlerinin tekrarlanması gerekli olmuştur.

WordPieceTR-2 tokenizerında özel olarak belirlenmiş kurallar yerine tokenize edilen tüm kelimelerdeki son hecelerde büyük ünlü uyumuna uyma özelliği aranmış uyan durumlar için WordPieceTR'de olduğu gibi genelleştirilmiş tek bir token kullanılmıştır.

Elde edilen yeni WordPieceTR-2 tokenizer'ı ile düzenlenmiş model (BERT-TR) ile ön-eğitim yapılmış ve yeni bir model daha edilmiştir. Elde edilen yeni model kullanılarak daha önceki dönemde 7 etikete indirgenmiş BIO şemasına göre yapılan VİT çalışması tekrarlanmış ve Tablo 4.4 sonuçlara ulaşılmıştır.

İkinci safha çalışmalarında hem derlem boyutunun büyütülmesi hem de Modelde Türkçe'ye özgü yapılabilecek ilave iyileştirmeler ile daha da yüksek bir başarı elde edilebileceği değerlendirilerek WordPieceTR tokenizasyon algoritmasının daha geniş bir kural setini içerecek şekilde geliştirilmesi neticesinde başarımda gözle görülür bir iyileşme oluştuğu görülmüştür.

Model	**Süre (dk.)**	**Epoch**	**F1**	**Kesinlik**	**Duyarlılık**
BERT	62882	16	92,786%	92,751%	92,821%
BERT-TR	71212	16	93,038%	93,038%	93,143%

Tablo 4.9: BERT ve BERT-TR modelleri ile 7 etikete indirgenmiş IOB şemasına göre tekrarlanan VİT çalışması

Model	**F1**	**Kesinlik**	**Duyarlılık**
BERT	92,703%	95,367%	91,369%
BERT-TR-2	93,144%	95,539%	91,965%

Tablo 4.10: BERT ve BERT-TR-2 modellerinin sonuçlarının karşılaştırılması

Elde edilen modellerin performansını daha kapsamlı bir şekilde değerlendirmek için modellerin tahminleri ile gerçek değerler arasındaki ilişkiyi görsel olarak da açıklayan karışıklık matrisleri (Confusion Matrix) Şekil 4.4) ve Şekil 4.5 de verilmiştir.

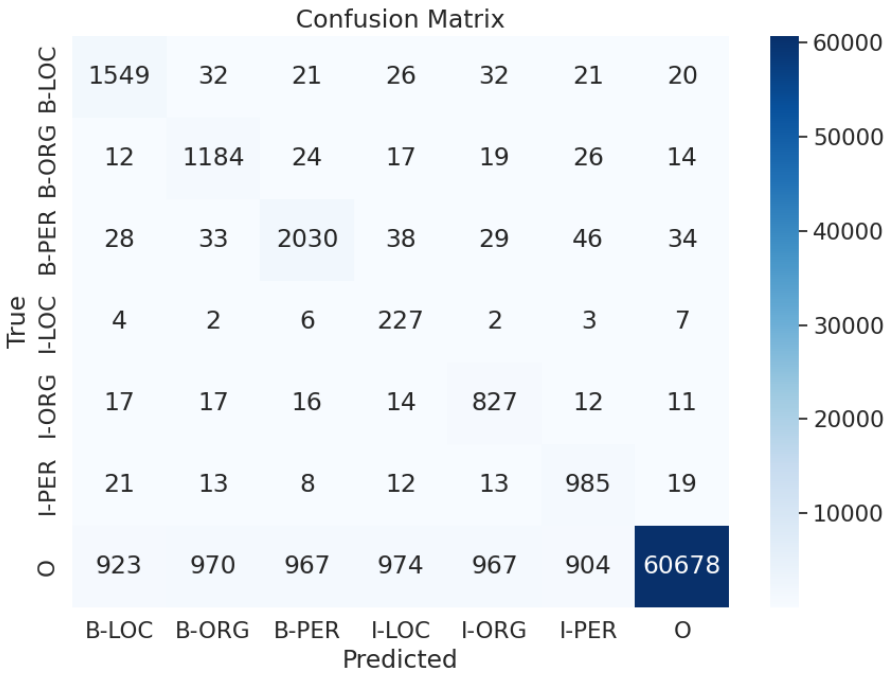
VİT etiketleme görevi sonunda oluşan hatalar her iki model için de incelendiğinde;

1. Tipik olarak BERT ve BERT-TR modellerinin benzer noktalarda benzer hatalar yaptıkları görülmektedir. Bununla birlikte Derlem etiketlemesinin doğruluk oranının da sorgulmaya açık olduğu görülmüştür.

2. BERT'nin hata yaptığı ancak BERT-TR'nin doğru sonucu verebilmiş olduğu tipik bazı örneklerde BERT'in hatalarının çoğunlukla B- yerine I- etiketi kullanmış olması olduğu görülmüştür.

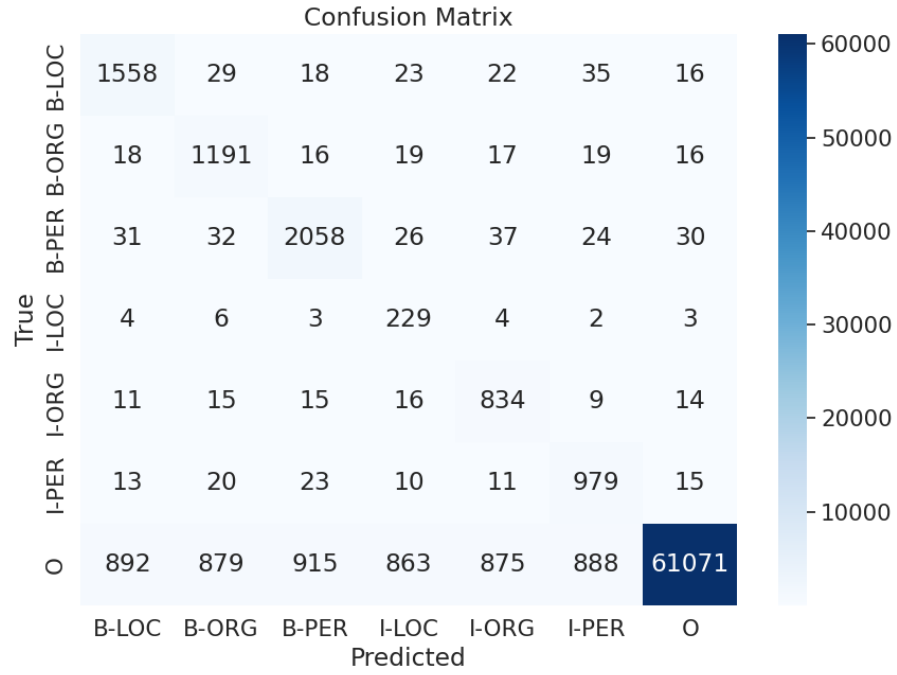
3. BERT'nin hata yaptığı ancak BERT-TR'nin doğru sonuçları incelendiğinde birisinin diğerine baskın şekilde üstün geldiği bir etiket tipi görülmemiştir.

4. BERT'nin ve BERT-TR'nin hataları incelendiğinde sürekli olarak belirli bir etiketin eksik veya fazla şekilde hatalı kullanılması, bir varlık isminin sürekli yanlış etiketlenmesi (belirli bir bir örgütün sürekli kişi olarak



Şekil 4.4: BERT-TR modelinin sonuçlarına ilişkin karışıklık matrisi

etiketlenmesi vb.) gibi hatalara denk gelinmemiştir.



Şekil 4.5: BERT-TR modelinin sonuçlarına ilişkin karışıklık matrisi

5 SONUÇ

Bu bölümde tezin kısa bir özetine ve literatüre yapılan katkılara değinilerek, tez çalışmasında elde edilen sonuçlar tartışılacak, ileride yapılabilecek çalışmalara yönelik öneriler ve açık problem sahalarından bahsedilecektir.

5.1 Katkılar ve Tartışma

Türkçe arkaplan bölümünde detaylarına değinilen zengin özelliklere sahiptir. Bu çalışmada temelde türkçenin büyük sesli uyumu özelliğini dikkate alacak şekilde iyileştirilmiş bir tokenizer geliştirilmiştir. Bir tokenizer'ın çalışma mantığı gereği, encode edilen verinin decode edilebilmesi ve decode edildiğinde orjinal veriye deterministik bir şekilde dönüş yapılabilmesi gerekmektedir. Bu çalışma kapsamında deterministik olarak büyük ünlü uyumuna ilişkin kurallar WordPieceTR tokenizer algoritmasına dahil edilmiştir. BERT-TR modelinin eğitimi sırasında bu tokenizer kullanıldığından tokenizer'a dahil edilen dile özgü özelliğin BERT-TR modeline yansıtılması da mümkün olmuştur. İleride yapılacak farklı çalışmalarda Türkçe'ye özgü farklı özelliklerin

deterministik bir şekilde ifade edilebilmesi veya tokenizer içine dahil edilebilmesi durumunda dile özgü öznitelikleri daha iyi seviyede yansıtabilen farklı modeller oluşturulabilecektir.

Bununla birlikte Türkçe'ye özgü tüm öznitelikler için deterministik kurallar ortaya konulması mümkün olmayabilir. Örnekleri arkaplan bölümünde de verilen bazı durumlarda bir ek biçimi birden fazla anlama gelebilmektedir. Deterministrik kurallarla çözülemeyecek durumların tokenizasyon aşamasında değil sinir ağı katmanları içinde algılanabilecek örüntüler olarak ele alınması çoğunlukla daha uygun bir yaklaşım olacaktır. Bu çalışmada farklı anlamlara gelebilecek tokenlara ait olası tüm anlamların ayrı tokenlar ile temsil edilmesi gibi bir yaklaşıma gidilmemiştir. Ancak bu gibi durumların tokenizasyon aşamasında ayırt edilmeye çalışılması ile daha iyi sonuçlar elde edilip edilemeyeceği incelemeye açık bir konudur.

Bu çalışmada elde ettiğimiz sonuçlar dikkate alındığında, Türkçe'nin kendine özgü dil kuralları ve yapısının, dil bağımsız modellerin tam potansiyelini ortaya çıkarmasını zorlaştırabildiği ve daha uzun eğitim süreleri oluşmasına neden olduğu ifade edilebilir. Dil bağımsız modellerin son dönemde ortaya koydukları yüksek başarımın göz ardı edilmesi mümkün olmamakla birlikte, özellikle Türkçe gibi zengin özelliklere sahip diller için özelleştirilmiş modellerin geliştirilmesine yönelik çalışmaların daha hızlı veya kolay eğitilebilen modellerle daha iyi performans ve doğruluk sağlamak için halen önemini koruduğu görülmektedir.

5.2 Açık Problemler ve Öneriler

VİT problemi özelinde, temsil katmanlarında ortaya çıkarılan tokenlere ilave olarak, örneğin kelimelerin ilk harfinin büyük olması, kısaltmalardan sonra nokta konması, tüm harflerin büyük olması gibi biçimsel desen ve özelliklerinin genelleştirilerek dilden bağımsız bir ayrı bir özellik seti şeklinde ayrı bir özellik vektörü olarak tanımlanması ve kullanılan sinir ağı modeline mevcut tokenler dışında ayrı/ilave bir giriş olarak verilmesi, (Xu et al., 2021) modelin daha fazla bağlam ve bilgiyi kapsamasına imkan sağlayarak Türkçe

özelinde VİT başarımını daha da artırmak için uygulanabilecek bir yöntem olabilir.

Literatürde BERT modelleri genelde küçük büyük harf ayrımı yapan ve yapmayan olarak iki farklı türdeki modeller halinde eğitilmekte bu şekilde sunulmaktadır. Bu çalışmada ise zaman kısıtları nedeniyle VİT problemi alanında daha başarılı sonuçlar vereceği değerlendirilen ve küçük büyük harfi ayrımı yapan tek bir bir model eğitilmiştir. Küçük büyük harf ayrımı yapmayan modeller de eğitilerek farklı problem sahalarındaki başarımları incelenebilir.

Bu çalışmada yöntem bölümünde detayları anlatılmış olan desterministik yöntemlerle Türkçe'nin temel ses uyumu özelliklerini dikkate alan yeni bir tokenizer geliştirmiştir. Eklerin ayrı ayrı analiz edilmesi veya ek gövde ayrımı yapılması ya da sessiz harflarde de meydana gelen dönüşümler gibi özellikler çalışmanın kapsamı dışında tutulmuştur. Bu tarz ilave özellikleri de dikkate alarak veya gelişmiş bir biçimbilimsel çözümleyici ile birlike daha kapsayıcı kurallar oluşturularak Türkçe özelinde olumlu yönde daha büyük etkiler oluşturabilecek daha gelişmiş bir tokenizer ortaya konulabilir. Ancak bu tarz iyileştirme çalışmalarında Türkçe'deki bazı kuralların yabancı dillerden dilimize geçmiş kelimeler gibi özellikle yazılışı ile okunuşu farklı olabilen kelimeler için istisnai durumlarının olabileceği unutulmamalıdır.

Yapılan çalışma neticesinde elde edilen BERT-TR modelleri bu tez kapsamı çerçevesinde sadece VİT görevi özelinde incelenmiş ve değerlendirilmiştir. Bununla birlikte bu modellerin esasen sadece VİT görevi özelinde kullanılması zorunlu bir durum değildir. Bu ve benzeri modellerin Türkçe özelinde duygudurum tespiti, metin özetleme veya benzeri farklı doğal dil işleme problem sahalarında da kullanılması durumda modellerin başarımına ve/veya eğitim hızlarına katkı sağlaması olasıdır. Benzer çalışmaların farklı doğal dil işleme görevlerinde de uygulanması ile dile özgü özniteliklerin tokenizasyon aşamasında ele alınmasının genel doğal dil işleme görevlerinin sonuca etkisinin birden fazla alanda incelenmesine ve böylece daha kapsamlı bir değerlendirme yapılmasına imkan oluşturulabilir.

Kaynaklar

Akalın, Ş. H.: 2009, Türk Dili: Dünya Dili, *Türk Dili* **(687)**, 195

Akbik, A., Blythe, D., and Vollgraf, R.: 2018, Contextual string embeddings for sequence labeling, *arXiv preprint arXiv:1708.00107*

Aksakallı, V. and Türk, E.: 2019, Evaluation of pre-trained language models for turkish named entity recognition, in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 1–4 pp

Alfonseca, E. and Manandhar, S.: 2002, An unsupervised method for general named entity recognition and automated concept discovery, in *Proceedings of the 1st İnternational Conference on General WordNet, Mysore, India*, 34–43 pp, WordNet kullanılmış.

Altintas, E., Karsligil, E., and Coskun, V.: 2005, The Effect of Windowing in Word Sense Disambiguation, in D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, p. Yolum, T. Güngör, F. Gürgen, and C. Özturan (eds.), *Computer and Information Sciences - ISCIS 2005*, Vol. 3733, Springer Berlin Heidelberg, 626–635 pp, Berlin, Heidelberg

Bengio, Y., Courville, A., and Vincent, P.: 2013, Representation learning: A review and new perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35(8)**, 1798

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T.: 2016, Enriching Word Vectors with Subword Information, *arXiv:1607.04606 [cs]*, arXiv: 1607.04606

Comment: Accepted to TACL. The two first authors contributed equally

Bolucu, N., Akgol, D., and Tuc, S.: 2019, Bidirectional LSTM-CNNs with Extended Features for Named Entity Recognition, *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)* 1–4 pp, [TLDR] Evaluation shows that adding syntactic

and semantic information about words without feature engineering to the model surpasses the baseline model tested on CoNLL-2003 English NER dataset.

Borthwick, A.: 1996, Automatic acquisition of domain knowledge for information extraction, in *Proceedings of the 16th İnternational Conference on Computational Linguistics (COLING-1996)*, 199–204 pp

Bostrom, K. and Durrett, G.: 2020, Byte Pair Encoding is Suboptimal for Language Model Pretraining

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D.: 2015, A fast unified model for parsing and sentence understanding, *Proceedings of the 2015 conference on empirical methods in natural language processing* 1321–1331 pp

Çelikkaya, G., Torunoğlu, D., and Eryiğit, G.: 2013, Named entity recognition on real data: A preliminary investigation for Turkish, in *Application of Information and Communication Technologies (AICT), 2013 7th International Conference On*, IEEE, 1–5 pp

Chen, H., Huang, S., Chiang, D., Dai, X., and Chen, J.: 2018, Combining Character and Word Information in Neural Machine Translation Using a Multi-Level Attention, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, Association for Computational Linguistics, 1284–1293 pp

Chinchor, N.: 1998, Overview of MUC-7/MET-2, in *Proceedings of the 7th Message Understanding Conference (MUC-7)*, 21–31 pp

Chitnis, R. and DeNero, J.: 2015a, Variable-Length Word Encodings for Neural Translation Models, *Conference on Empirical Methods in Natural Language Processing*

Chitnis, R. and DeNero, J.: 2015b, Variable-length word encodings for neural translation models, in *Proceedings of the 2015 Conference*

on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2084–2089 pp

Chiu, J. P. and Nichols, E.: 2016, Named Entity Recognition with Bidirectional LSTM-CNNs, *Transactions of the Association for Computational Linguistics* **4**, 357, src: https://github.com/mshehrozsajjad/Named-Entity-Recognition-with-Bidirectional-LSTM-CNNs

Chomsky, N.: 2006, *Language and Mind*, Cambridge University Press, Cambridge ; New York, 3rd ed edition

Church, K. W.: 2020, Emerging trends: Subwords, seriously?, *Natural Language Engineering* **26(3)**, 375

Clark, K., Luong, M.-T., and Le, Q. V.: 2020, ELECTRA: Pre-training text encoders as discriminators rather than generators, in *International Conference on Learning Representations*

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P.: 2011, *Natural Language Processing (Almost) from Scratch*

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ..., and Lample, G.: 2019, Unsupervised cross-lingual representation learning at scale, *arXiv preprint arXiv:1911.02116*

Cotterell, R. and Schütze, H.: 2015, Morphological Word-Embeddings, in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, Association for Computational Linguistics, 1287–1292 pp

Dalkılıç, G. and Çebi, Y.: 2005, Zipf's Law and Mandelbrot's Constants for Turkish Language Using Turkish Corpus (TurCo), in T. Yakhno (ed.), *Advances in Information Systems*, Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, 273–282 pp

Demir, H. and Ozgur, A.: 2014, Improving Named Entity Recognition for Morphologically Rich Languages Using Word Embeddings, in *2014 13th*

International Conference on Machine Learning and Applications, Detroit, MI, USA, IEEE, 117–122 pp

DemiRci, K.: 2011, On the Importance of the Phoneme Theory, *Journal of Turkish Studies* **Volume 6 Issue 2(6)**, 359

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: 2018, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv:1810.04805 [cs]*, arXiv: 1810.04805
BERT

Domingo, M., Garcia Martinez, M., Helle, A., and Casacuberta, F.: 2018, *How Much Does Tokenization Affect in Neural Machine Translation?*

Dridan, R. and Oepen, S.: 2012, *Tokenization: Returning to a Long Solved Problem: A Survey, Contrastive Experiment, Recommendations, and Toolkit*, Vol. 2

Eken, B. and Tantug, A. C.: 2015, Recognizing Named Entities in Turkish Tweets, in *Computer Science & Information Technology (CS & IT)*, Academy & Industry Research Collaboration Center (AIRCC), 155–162 pp

Ergenç, İ.: 2000, Dilin beyindeki organizasyonu ve konuşmanın gerçekleşmesi, *Multidisipliner yaklaşımla beyin ve kognisyon.(Editörler: Karakaş, Aydın, Erdemir ve Özemsi) Ankara: Çizgi Tıp Yayınevi*

Eryiğit, G.: 2016, State of the art in Turkish Named Entity Recognition 1, [TLDR] The introduced approach uses conditional random fields and obtains the highest results in the literature for Turkish NER with 92% CoNLL score on a dataset collected from Turkish news articles and ∼65% on different datasets collected from Web 2.0.

Friedman, R.: 2023, Tokenization in the Theory of Knowledge, *Encyclopedia* **3(1)**, 380

Göksel, A. and Kerslake, C.: 2005, *Turkish: A Comprehensive Grammar*, Routledge Comprehensive Grammars, Routledge, London, 1. publ edition

Goldberg, Y.: 2017, Neural network methods for natural language processing, *Synthesis Lectures on Human Language Technologies* **10(1)**, 1

Goldsmith, J.: 2001, Unsupervised learning of word segmentation rules with genetic algorithms, in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, ACL, goldsmith2001unsupervised p.

Gonenc, G.: 1973, Unique Decipherability of Codes With Constraints With Application to Syllabification of Turkish Words, in *COLING 1973 Volume 1: Computational And Mathematical Linguistics: Proceedings of the International Conference on Computational Linguistics*

Gowda, T. and May, J.: 2020, Finding the Optimal Vocabulary Size for Neural Machine Translation, in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online, Association for Computational Linguistics, 3955–3964 pp

Goyal, A., Gupta, V., and Kumar, M.: 2018, Recent Named Entity Recognition and Classification techniques: A systematic review, *Computer Science Review* **29**, 21

Gungor, O., Uskudarli, S., and Gungor, T.: 2018, Recurrent neural networks for Turkish named entity recognition, in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, Izmir, IEEE, 1–4 pp

Haspelmath, M. and Sims, A. D.: 2010, *Understanding Morphology*, Understanding Language Series, Hodder Education, London, 2nd ed edition

Hirschmann, F., Nam, J., and Fürnkranz, J.: 2016, What makes word-level neural machine translation hard: A case study on english-german translation, in *International Conference on Computational Linguistics*

Hochreiter, S. and Schmidhuber, J.: 1997, Long short-term memory, *Neural computation* **9(8)**, 1735

Huang, Z., Xu, W., and Yu, K.: 2015, Bidirectional LSTM-CRF models for sequence tagging, *arXiv preprint arXiv:1508.01991*

İlgen, B., Adalı, E., and Tantuğ, A. C.: 2013, *A Comparative Study to Determine the Effective Window Size of Turkish Word Sense Disambiguation Systems*, Springer, 233 Spring Street, New York, Ny 10013, United States

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T.: 2016, FastText.zip: Compressing text classification models, *arXiv:1612.03651 [cs]*, arXiv: 1612.03651

Kazakov, D. and Manandhar, S.: 2000, Unsupervised Learning of Word Segmentation Rules with Genetic Algorithms and Inductive Logic Programming, *Machine Learning* 43

Kim, Y.: 2014, Convolutional neural networks for sentence classification, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751 pp

Korkmaz, Z.: 2017, *Türkiye Türkçesi Grameri Şekil Bilgisi*, Türk Dil Kurumu Yayınları

Koroteev, M. V.: 2021, *BERT: A Review of Applications in Natural Language Processing and Understanding*, Comment: 18 pages, 7 figures

Küçük, D., Arıcı, N., and Küçük, D.: 2017, *Named Entity Recognition in Turkish: Approaches and Issues*

Küçük, D. and Yazici, A.: 2010, A Hybrid Named Entity Recognizer for Turkish with Applications to Different Text Genres, in *Lecture Notes in Electrical Engineering*, Vol. 62, 113–116 pp

Kudo, T.: 2018, Subword regularization: Improving neural network translation models with multiple subword candidates, *CoRR* abs/1804.10959, arXiv:1804.10959 [cs]
Comment: Accepted as a long paper at ACL2018

Kudo, T. and Richardson, J.: 2018, SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, in *Proceedings of the 2018 Conference on Empirical Methods in*

Natural Language Processing: System Demonstrations, Brussels, Belgium, Association for Computational Linguistics, 66–71 pp

Kuru, O., Can, O. A., and Yuret, D.: 2016, CharNER: Character-Level Named Entity Recognition, in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, The COLING 2016 Organizing Committee, 911–921 pp

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R.: 2020, ALBERT: A lite BERT for self-supervised learning of language representations, in *International Conference on Learning Representations*

Leaman, R. and Gonzalez, G.: 2008, Named entity recognition and beyond: A prototype for domain-specific information extraction, *Transactions of the Association for Computational Linguistics* **4**, 231

LeCun, Y., Bengio, Y., and Hinton, G.: 2015, Deep learning, *Nature* **521(7553)**, 436

Li, J., Sun, A., Han, J., and Li, C.: 2022, A Survey on Deep Learning for Named Entity Recognition, *IEEE Transactions on Knowledge and Data Engineering* **34(1)**, 50, arXiv: 1812.09449
Comment: 20 pages, 12 figures, 3 tables

Liu, P., Guo, Y., Wang, F., and Li, G.: 2022, Chinese named entity recognition: The state of the art, *Neurocomputing* **473**, 37

Liu, Y., Li, M., Huang, Y., and Zhao, J.: 2019a, BERT for domain-specific question answering, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4395–4405 pp

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., and Stoyanov, V.: 2019b, RoBERTa: A robustly optimized BERT pretraining approach, *arXiv preprint arXiv:1907.11692*

Luoma, J., Chang, L.-H., Ginter, F., and Pyysalo, S.: 2021, Fine-grained Named Entity Annotation for Finnish, in *Proceedings of the 23rd*

Nordic Conference on Computational Linguistics (NoDaLiDa), Reykjavik, Iceland (Online), Linköping University Electronic Press, Sweden, 135–144 pp

McCann, B., Bradbury, J., Xiong, C., and Socher, R.: 2017, Learned in translation: Contextualized word vectors, in *Advances in Neural İnformation Processing Systems*, 6297–6308 pp

Mikolov, T., Chen, K., Corrado, G., and Dean, J.: 2013a, Efficient Estimation of Word Representations in Vector Space, *arXiv:1301.3781 [cs]*

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J.: 2013b, Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems* **26**, 3111

Nadeau, D. and Sekine, S.: 2007, A survey of named entity recognition and classification, *Lingvisticae Investigationes* **30(1)**, 3

Nivre, J. and Hall, J.: 2008, Integrating graph-based and transition-based dependency parsers, in *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, Association for Computational Linguistics, 607–615 pp

Oflazer, K.: 2018, *Türkçe Doğal Dil İşleme*

Ortiz Suárez, P. J., Romary, L., and Sagot, B.: 2020, A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, Association for Computational Linguistics, 1703–1714 pp

Park, K., Lee, J., Jang, S., and Jung, D.: 2020, *An Empirical Study of Tokenization Strategies for Various Korean NLP Tasks*, Comment: Accepted to AACL-IJCNLP 2020

Park, S., Byun, J., Baek, S., Cho, Y., and Oh, A.: 2018, Subword-level Word Vector Representations for Korean, in *Proceedings of the 56th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, Association for Computational Linguistics, 2429–2438 pp

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L.: 2018, Deep contextualized word representations, *arXiv preprint arXiv:1802.05365*

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I.: 2018, Improving language understanding by generative pre-training, *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language understanding paper. pdf*

Sanh, V., Debut, L., Chaumond, J., and Wolf, T.: 2019, DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter, *arXiv preprint arXiv:1910.01108*

Schmidhuber, J.: 2015, Deep learning in neural networks: An overview, *Neural Networks* **61**, 85

Schweter, S.: 2020, *BERTurk - BERT Models for Turkish*, Zenodo

Şeker, A. and Köse, H.: 2020, Turkish named entity recognition with BERT-CRF, in *Proceedings of the 6th İnternational Symposium on İnnovative Approaches in Applied İnformatics*, Springer, 288–296 pp

Sennrich, R., Haddow, B., and Birch, A.: 2016, Neural Machine Translation of Rare Words with Subword Units, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, Association for Computational Linguistics, 1715–1725 pp

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., and Wang, H.: 2019, ERNIE 2.0: A continual pre-training framework for language understanding, *arXiv preprint arXiv:1907.12412*

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., and Wang, H.: 2021, ERNIE: Enhanced language representation with informative entities, *arXiv preprint arXiv:1905.07129*

Taher, E., Hoseini, S. A., and Shamsfard, M.: 2020, *Beheshti-NER: Persian Named Entity Recognition Using BERT*

Taş, İ.: 2021, Elision In Turkish, *Karabük Türkoloji Dergisi* **3(3)**, 44

Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., Poulton, A., Kerkez, V., and Stojnic, R.: 2022, *Galactica: A Large Language Model for Science*, Bu makale gerçekten çok ilginç.

Temizyurek, F.: 2007, Çocukta Dil Gelişim Süreci, *Hacettepe Üniversitesi Türkiyat Araştırmaları (HÜTAD)* 8 p.

Tjong Kim Sang, E. F.: 2002, Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition, in *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*

Tjong Kim Sang, E. F. and De Meulder, F.: 2003, Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003* -, Vol. 4, Edmonton, Canada, Association for Computational Linguistics, 142–147 pp

Turing, A. M.: 1950, Computing Machinery and Intelligence, *Mind* **LIX(236)**, 433

Türkoğlu, M. F. and Iscan, O. F.: 2021, Named entity recognition for turkish with deep learning, in *Proceedings of the İnternational Conference on Applied Analysis and Mathematical Modeling*, Springer, 231–235 pp

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.: 2017, Attention is all you need, in *Advances in Neural İnformation Processing Systems*, 5998–6008 pp

Vikipedi: 2023, *Vikipedi, Türkçe*, Page Version ID: 29129310

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F., and Pyysalo, S.: 2019, *Multilingual İs Not Enough: BERT for Finnish*

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: 2016, Google's neural machine translation system: Bridging the gap between human and machine translation, *arXiv preprint arXiv:1609.08144*

Xu, L., Jie, Z., Lu, W., and Bing, L.: 2021, *Better Feature Integration for Named Entity Recognition*, Comment: Accepted by NAACL 2021. 13 pages, 9 Figure, 10 Tables

Yadav, V. and Bethard, S.: 2018, A Survey on Recent Advances in Named Entity Recognition from Deep Learning models, in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, Association for Computational Linguistics, 2145–2158 pp

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V.: 2019, XLNet: Generalized autoregressive pretraining for language understanding, *arXiv preprint arXiv:1906.08237*

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E.: 2016, Hierarchical attention networks for document classification, in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 1480–1489 pp

Young, T., Hazarika, D., Poria, S., and Cambria, E.: 2018, Recent trends in deep learning based natural language processing, *IEEE Computational Intelligence Magazine* **13(3)**, 55

Yu, S., Xu, C., and Liu, H.: 2018, *Zipf's Law in 50 Languages: İts Structural Pattern, Linguistic İnterpretation, and Cognitive Motivation*, arXiv:1807.01855 [cs]

Zhang, X., Zhao, J., and LeCun, Y.: 2015, Character-level Convolutional Networks for Text Classification, *arXiv:1509.01626 [cs]*, Comment: An early version of this work entitled "Text Understanding from Scratch" was posted in Feb 2015 as arXiv:1502.01710. The present paper has considerably more experimental results and a rewritten introduction, Advances in Neural Information Processing Systems 28 (NIPS 2015)

Zheng, B., Dong, L., Huang, S., Singhal, S., Che, W., Liu, T., Song, X., and Wei, F.: 2021, Allocating Large Vocabulary Capacity for Cross-Lingual Language Model Pre-Training, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic, Association for Computational Linguistics, 3203–3215 pp

Zulfikar, H.: 2013, Çokluk Kavramı ve -lar (-ler) Ekinin Kullanımı, *Türk Dili*

TEŞEKKÜR

Bu çalışma ile benden önce yola çıkan nicelerinin izlerini takip etmeye çalışarak Türkçe Doğal Dil İşleme literatürüne küçük bir katkı sağlamayı hedefledim.

Bu hedefe ulaşmam için bana yolculuğumun başından itibaren sağlıklı bir rota çizen ve bu rotada ilerleyebilmek için gereken her türlü desteği sağlayan danışmanım Sayın Prof.Dr.Oğuz DİKENELLİ'ye ve yolculuğum boyunca geri bildirimleri ile önümdeki engelleri görmeme ve aşmama yardımcı olan saygıdeğer hocalarım Prof.Dr. Yalçın ÇEBİ, Prof.Dr. Murat Osman ÜNALIR ve sayın Doç.Dr. Selma TEKİR'e teşekkürü bir borç biliyorum.

Ayrıca bugüne kadar üzerimde emeği olan bütün hocalarıma, hayatım boyunca varlıklarıyla bana güç veren anneme ve babama, akademi dışında da profesyonel hayatı olan kişiler için daha da güçlü odaklanma becerileri gerektiren bu zorlu yolculukta bana her zaman moral ve motivasyon kaynağı olan eşime ve çocuklarıma teşekkür ederim.

Bu araştırmanın deney aşamalarında uygulanan ve özellikle yüksek başarım gerektiren nümerik hesaplamalar için TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezi'nce işletilen TRUBA GPU (CUDA) kaynakları (akya, barbun, palamut) ile Google ve Kaggle tarafından sağlanan TPU ve GPU kaynaklarından faydalanılmıştır.

11/07/2023

Ergin ALTINTAŞ

ÖZGEÇMİŞ

ÖZLÜK BILGILERI

Adı Soyadı : Ergin ALTINTAŞ

Doğum Tarihi : 1978

Doğum Yeri : İstanbul

Uyruğu : T.C.

EĞİTİM

2003-2005: Yüksek Lisans - Deniz Harp Okulu, Deniz Bilimleri ve Teknolojisi Enstitüsü, Bilgisayar Mühendisliği

1995-1999: Lisans - Deniz Harp Okulu, Bilgisayar Mühendisliği

YAYINLAR

2005 Altintas, E., Karsligil E., Coskun, V., A New Semantic Similarity Measure Evaluated In Word Sense Disambiguation, The 15th Nordic Conference of Computational Linguistics (NODALIDA-2005), May 20-21 2005, Joensuu, Finland.

2005 Altintas, E., Karsligil E., Coskun, V., The Effect of Windowing in Word Sense Disambiguation, The 20 th International Symposium on Computer and Information Sciences (ISCIS'05), 26-28 Oct 2005, Istanbul, Turkey.

İLGİ ALANLARI

Yapay zeka, yapay sinir ağları, derin öğrenme, doğal dil işleme/anlama, siber güvenlik, Linux ve açık kaynak kodlu yazılımlar.

A EK TABLOLAR

Harfler	Farklı Kullanım Örnekleri	Token Örüntüsü
-arak	anlatarak, bilerek, belirleyerek, diyerek, gülerek, zorlayarak, çatırdayarak	##[y]{a,e}r{a|e}k
-madan	bozulmadan, dinlemeden, uğramadan	##m{a|e}d{a|e}n
-maksızın	gecikmeksizin, yorulmaksızın, beklemeksizin, düşünmeksizin	##m{a|e}ks{ı|i}z{ı|i}n
-ınca	başlayınca, gelince, çalınca, olunca, ölünce	##[y]{u,ı|ü,i}nc{a|e}
-dıkça	araştırdıkça, gittikçe, ilerledikçe, gördükçe,okudukça	##{d|t}{u,ı|ü,i}kç{a|e}
-casına	istercesine, parçalarcasına, anlamışçasına	##{c|ç}{a|e}s{ı|i}ne
-maz	paslanmaz, solmaz, unutulmaz	##m{a|e}z
-asıya	gelesiye, doyasıya, ölesiye, öldüresiye	##[y]{a|e}s{ı|i}y{a|e}
-cak	evcek, çabucak	##c{a|e}k
-lık	vatandaşlık, sebzelik, dutluk, çiftlik, önlük	##l{u,ı|ü,i}k
-sız	ilgisiz, alakasız, korkusuz, ödünsüz	##s{u,ı|ü,i}z
-lar	kitaplar, kalemler	##l{a|e}r
-ırıp	bitirip, kaçırıp, götürüp, uçurup	##{u|ı|ü|i}r{u|ı|ü|i}p
-alı	çıkalı, gideli	##[y]{a|e}l{ı|i}
-cağız	adamcağız, köyceğiz	##c{a|e}ğ{ı|i}z
-sal	duygusal, sezgisel, bölgesel	##s{a|e}l
-ca	akıllıca, sizce, gönlünce, çocukça, delice	##{c,ç}{a|e}
-cı	avcı, oyuncu, simitçi, sütçü	##{c,ç}{u,ı,ü,i}
-lı	alaylı, köylü, istanbullu, üniversiteli	##l{u,ı,ü,i}
-daş	arkadaş, meslektaş, özdeş, sesteş	##{d,t}{a,e}ş
-sı	çocuksu	##s{u,ı,ü,i}
-msi	ekşimsi, acımsı	##ms{u,ı,ü,i}

Tablo A.1: WordPieceTR algoritması kapsamında özel olarak tokenize edilen harf örüntüleri (sonraki sayfada devam eder).

Harfler	**Farklı Kullanım Örnekleri**	**Token Örüntüsü**
-mak	açmak, sevmek	##m{a|e}k
-gan	kırılgan, yapışkan, ötüşken	##{g|k}{a|e}n
-cıl	bencil, evcil, ölümcül, balıkçıl, etçil	##{c,ç}{u,ı,ü,i}l
-layın	akşamleyin, sabahleyin	##l{a|e}y{ı|i}n
-sak	kursak, tümsek	##s{a|e}k
-şın	sarışın, gökşin	##ş{ı|i}n

Tablo A.1: WordPieceTR algoritması kapsamında özel olarak tokenize edilen harf örüntüleri.

Metin girdisi	**VİT Etiketi (BLIOU)**	**VİT Etiketi (BIO)**
Emekli	B-ORG	B-ORG
Sandığı	L-ORG	I-ORG
,	O	O
SSK	U-ORG	B-ORG
ve	O	O
Bağ	B-ORG	B-ORG
-	I-ORG	I-ORG
Kur	L-ORG	I-ORG
'un	O	O
emeklilik	O	O
sigortaları	O	O
Ulusal	B-ORG	B-ORG
Sosyal	I-ORG	I-ORG
Güvenlik	I-ORG	I-ORG
Sigorta	I-ORG	I-ORG
Kurumu	L-ORG	I-ORG
çatısı	O	O
altında	O	O
toplanmalıdır	O	O

Tablo A.2: BILOU kodlamasından BIO kodlamasına dönüşüm örneği

Etiket	**Sıklık**
O	438.701
U-LOC	10.482
U-PER	9.205
B-PER	7.087
L-PER	7.087
U-ORG	5.399
B-ORG	3.784
L-ORG	3.784
I-ORG	3.096
B-LOC	1.234
L-LOC	1.234
I-PER	659
I-LOC	589

Tablo A.3: BILOU şemasına VİT etiketi sıklıkları