İhtiyacınız Olan Tek Şey Dikkat

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Araştırma
nikip@google.com

Jakob Uszkoreit*
Google Araştırma
usz@google.com

Llion Jones*
Google Araştırması
llion@google.com

Aidan N. Gomez*
†Toronto Üniversitesi
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Beyin
lukaszkaiser@google.com

Illia Polosukhin*‡
illia.polosukhin@gmail.com

Özet

Baskın dizi iletim modelleri, bir kodlayıcı ve bir kod çözücü içeren karmaşık tekrarlayan veya evrişimli sinir ağlarına dayanmaktadır. En iyi performans gösteren modeller ayrıca kodlayıcı ve kod çözücüyü bir dikkat mekanizması aracılığıyla birbirine bağlar. Biz, yineleme ve konvolüsyonlardan tamamen vazgeçerek, yalnızca dikkat mekanizmalarına dayanan yeni ve basit bir ağ mimarisi olan Transformer'ı öneriyoruz. İki makine çevirisi görevi üzerinde yapılan deneyler, bu modellerin daha üstün kalitede olduğunu, daha paralelleştirilebilir olduğunu ve eğitmek için önemli ölçüde daha az zaman gerektirdiğini göstermektedir. Modelimiz WMT 2014 İngilizceden Almancaya çeviri görevinde 28,4 BLEU elde ederek, topluluklar da dahil olmak üzere mevcut en iyi sonuçlara göre 2 BLEU'nun üzerinde bir iyileşme sağlamıştır. WMT 2014 İngilizceden Fransızcaya çeviri görevinde modelimiz, literatürdeki en iyi modellerin eğitim maliyetlerinin küçük bir kısmı olan sekiz GPU'da 3,5 gün boyunca eğitim aldıktan sonra 41,8'lik yeni bir tek modelli son teknoloji BLEU puanı oluşturuyor. Transformer'ın hem büyük hem de sınırlı eğitim verileriyle İngilizce seçim ayrıştırmasına başarıyla uygulanarak diğer görevlere iyi bir şekilde genelleştirildiğini gösteriyoruz.

*Eşit katkı. Listeleme sırası rastgeledir. Jakob, RNN'leri öz dikkat ile değiştirmeyi önerdi ve bu fikri değerlendirme çabasını başlattı. Ashish, Illia ile birlikte ilk Transformer modellerini tasarladı ve uyguladı ve bu çalışmanın her yönüne önemli ölçüde dahil oldu. Noam, ölçeklendirilmiş nokta çarpımı dikkatini, çoklu kafa dikkatini ve parametresiz konum temsilini önerdi ve neredeyse her ayrıntıda yer alan diğer kişi oldu. Niki, orijinal kod tabanımızdaki ve tensor2tensor'daki sayısız model varyantını tasarladı, uyguladı, ayarladı ve değerlendirdi. Llion da yeni model varyantlarını denedi, ilk kod tabanımızdan ve etkili çıkarım ve görselleştirmelerden sorumluydu. Lukasz ve Aidan, tensor2tensor'un çeşitli bölümlerini tasarlamak ve uygulamak için sayısız uzun günler harcadılar, önceki kod tabanımızı değiştirdiler, sonuçları büyük ölçüde iyileştirdiler ve araştırmamızı büyük ölçüde hızlandırdılar.

†Google Brain'deyken gerçekleştirilen çalışma.

‡Google Research'te gerçekleştirilen çalışma.

31. Sinirsel Bilgi İşleme Sistemleri Konferansı (NIPS 2017), Long Beach, CA, ABD.

1 Giriş

Tekrarlayan sinir ağları, özellikle uzun kısa süreli bellek [13] ve geçitli tekrarlayan [7] sinir ağları, dil modelleme ve makine çevirisi gibi dizi modelleme ve dönüştürme problemlerinde son teknoloji yaklaşımlar olarak sağlam bir şekilde kurulmuştur [35, 2, 5]. O zamandan beri çok sayıda çaba, tekrarlayan dil modellerinin ve kodlayıcı-kod çözücü mimarilerinin sınırlarını zorlamaya devam etmiştir [38, 24, 15].

Tekrarlayan modeller tipik olarak giriş ve çıkış dizilerinin sembol konumları boyunca hesaplamayı hesaba katar. Pozisyonları hesaplama süresindeki adımlarla *hizalayarak*, önceki gizli durum $h_{(t-1)}$ ve t pozisyonu için girdinin bir fonksiyonu olarak bir dizi gizli durum h_t üretirler. Bu doğal olarak sıralı yapı, eğitim örnekleri içinde paralelleştirmeyi engeller, bu da bellek kısıtlamaları örnekler arasında gruplamayı sınırladığından daha uzun dizi uzunluklarında kritik hale gelir. Son zamanlarda yapılan çalışmalar, çarpanlara ayırma hileleri [21] ve koşullu hesaplama [32] yoluyla hesaplama verimliliğinde önemli gelişmeler sağlarken, ikincisi durumunda model performansını da iyileştirmiştir. Bununla birlikte, sıralı hesaplamanın temel kısıtlaması devam etmektedir.

Dikkat mekanizmaları, girdi veya çıktı dizilerindeki mesafelerine bakılmaksızın bağımlılıkların modellenmesine izin vererek çeşitli görevlerde zorlayıcı dizi modelleme ve aktarım modellerinin ayrılmaz bir parçası haline gelmiştir [2, 19]. Ancak birkaç örnek dışında [27], bu tür dikkat mekanizmaları tekrarlayan bir ağ ile birlikte kullanılmaktadır.

Bu çalışmada, yinelemeden kaçınan ve bunun yerine girdi ve çıktı arasındaki küresel bağımlılıkları çizmek için tamamen bir dikkat mekanizmasına dayanan bir model mimarisi olan Transformer'ı öneriyoruz. Transformer önemli ölçüde daha fazla paralelleştirmeye izin verir ve sekiz P100 GPU üzerinde on iki saat gibi kısa bir süre eğitildikten sonra çeviri kalitesinde yeni bir sanat durumuna ulaşabilir.

2 Arka plan

Sıralı hesaplamayı azaltma hedefi, hepsi de temel yapı taşı olarak evrişimli sinir ağlarını kullanan ve gizli temsilleri tüm giriş ve çıkış konumları için paralel olarak hesaplayan Genişletilmiş Sinir GPU'sunun [16], ByteNet'in [18] ve ConvS2S'in [9] de temelini oluşturur. Bu modellerde, iki rastgele giriş veya çıkış konumundan gelen sinyalleri ilişkilendirmek için gereken işlem sayısı, ConvS2S için doğrusal ve ByteNet için logaritmik olarak konumlar arasındaki mesafe arttıkça artar. Bu, uzak konumlar arasındaki bağımlılıkları öğrenmeyi daha zor hale getirir [12]. Transformer'da bu, dikkat ağırlıklı konumların ortalamasının a l ı n m a s ı nedeniyle etkin çözünürlüğün azalması pahasına da olsa sabit sayıda işleme indirgenir; bu etkiyi bölüm 3.2'.de açıklandığı gibi Çok Başlı Dikkat ile önlüyoruz

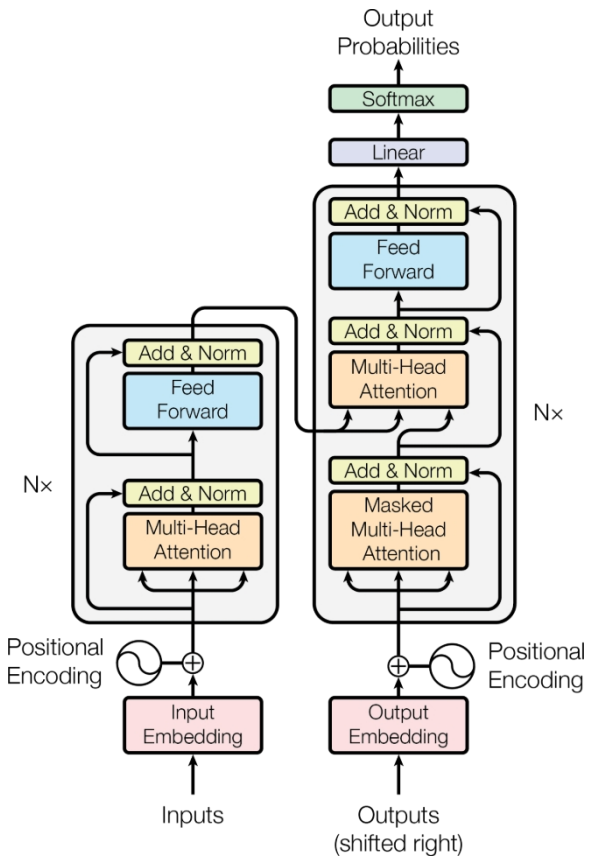
Bazen iç dikkat olarak da adlandırılan öz dikkat, dizinin bir temsilini hesaplamak için tek bir dizinin farklı konumlarını ilişkilendiren bir dikkat mekanizmasıdır. Kendi kendine dikkat, okuduğunu anlama, soyutlayıcı özetleme, metinsel gerektirme ve görevden bağımsız cümle temsillerini öğrenme gibi çeşitli görevlerde başarıyla kullanılmıştır [4, 27, 28, 22].

Uçtan uca bellek ağları, dizi hizalı yineleme yerine yinelemeli bir dikkat mekanizmasına dayanmaktadır ve basit dilli soru yanıtlama ve dil modelleme görevlerinde iyi performans gösterdiği gösterilmiştir [34].

Ancak bildiğimiz kadarıyla Transformer, sıralı RNN'ler veya konvolüsyon kullanmadan girdi ve çıktısının temsillerini hesaplamak için tamamen öz-dikkate dayanan ilk transdüksiyon modelidir. Aşağıdaki bölümlerde, Transformatör'ü tanımlayacak, öz dikkati motive edecek ve [17, 18] ve [9] .gibi modellere göre avantajlarını tartışacağız

3 Model Mimarisi

Çoğu rekabetçi nöral dizi iletim modeli bir kodlayıcı-kod çözücü yapısına sahiptir [5, 2, 35]. Burada kodlayıcı, sembol temsillerinden oluşan bir giriş dizisini $(x_1, ..., x_n)$ sürekli temsillerden oluşan bir diziye $\mathbf{z} = (z_1, ..., z_n)$ eşler. $\mathbf{Z}$ verildiğinde, kod çözücü daha sonra her seferinde bir eleman olmak üzere sembollerden oluşan bir çıkış dizisi $(y_1, ..., y_m)$ üretir. Her adımda model otomatik regresiftir [10] ve bir sonrakini üretirken daha önce üretilen sembolleri ek girdi olarak kullanır.



Şekil 1: Transformatör - model mimarisi.

Transformatör, Şekil 1'in sırasıyla sol ve sağ yarısında gösterilen hem kodlayıcı hem de kod çözücü için yığılmış öz dikkat ve noktasal, tam bağlantılı katmanlar kullanarak bu genel mimariyi takip eder.

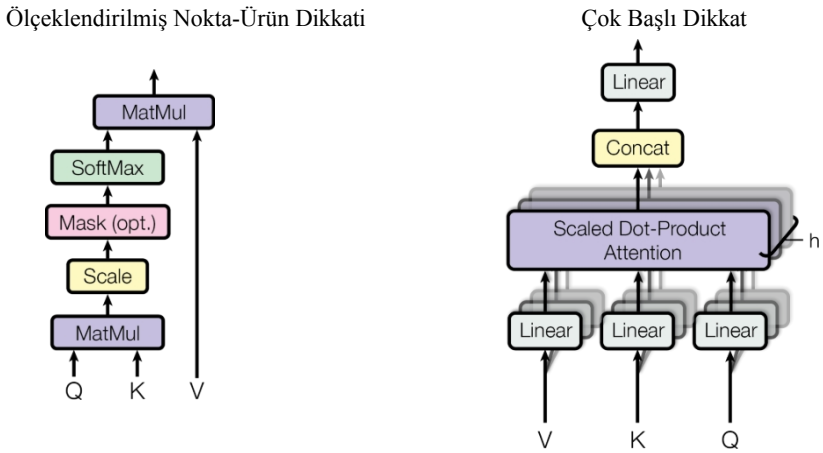
3.1 Kodlayıcı ve Kod Çözücü Yığınları

Kodlayıcı**:** Kodlayıcı**,** N= 6 özdeş katmandan oluşan bir yığından oluşur. Her katmanın iki alt katmanı vardır. Birincisi çok kafalı bir kendi kendine dikkat mekanizması, ikincisi ise basit, konum bilgisine sahip tam bağlı ileri beslemeli bir ağdır. İki alt katmanın her birinin etrafında bir artık bağlantı [11] ve ardından katman normalizasyonu [1] kullanıyoruz. Yani, her bir alt katmanın çıktısı LayerNorm(x+ Sublayer(x)), burada Sublayer(x) alt katmanın kendisi tarafından uygulanan işlevdir. Bu artık bağlantıları kolaylaştırmak için, modeldeki tüm alt katmanlar ve gömme katmanları, d_{model}= 512 boyutunda çıktılar üretir.

Kod Çözücü: Kod çözücü de N= 6 özdeş katmandan oluşan bir yığından oluşur. Her bir kodlayıcı katmanındaki iki alt katmana ek olarak, kod çözücü, kodlayıcı yığınının çıktısı üzerinde çok kafalı dikkat gerçekleştiren üçüncü bir alt katman ekler. Kodlayıcıya benzer şekilde, alt katmanların her birinin etrafında artık bağlantılar ve ardından katman normalizasyonu kullanıyoruz. Ayrıca, pozisyonların sonraki pozisyonlara katılmasını önlemek için kod çözücü yığınındaki kendi kendine dikkat alt katmanını da değiştiriyoruz. Bu maskeleme, çıktı katıştırmalarının bir konum kaydırılmış olması gerçeğiyle birleştiğinde, i konumu için tahminlerin yalnızca i'den daha küçük konumlardaki bilinen çıktılara bağlı olmasını sağlar.

3.2 Dikkat

Bir dikkat fonksiyonu, bir sorguyu ve bir dizi anahtar-değer çiftini bir çıktıya eşlemek olarak tanımlanabilir; burada sorgu, anahtarlar, değerler ve çıktının tümü vektördür. Çıktı, ağırlıklı bir toplam olarak hesaplanır



Şekil 2: (solda) Ölçeklendirilmiş Nokta Çarpımı Dikkati. (sağda) Çok Kafalı Dikkat, paralel olarak çalışan birkaç dikkat katmanından oluşur.

Her bir değere atanan ağırlığın, sorgunun ilgili anahtarla uyumluluk fonksiyonu tarafından hesaplandığı değerlerin

3.2.1 Ölçeklendirilmiş Nokta-Ürün Dikkati

Özel dikkatimizi "Ölçeklendirilmiş Nokta Çarpımı Dikkati" olarak adlandırıyoruz (Şekil 2). Girdi şunlardan oluşur
$d_k$boyutunda sorgular ve anahtarlar, a $\sqrt{n\ d}$ $d_v$boyutunda değerler. Nokta çarpımlarını hesaplıyoruz
tüm anahtarlarla sorgu yapın, her d_k, ve softmax fonksiyonu uygulayarak ağırlıkları elde edin.
birini değerlere bölün.

Pratikte, dikkat fonksiyonunu aynı anda bir dizi sorgu üzerinde hesaplarız, bunlar Q matrisinde bir araya getirilir. Anahtarlar ve değerler de K ve V matrislerinde bir araya getirilir. Çıktıların matrisini şu şekilde hesaplarız:

$$\text{Dikkat}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

En yaygın olarak kullanılan iki dikkat fonksiyonu, eklemeli dikkat [2] ve nokta çarpımlı (çok çarpımlı) dikkattir. Nokta-çarpım dikkati, $\frac{1}{\sqrt{d_k}}$ölçekleme faktörü dışında bizim algoritmamızla aynıdır. Eklemeli dikkat, aşağıdaki özelliklere sahip ileri beslemeli bir ağ kullanarak uyumluluk işlevini hesaplar
tek bir gizli katman. Her ikisi de teorik karmaşıklık açısından benzer olsa da, nokta çarpım dikkati pratikte çok daha hızlı ve daha az yer kaplar, çünkü yüksek düzeyde optimize edilmiş matris çarpma kodu kullanılarak uygulanabilir.

Küçük $d_k$değerleri için iki mekanizma benzer performans gösterirken, daha büyük $d_k$değerleri için ölçekleme olmaksızın eklemeli dikkat nokta çarpımı dikkatinden daha iyi performans gösterir [3]. Büyük $d_k$değerleri için nokta çarpımların büyüklüklerinin arttığından ve softmax fonksiyonunu son derece küçük gradyanlara sahip olduğu bölgelere ittiğinden şüpheleniyoruz [4]. Bu etkiyi ortadan kaldırmak için nokta çarpımlarını $\frac{1}{\sqrt{d_k}}$ ile ölçeklendiriyoruz.

3.2.2 Çok Başlı Dikkat

d_{model}boyutlu anahtarlar, değerler ve sorgularla tek bir dikkat fonksiyonu gerçekleştirmek yerine, sorguları, anahtarları ve değerleri h kez farklı, öğrenilmiş doğrusal projeksiyonlarla sırasıyla $d_{(k)}$, $d_k$ve $d_v$boyutlarına doğrusal olarak yansıtmanın faydalı olduğunu gördük. Sorguların, anahtarların ve değerlerin bu yansıtılmış versiyonlarının her birinde dikkat fonksiyonunu paralel olarak gerçekleştiririz ve d_v-boyutlu

[4] Nokta çarpımlarının neden büyük olduğunu göstermek için q ve k bileşenlerinin bağımsız rastgele olduğunu varsayalım ortalaması 0 ve varyansı 1 olan değişkenler. Daha sonra bunların nokta çarpımı, $q \cdot k = \sum_{i=1}^{d_k} q_i k_i$, 0 ortalama ve d_k varyansa sahiptir.

çıktı değerleri. Bunlar birleştirilir ve bir kez daha yansıtılır, böylece Şekil 2'de gösterildiği gibi nihai değerler elde edilir.

Çok kafalı dikkat, modelin farklı konumlardaki farklı temsil alt uzaylarından gelen bilgilere birlikte katılmasını sağlar. Tek bir dikkat kafası ile ortalama alma bunu engeller.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\, W^O$$
$$\text{burada head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Projeksiyonların parametre matrisleri olduğu durumlarda $W_i^Q \in \mathbb{R}^{(d_{(\text{model})} \times (d_{(k)})}$, $W_i^K \in \mathbb{R}^{(d_{(\text{model})} \times (d_{(k)})}$, $W_i^V \in \mathbb{R}^{(d_{(\text{model})} \times (d_{(v)})}$ ve $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

Bu çalışmada $h = 8$ paralel dikkat katmanı veya kafa kullanıyoruz. Bunların her biri için $d_k = d_v = d_{\text{model}}/h = 64$ kullanıyoruz. Her bir kafanın boyutunun küçültülmesi nedeniyle, toplam hesaplama maliyeti, tam boyutsallığa sahip tek kafalı dikkatinkine benzerdir.

3.2.3 Dikkatin Modelimizdeki Uygulamaları

Transformatör çok kafalı dikkati üç farklı şekilde kullanır:

- "Kodlayıcı-kod çözücü dikkat" katmanlarında, sorgular bir önceki kod çözücü katmanından gelir ve bellek anahtarları ve değerleri kodlayıcının çıkışından gelir. Bu, kod çözücüdeki her pozisyonun giriş dizisindeki tüm pozisyonlara katılmasını sağlar. Bu, [38, 2, 9] .gibi diziden diziye modellerde tipik kodlayıcı-kod çözücü dikkat mekanizmalarını taklit eder
- Kodlayıcı kendi kendine dikkat katmanları içerir. Öz dikkat katmanında tüm anahtarlar, değerler ve sorgular aynı yerden, bu durumda kodlayıcıdaki bir önceki katmanın çıktısından gelir. Kodlayıcıdaki her pozisyon, kodlayıcının bir önceki katmanındaki tüm pozisyonlara katılabilir.
- Benzer şekilde, kod çözücüdeki öz dikkat katmanları, kod çözücüdeki her bir konumun, kod çözücüdeki o konuma kadar ve o konum dahil tüm konumlara katılmasına izin verir. Otomatik regresif özelliği korumak için kod çözücüde sola doğru bilgi akışını önlememiz gerekir. Bunu, softmax girişindeki yasadışı bağlantılara karşılık gelen tüm değerleri maskeleyerek (olarak ayarlayarak) ölçeklendirilmiş nokta çarpımı dikkatinin içinde uyguluyoruz. Şekil 2'ye .bakınız

3.3 Konum Bazlı İleri Beslemeli Ağlar

Dikkat alt katmanlarına ek olarak, kodlayıcı ve kod çözücümüzdeki katmanların her biri, her konuma ayrı ayrı ve aynı şekilde uygulanan tam bağlantılı bir ileri besleme ağı içerir. Bu, aralarında bir ReLU aktivasyonu bulunan iki doğrusal dönüşümden oluşur.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_{(2)} \qquad (2)$$

Doğrusal dönüşümler farklı konumlarda aynı olsa da, katmandan katmana farklı parametreler kullanırlar. Bunu tanımlamanın bir başka yolu da çekirdek boyutu 1 olan iki konvolüsyondur. Giriş ve çıkışın boyutluluğu $d_{\text{model}} = 512$'dir ve iç katman $d_{ff} = 2048$ boyutluluğuna sahiptir.

3.4 Gömmeler ve Softmax

Diğer sekans dönüştürme modellerine benzer şekilde, giriş jetonlarını ve çıkış jetonlarını d_{model} boyutundaki vektörlere dönüştürmek için öğrenilmiş katıştırmaları kullanırız. Ayrıca, kod çözücü çıktısını tahmin edilen sonraki sözcük olasılıklarına dönüştürmek için olağan öğrenilmiş doğrusal dönüştürme ve softmax işlevini kullanırız. İçinde
modelimizde, iki gömme katmanı ve ön-softmax arasında aynı ağırlık matrisini paylaşıyoruz, böylece doğrusal dönüşüm, [30]'a benzer şekilde. Gömme katmanlarında, bu ağırlıkları şu değerlerle çarpıyoruz $\sqrt{d_{\text{model}}}$.

Tablo 1: Farklı katman türleri için maksimum yol uzunlukları, katman başına karmaşıklık ve minimum sıralı işlem sayısı. n sıra uzunluğu, d temsil boyutu, k konvolüsyonların çekirdek boyutu ve r kısıtlı öz dikkatte komşuluğun boyutudur.

Katman Tipi	Katman Başına Karmaşıklık	Sıralı Operasyonlar	Maksimum Yol Uzunluğu
Öz Dikkat	$O(n^{(2)} \cdot d)$	$O(1)$	$O(1)$
Tekrarlayan	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Konvolüsyonel	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_{(k)}(n))$
Öz Dikkat (kısıtlı)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

3.5 Konumsal Kodlama

Modelimiz yineleme ve evrişim içermediğinden, modelin dizinin sırasını kullanabilmesi için, dizideki belirteçlerin göreli veya mutlak konumu hakkında bazı bilgiler enjekte etmeliyiz. Bu amaçla, kodlayıcı ve kod çözücü yığınlarının alt kısımlarındaki girdi katıştırmalarına "konumsal kodlamalar" ekleriz. Konumsal kodlamalar, katıştırmalarla aynı d_{model}boyutuna sahiptir, böylece ikisi toplanabilir. Öğrenilmiş ve sabit olmak üzere birçok konumsal kodlama seçeneği vardır [9].

Bu çalışmada, farklı frekanslarda sinüs ve kosinüs fonksiyonları kullanılmaktadır:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Burada pos pozisyon ve i boyuttur. Yani, konumsal kodlamanın her boyutu bir sinüzoide karşılık gelir. Dalga boyları 2π'den 10000 2π'ye kadar geometrik bir ilerleme oluşturur. Bu fonksiyonu seçtik çünkü herhangi bir sabit k ofseti için PE_{pos+k} $PE_{(pos)}$'un doğrusal bir fonksiyonu olarak temsil edilebileceğinden, modelin göreceli konumlara göre katılmayı kolayca öğrenmesine izin vereceğini varsaydık.

Bunun yerine öğrenilmiş konumsal katıştırmaları [9] kullanmayı da denedik ve iki versiyonun neredeyse aynı sonuçları verdiğini gördük (bkz. Tablo 3 satır (E)). Sinüzoidal versiyonu seçtik çünkü modelin eğitim sırasında karşılaşılanlardan daha uzun dizi uzunluklarına ekstrapolasyon yapmasına izin verebilir.

4 Neden Öz Dikkat

Bu bölümde, öz-dikkat katmanlarının çeşitli yönlerini, değişken uzunluktaki bir sembol gösterimi dizisini $(x_1, ..., x_n)$ eşit uzunluktaki başka bir diziye $(z_1, ..., z_n)$ eşlemek için yaygın olarak kullanılan tekrarlayan ve konvolüsyonel katmanlarla karşılaştırıyoruz. $x_i, z_i \in \mathbb{R}^d$, tipik bir dizi iletim kodlayıcı veya kod çözücüsündeki gizli katman gibi. Öz-dikkat kullanımımızı motive etmek için üç arzuyu göz önünde bulunduruyoruz.

Bunlardan biri katman başına toplam hesaplama karmaşıklığıdır. Diğeri, gerekli minimum sıralı işlem sayısı ile ölçüldüğü üzere paralelleştirilebilen hesaplama miktarıdır.

Üçüncüsü ise ağdaki uzun menzilli bağımlılıklar arasındaki yol uzunluğudur. Uzun menzilli bağımlılıkları öğrenmek, birçok dizi dönüştürme görevinde önemli bir zorluktur. Bu tür bağımlılıkları öğrenme yeteneğini etkileyen önemli bir faktör, ileri ve geri sinyallerin ağda kat etmesi gereken yolların uzunluğudur. Giriş ve çıkış dizilerindeki pozisyonların herhangi bir kombinasyonu arasındaki bu yollar ne kadar kısa olursa, uzun menzilli bağımlılıkları öğrenmek o kadar kolay olur [12]. Bu nedenle, farklı katman türlerinden oluşan ağlarda herhangi iki giriş ve çıkış konumu arasındaki maksimum yol uzunluğunu da karşılaştırıyoruz.

Tablo 1'de belirtildiği gibi, bir öz dikkat katmanı tüm konumları sabit sayıda sıralı olarak yürütülen işlemlerle birbirine bağlarken, tekrarlayan bir katman $O(n)$ sıralı işlem gerektirir. Hesaplama karmaşıklığı açısından, öz-dikkat katmanları, aşağıdaki durumlarda tekrarlayan katmanlardan daha hızlıdır

n uzunluğu d temsil boyutluluğundan daha küçüktür, ki bu genellikle makine çevirilerinde kelime-parça gibi son teknoloji modeller tarafından kullanılan cümle temsilleri için geçerlidir.
[38] ve bayt çifti [31] gösterimleri. Çok uzun dizileri içeren görevlerde hesaplama performansını artırmak için, öz dikkat, ilgili çıktı konumunun etrafında merkezlenen giriş dizisinde yalnızca r boyutunda bir komşuluğu dikkate almakla sınırlandırılabilir. Bu, maksimum yol uzunluğunu $O(n/r)$'ye çıkaracaktır. Gelecekteki çalışmalarda bu yaklaşımı daha fazla araştırmayı planlıyoruz.

Çekirdek genişliği $k < n$ olan tek bir evrişimsel katman, tüm giriş ve çıkış konumu çiftlerini birbirine bağlamaz. Bunu yapmak, bitişik çekirdekler durumunda bir $O(n/k)$ evrişimsel katman yığını veya genişletilmiş evrişimler durumunda $O(log_{(k)}(n))$ gerektirir [18], bu da ağdaki herhangi iki konum arasındaki en uzun yolların uzunluğunu artırır. Evrişimli katmanlar genellikle tekrarlayan katmanlardan k kat daha pahalıdır. Ancak ayrılabilir evrişimler [6] karmaşıklığı önemli ölçüde azaltarak $O(k\ n\ d + n\ d^{2})$'ye düşürür. Bununla birlikte, $k = n$ olsa bile, ayrılabilir bir konvolüsyonun karmaşıklığı, modelimizde benimsediğimiz yaklaşım olan öz dikkat katmanı ve noktasal ileri besleme katmanının kombinasyonuna eşittir.

Yan fayda olarak, öz dikkat daha yorumlanabilir modeller ortaya çıkarabilir. Modellerimizden dikkat dağılımlarını inceliyor ve ekte örnekler sunup tartışıyoruz. Bireysel dikkat kafaları sadece farklı görevleri yerine getirmeyi açıkça öğrenmekle kalmıyor, birçoğu cümlelerin sözdizimsel ve anlamsal yapısıyla ilgili davranışlar sergiliyor gibi görünüyor.

5 Eğitim

Bu bölümde modellerimiz için eğitim rejimi açıklanmaktadır.

5.1 Eğitim Verileri ve Gruplama

Yaklaşık 4,5 milyon cümle çiftinden oluşan standart WMT 2014 İngilizce-Almanca veri kümesi üzerinde eğitildik. Cümleler, yaklaşık 37000 jetonluk ortak bir kaynak-hedef kelime dağarcığına sahip olan bayt-çifti kodlaması [3] kullanılarak kodlandı. İngilizce-Fransızca için, 36 milyon cümleden oluşan ve belirteçleri 32000 kelime parçalı bir kelime hazinesine bölen çok daha büyük WMT 2014 İngilizce-Fransızca veri kümesini kullandık [38]. Cümle çiftleri yaklaşık dizi uzunluğuna göre bir araya getirilmiştir. Her eğitim grubu, yaklaşık 25000 kaynak belirteç ve 25000 hedef belirteç içeren bir dizi cümle çifti içeriyordu.

5.2 Donanım ve Program

Modellerimizi 8 NVIDIA P100 GPU'lu bir makinede eğittik. Makale boyunca açıklanan hiperparametreleri kullanan temel modellerimiz için her eğitim adımı yaklaşık 0,4 saniye sürdü. Temel modelleri toplam 100.000 adım veya 12 saat boyunca eğittik. Büyük modellerimiz için (tablo 3), 'ün alt satırında açıklanmıştıradım süresi 1.0 saniyeydi. Büyük modeller 300.000 adım (3,5 gün) boyunca eğitilmiştir.

5.3 Optimize Edici

Adam optimizerini [20] $\beta_1 = 0.9$, $\beta_2 = 0.98$ ve $\epsilon = 10^{-(9)}$ ile kullandık. Eğitim süresince öğrenme oranını aşağıdaki formüle göre değiştirdik:

$$lrate = d_{\text{model}}^{-0(.)(5)} \cdot \min(step\_num^{-0.(5)}, step\_num \cdot warmup\_steps^{-1.5}) \quad (3)$$

Bu, ilk $warmup\_steps$ eğitim adımları için öğrenme oranını doğrusal olarak artırmaya ve daha sonra adım sayısının ters karekökü ile orantılı olarak azaltmaya karşılık gelir. $Warmup\_steps = 4000$ kullandık.

5.4 Düzenli hale getirme

Eğitim sırasında üç tür düzenlileştirme kullanıyoruz:

Tablo 2: Transformer, İngilizce'den Almanca'ya ve İngilizce'den Fransızca'ya newstest2014 testlerinde eğitim maliyetinin çok altında bir maliyetle önceki son teknoloji modellerden daha iyi BLEU puanları elde etmektedir.

Model	BLEU		Eğitim Maliyeti (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att+ PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT+ RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
Çevre ve Şehircilik Bakanlığı [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att+ PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT+ RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformatör (temel model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformatör (büyük)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Artık Bırakma Alt katman girişine eklenmeden ve normalleştirilmeden önce her alt katmanın çıkışına bırakma [33] uyguluyoruz. Buna ek olarak, hem kodlayıcı hem de kod çözücü yığınlarındaki gömme ve konumsal kodlamaların toplamlarına dropout uyguluyoruz. Temel model için $P_{drop} = 0.1$ oranını kullanıyoruz.

Etiket Düzgünleştirme Eğitim sırasında, $\epsilon_{(ls)} = 0.1$ değerinde etiket düzgünleştirme kullandık [36]. Bu, model daha emin olmamayı öğrendiği için karmaşıklığa zarar verir, ancak doğruluğu ve BLEU puanını artırır.

6 Sonuçlar

6.1 Makine Çevirisi

WMT 2014 İngilizceden Almancaya çeviri görevinde, büyük dönüştürücü modeli (Tablo 2'de Transformer (big)) daha önce bildirilen en iyi modellerden (topluluklar dahil) 2,0 BLEU'dan fazla daha iyi performans göstererek 28,4'lük yeni bir son teknoloji BLEU puanı oluşturmuştur. Bu modelin yapılandırması Tablo 3'ün . en alt satırında listelenmiştirEğitim 8 P100 GPU üzerinde 3,5 gün sürmüştür. Temel modelimiz bile, rakip modellerden herhangi birinin eğitim maliyetinin çok altında bir maliyetle, daha önce yayınlanmış tüm modelleri ve toplulukları geride bırakmaktadır.

WMT 2014 İngilizce'den Fransızca'ya çeviri görevinde, büyük modelimiz 41,0 BLEU puanı elde ederek daha önce yayınlanmış tüm tek modellerden daha iyi performans göstermiş ve önceki en gelişmiş modelin eğitim maliyetinin 1/4'ünden daha azını karşılamıştır. İngilizce-Fransızca için eğitilen Transformer (büyük) modeli, 0,3 yerine $P_{(drop)} = 0.1$ bırakma oranı kullanmıştır.

Temel modeller için, 10 dakikalık aralıklarla yazılan son 5 kontrol noktasının ortalaması alınarak elde edilen tek bir model kullandık. Büyük modeller için son 20 kontrol noktasının ortalamasını aldık. Işın boyutu 4 ve uzunluk cezası $\alpha = 0.6$ olan ışın arama yöntemini kullandık [38]. Bu hiperparametreler, geliştirme seti üzerinde yapılan deneylerden sonra seçilmiştir. Çıkarım sırasında maksimum çıktı uzunluğunu girdi uzunluğu + 50 olarak ayarladık, ancak mümkün olduğunda erken sonlandırdık [38].

Tablo 2 sonuçlarımızı özetler ve çeviri kalitemizi ve eğitim maliyetlerimizi literatürdeki diğer model mimarileriyle karşılaştırır. Bir modeli eğitmek için kullanılan kayan nokta işlemlerinin sayısını, eğitim süresini, kullanılan GPU sayısını ve her bir GPU'nun sürekli tek hassasiyetli kayan nokta kapasitesinin bir tahminini çarparak tahmin ediyoruz [5].

6.2 Model Varyasyonları

Dönüştürücünün farklı bileşenlerinin önemini değerlendirmek için, temel modelimizi farklı şekillerde değiştirdik ve İngilizce'den Almanca'ya çeviri performansındaki değişimi

[5] K80, K40, M40 ve P100 için sırasıyla 2,8, 3,7, 6,0 ve 9,5 TFLOPS değerlerini kullandık.

Tablo 3: Transformer mimarisi üzerindeki varyasyonlar. Listelenmemiş değerler temel modelinkilerle aynıdır. Tüm ölçümler İngilizce'den Almanca'ya çeviri geliştirme seti newstest2013 üzerinedir. Listelenen çapraşıklıklar, bayt çifti kodlamamıza göre kelime parçası başına olup, kelime başına çapraşıklıklarla karşılaştırılmamalıdır.

	N	dmodel	dff	h	d_k	d_v	$Pdrop$	ϵls	tren adıml	PPL (dev)	BLEU (dev)	params $\times 10^6$
taban	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)		sinüzoidler yerine konumsal gömme								4.92	25.7	
büyük	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

geliştirme seti, newstest2013. Önceki bölümde açıklandığı gibi ışın araması kullandık, ancak kontrol noktası ortalaması almadık. Bu sonuçları Tablo 3'te sunuyoruz.

Tablo 3 3.2.2'. satır (A)'da, Bölüm de açıklandığı gibi hesaplama miktarını sabit tutarak dikkat kafalarının sayısını ve dikkat anahtar ve değer boyutlarını değiştiriyoruzTek kafalı dikkat en iyi ayardan 0,9 BLEU daha kötü olsa da, kalite çok fazla kafa ile de düşmektedir.

Tablo 3 satır (B)'de, dikkat anahtarı boyutunun $d_k$azaltılmasının model kalitesine zarar verdiğini gözlemliyoruz. Bu, uyumluluğu belirlemenin kolay olmadığını ve nokta çarpımından daha sofistike bir uyumluluk fonksiyonunun faydalı olabileceğini göstermektedir. Ayrıca (C) ve (D) satırlarında, beklendiği gibi, daha büyük modellerin daha iyi olduğunu ve bırakmanın aşırı uyumdan kaçınmada çok yardımcı olduğunu gözlemliyoruz. (E) satırında sinüzoidal konumsal kodlamamızı öğrenilmiş konumsal yerleştirmelerle [9] değiştiriyoruz ve temel modelle neredeyse aynı sonuçları gözlemliyoruz.

6.3 İngilizce Seçim Bölgesi Ayrıştırma

Dönüştürücünün diğer görevlere genelleştirilip genelleştirilemeyeceğini değerlendirmek için İngilizce bileşen ayrıştırma üzerinde deneyler yaptık. Bu görev özel zorluklar içermektedir: çıktı güçlü yapısal kısıtlamalara tabidir ve girdiden önemli ölçüde daha uzundur. Ayrıca, RNN diziden diziye modelleri küçük veri rejimlerinde son teknoloji sonuçlara ulaşamamıştır [37].

Penn Treebank'in [25] Wall Street Journal (WSJ) bölümünde, yaklaşık 40 bin eğitim cümlesi üzerinde d_{model}= 1024 ile 4 katmanlı bir dönüştürücü eğittik. Ayrıca, yaklaşık 17 milyon cümle içeren daha büyük yüksek güvenirlikli ve BerkleyParser derlemlerini kullanarak yarı denetimli bir ortamda eğittik [37]. Yalnızca WSJ ayarı için 16K belirteçten oluşan bir kelime haznesi ve yarı denetimli ayar için 32K belirteçten oluşan bir kelime haznesi kullandık.

Section 22 geliştirme setinde bırakma, hem dikkat hem de artık (bölüm 5.4), öğrenme oranları ve ışın boyutunu seçmek için yalnızca az sayıda deney gerçekleştirdik, diğer tüm parametreler İngilizce'den Almanca'ya temel çeviri modelinden değişmeden kaldı. Çıkarım sırasında, biz

Tablo 4: Transformer, İngilizce seçim bölgesi ayrıştırmasına iyi genelleme yapar (Sonuçlar WSJ'nin 23. Bölümündedir)

Ayrıştırıcı	Eğitim	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	Sadece WSJ, ayrımcı	88.3
Petrov ve diğerleri (2006) [29]	Sadece WSJ, ayrımcı	90.4
Zhu ve diğerleri (2013) [40]	Sadece WSJ, ayrımcı	90.4
Dyer ve diğerleri (2016) [8]	Sadece WSJ, ayrımcı	91.7
Transformatör (4 katman)	Sadece WSJ, ayrımcı	91.3
Zhu ve diğerleri (2013) [40]	yarı denetimli	91.3
Huang & Harper (2009) [14]	yarı denetimli	91.3
McClosky ve diğerleri (2006) [26]	yarı denetimli	92.1
Vinyals & Kaiser el al. (2014) [37]	yarı denetimli	92.1
Transformatör (4 katman)	yarı denetimli	92.7
Luong ve diğerleri (2015) [23]	çoklu görev	93.0
Dyer ve diğerleri (2016) [8]	üretken	93.3

maksimum çıkış uzunluğunu giriş uzunluğuna+ 300 yükseltti. Kiriş boyutu 21 ve α= 0,3 olarak kullanılmıştır. hem sadece WSJ hem de yarı denetimli ayar için.

Tablo 4'teki sonuçlarımız, göreve özgü ayarlama eksikliğine rağmen modelimizin şaşırtıcı derecede iyi performans gösterdiğini ve Tekrarlayan Sinir Ağı Grameri [8] haricinde daha önce bildirilen tüm modellerden daha iyi sonuçlar verdiğini göstermektedir.

RNN diziden diziye modellerinin [37] aksine, Transformer sadece 40 bin cümleden oluşan WSJ eğitim seti üzerinde eğitim yaparken bile Berkeley- Parser'dan [29] daha iyi performans göstermektedir.

7 Sonuç

Bu çalışmada, kodlayıcı-kod çözücü mimarilerinde en yaygın olarak kullanılan tekrarlayan katmanları çok başlı öz dikkat ile değiştirerek, tamamen dikkate dayalı ilk dizi dönüştürme modeli olan Transformer'ı sunduk.

Çeviri görevleri için Transformer, tekrarlayan veya evrişimsel katmanlara dayalı mimarilerden önemli ölçüde daha hızlı eğitilebilir. Hem WMT 2014 İngilizceden Almancaya hem de WMT 2014 İngilizceden Fransızcaya çeviri görevlerinde, teknolojinin yeni bir durumuna ulaştık. İlk görevde en iyi modelimiz daha önce bildirilen tüm topluluklardan bile daha iyi performans gösteriyor.

Dikkat tabanlı modellerin geleceği konusunda heyecanlıyız ve bunları diğer görevlere de uygulamayı planlıyoruz. Transformer'ı metin dışındaki girdi ve çıktı modalitelerini içeren problemlere genişletmeyi ve görüntüler, ses ve video gibi büyük girdi ve çıktıları verimli bir şekilde ele almak için yerel, kısıtlı dikkat mekanizmalarını araştırmayı planlıyoruz. Üretimi daha az sıralı hale getirmek bir diğer araştırma hedefimizdir.

Modellerimizi eğitmek ve değerlendirmek için kullandığımız kod `https://github.com/tensorflow/tensor2tensor` adresinde mevcuttur.

Teşekkür Nal Kalchbrenner ve Stephan Gouws'a verimli yorumları, düzeltmeleri ve ilhamları için minnettarız.

Referanslar

[1] Jimmy Lei Ba, Jamie Ryan Kiros, ve Geoffrey E Hinton. Katman normalizasyonu. *arXiv ön baskı arXiv:1607.06450*, 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho, ve Yoshua Bengio. Hizalamayı ve çevirmeyi birlikte öğrenerek nöral makine çevirisi. *CoRR*, abs/1409.0473, 2014.

[3] Denny Britz, Anna Goldie, Minh-Thang Luong, ve Quoc V. Le. Sinirsel makine çevirisi mimarilerinin büyük çaplı keşfi. *CoRR*, abs/1703.03906, 2017.

[4] Jianpeng Cheng, Li Dong, ve Mirella Lapata. Makine okuması için uzun kısa süreli bellek ağları. *arXiv ön baskı arXiv:1601.06733*, 2016.

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. İstatistiksel makine çevirisi için rnn kodlayıcı-kod çözücü kullanarak ifade temsillerini öğrenme. *CoRR*, abs/1406.1078, 2014.

[6] Francois Chollet. Xception: Derinlemesine ayrılabilir konvolüsyonlarla derin öğrenme. *arXiv ön baskı arXiv:1610.02357*, 2016.

[7] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, ve Yoshua Bengio. Geçitli tekrarlayan sinir ağlarının dizi modelleme üzerine ampirik değerlendirmesi. *CoRR*, abs/1412.3555, 2014.

[8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, ve Noah A. Smith. Tekrarlayan sinir ağı gramerleri. In *Proc. of NAACL*, 2016.

[9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, ve Yann N. Dauphin. Convolu- tional sequence to sequence learning. *arXiv preprint arXiv:1705.03122v2*, 2017.

[10] Alex Graves. Tekrarlayan sinir ağları ile diziler oluşturma. *arXiv ön baskı arXiv:1308.0850*, 2013.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, ve Jian Sun. Görüntü yaşı tanıma için derin artık öğrenme. *IEEE Bilgisayarla Görme ve Örüntü Tanıma Konferansı Bildirileri*, sayfa 770-778, 2016.

[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi ve Jürgen Schmidhuber. Tekrarlayan ağlarda gradyan akışı: uzun vadeli bağımlılıkları öğrenmenin zorluğu, 2001.

[13] Sepp Hochreiter ve Jürgen Schmidhuber. Uzun kısa süreli bellek. *Neural computation*, 9(8):1735-1780, 1997.

[14] Zhongqiang Huang ve Mary Harper. Diller arasında gizli ek açıklamalarla kendi kendini eğiten PCFG gramerleri. *Doğal Dil İşlemede Ampirik Yöntemler 2009 Konferansı Bildirileri,* sayfa 832-841. ACL, Ağustos 2009.

[15] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, ve Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

[16] Łukasz Kaiser ve Samy Bengio. Aktif bellek dikkatin yerini alabilir mi? *Nöral Bilgi İşleme Sistemlerindeki Gelişmeler içinde, (NIPS),* 2016.

[17] Łukasz Kaiser ve Ilya Sutskever. Nöral GPU'lar algoritmaları öğrenir. *Uluslararası Öğrenme Temsilleri Konferansı (ICLR)*, 2016.

[18] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves ve Ko- ray Kavukcuoglu. Neural machine translation in linear time. *arXiv ön baskı arXiv:1610.10099v2*, 2017.

[19] Yoon Kim, Carl Denton, Luong Hoang, ve Alexander M. Rush. Yapılandırılmış dikkat ağları. *Uluslararası Temsilleri Öğrenme Konferansı*, 2017.

[20] Diederik Kingma ve Jimmy Ba. Adam: Stokastik optimizasyon için bir yöntem. *ICLR*, 2015 içinde.

[21] Oleksii Kuchaiev ve Boris Ginsburg. Factorization tricks for LSTM networks. *arXiv ön baskı arXiv:1703.10722*, 2017.

[22] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou ve Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv ön baskı arXiv:1703.03130*, 2017.

[23] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, ve Lukasz Kaiser. Çoklu görev dizisinden diziye öğrenme. *arXiv ön baskı arXiv:1511.06114*, 2015.

[24] Minh-Thang Luong, Hieu Pham, ve Christopher D Manning. Dikkat tabanlı nöral makine çevirisine etkili yaklaşımlar. *arXiv ön baskı arXiv:1508.04025*, 2015.

[25] Mitchell P Marcus, Mary Ann Marcinkiewicz, ve Beatrice Santorini. Büyük bir açıklamalı İngilizce derlemi oluşturma: Penn treebank. *Computational linguistics*, 19(2):313-330, 1993.

[26] David McClosky, Eugene Charniak, ve Mark Johnson. Ayrıştırma için etkili kendi kendine eğitim. *NAACL İnsan Dili Teknolojisi Konferansı Bildirileri, Ana Konferans*, sayfa 152-159. ACL, Haziran 2006.

[27] Ankur Parikh, Oscar Täckström, Dipanjan Das ve Jakob Uszkoreit. Ayrıştırılabilir bir dikkat modeli. *Doğal Dil İşlemede Ampirik Yöntemler içinde,* 2016.

[28] Romain Paulus, Caiming Xiong, ve Richard Socher. A deep reinforced model for abstractive summarization. *arXiv ön baskı arXiv:1705.04304*, 2017.

[29] Slav Petrov, Leon Barrett, Romain Thibaux, ve Dan Klein. Doğru, kompakt ve yorumlanabilir ağaç açıklamaları öğrenme. İçinde *21. Uluslararası Hesaplamalı Dilbilim Konferansı ve ACL 44. Yıllık Toplantısı Bildirileri,* sayfa 433-440. ACL, Temmuz 2006.

[30] Ofir Press ve Lior Wolf. Using the output embedding to improve language models. *arXiv ön baskı arXiv:1608.05859*, 2016.

[31] Rico Sennrich, Barry Haddow ve Alexandra Birch. Nadir kelimelerin alt kelime birimleriyle nöral makine çevirisi. *arXiv ön baskı arXiv:1508.07909*, 2015.

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, ve Jeff Dean. Aşırı derecede büyük sinir ağları: The sparsely-gated mixture-of-experts layer. *arXiv ön baskı arXiv:1701.06538*, 2017.

[33] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, ve Ruslan Salakhutdi- nov. Dropout: sinir ağlarının aşırı uyum sağlamasını önlemenin basit bir yolu. *Journal of Machine Learning Research*, 15(1):1929-1958, 2014.

[34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston ve Rob Fergus. Uçtan uca bellek ağları. C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama ve R. Garnett, editörler, *Advances in Neural Information Processing Systems 28*, sayfa 2440-2448. Curran Associates, Inc., 2015.

[35] Ilya Sutskever, Oriol Vinyals, ve Quoc VV Le. Sinir ağları ile diziden diziye öğrenme. *Advances in Neural Information Processing Systems içinde*, sayfa 3104-3112, 2014.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, ve Zbigniew Wojna. Bilgisayarla görme için başlangıç mimarisini yeniden düşünmek. *CoRR*, abs/1512.00567, 2015.

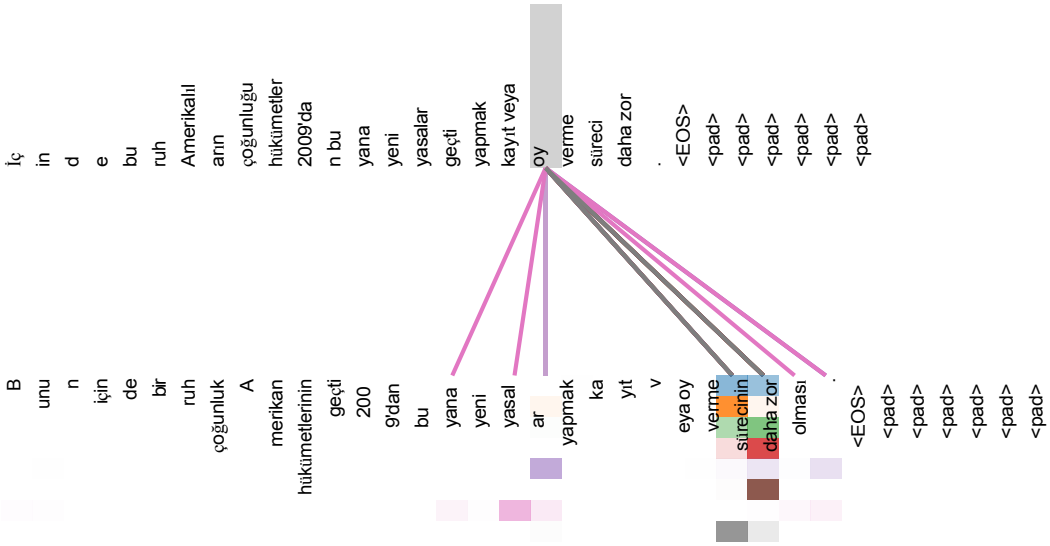
[37] Vinyals & Kaiser, Koo, Petrov, Sutskever ve Hinton. Yabancı dil olarak dilbilgisi. İçinde *Advances in Neural Information Processing Systems*, 2015.

[38] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google'ın nöral makine çeviri sistemi: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

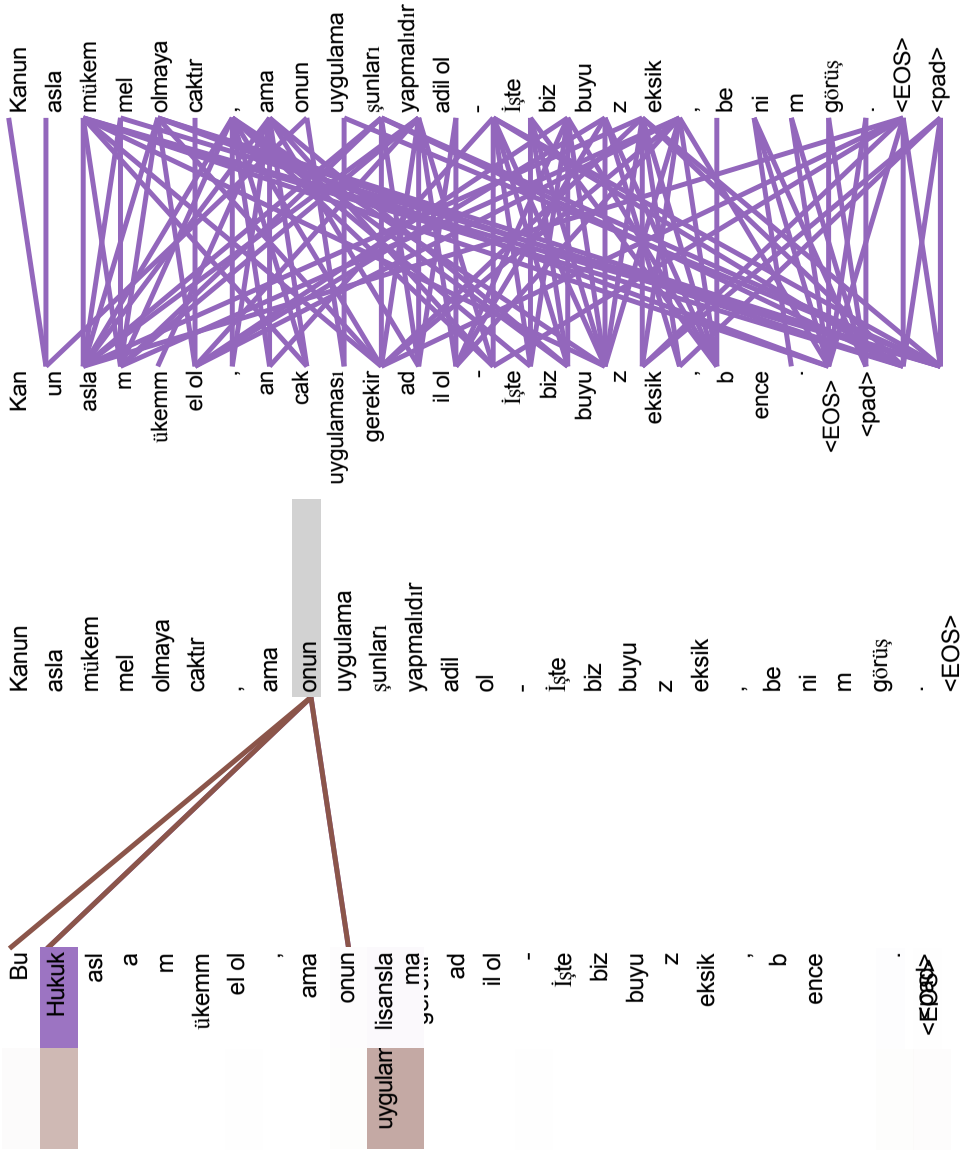
[39] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, ve Wei Xu. Nöral makine çevirisi için hızlı ileri bağlantılara sahip derin tekrarlayan modeller. *CoRR*, abs/1606.04199, 2016.

[40] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, ve Jingbo Zhu. Hızlı ve doğru shift-reduce kurucu ayrıştırma. *ACL 51. Yıllık Toplantısı Bildirileri (Cilt 1: Uzun Bildiriler)*, sayfa 434-443. ACL, Ağustos 2013.

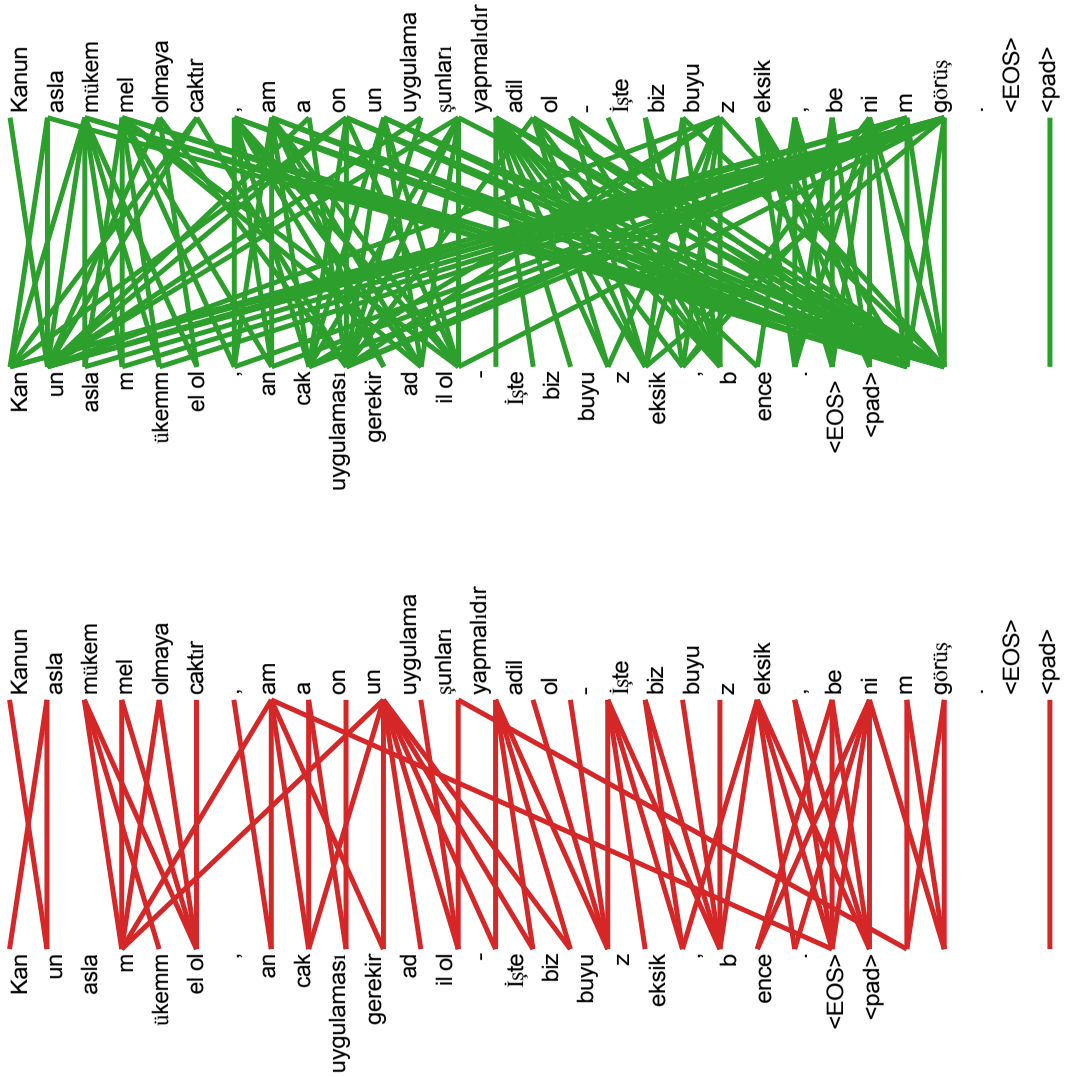
Dikkat Görselleştirmeleri



Şekil 3: 6'nın 5. katmanındaki kodlayıcı öz dikkatinde uzun mesafeli bağımlılıkları takip eden dikkat mekanizmasına bir örnek. Dikkat kafalarının çoğu 'yapmak' fiilinin uzak bir bağımlılığına katılarak 'yapmak...daha zor' ifadesini tamamlar. Buradaki dikkatler sadece 'yapmak' kelimesi için gösterilmiştir. Farklı renkler farklı başlıkları temsil etmektedir. En iyi renkli olarak görülebilir.



Şekil 4: Görünüşe göre anafora çözümlemesine dahil olan 6'nın 5. katmanında da bulunan iki dikkat kafası. Üst: 5. kafa için tam dikkat. Altta: Dikkat kafaları 5 ve 6 için sadece 'onun' kelimesinden izole edilmiş dikkatler. Bu kelime için dikkatlerin çok keskin olduğuna dikkat edin.



Şekil 5: Dikkat kafalarının birçoğu cümlenin yapısıyla ilgili görünen davranışlar sergilemektedir. Yukarıda, 6'nın 5. katmanındaki kodlayıcı öz dikkatinden iki farklı kafadan bu tür iki örnek veriyoruz. Kafalar açıkça farklı görevleri yerine getirmeyi öğrenmiştir.