Achtung ist alles, was du brauchst

Ashish Vaswani
Google Brain
avaswani@google.com

Noam Shazeer
Google Brain
noam@google.com

Niki Parmar
Google Research
nikip@google.com

Jakob Uszkoreit
Google Research
usz@google.com

Llion Jones
Google Research
Llion@google.com

Aidan N. Gomez
Universität Toronto
aidan@cs.toronto.edu

Lukasz Kaiser
Google Brain
lukaszkaiser@google.com

Illia Polosukhin
illia.polosukhin@gmail.com

Zusammenfassung

Die vorherrschenden Sequenztransduktionsmodelle basieren auf komplexen rezidivierenden oder konvolutionäre neuronale Netzwerke, die einen Encoder und einen Decoder enthalten. Ausführung Modelle verbinden auch den Encoder und Decoder durch eine Aufmerksamkeit Wir schlagen eine neue einfache Netzwerkarchitektur vor, den Transformer, ausschließlich auf der Grundlage von Aufmerksamkeitsmechanismen, Dispensing mit Rezidiv und Konvolutionen Die Experimente an zwei maschinellen Übersetzungsaufgaben zeigen, dass diese Modelle in der Qualität überlegen sein, während parallelisierbarer und erfordern erheblich weniger Zeit zum Trainieren. Unser Modell erreicht 28,4 BLEU auf der WMT 2014 Englisch-Deutsch-Übersetzungsaufgabe, Verbesserung über die bestehenden besten Ergebnisse, einschließlich ensembles, von über 2 BLEU. Auf der WMT 2014 Englisch-Französische Übersetzungsaufgabe, unser Modell stellt ein neues Modell auf dem neuesten Stand der Technik BLEU-Score von 41,8 nach Ausbildung für 3,5 Tage auf acht GPUs, ein kleiner Teil der Ausbildungskosten der beste Modelle aus der Literatur. Wir zeigen, dass der Transformer gut verallgemeinert andere Aufgaben, indem sie erfolgreich auf englischen Wahlkreis Parsing beide mit große und begrenzte Trainingsdaten.

Gleicher Beitrag. Listing-Ordnung ist zufällig. Jakob schlug vor, RNNs durch Selbstachtung zu ersetzen und begann der Versuch, diese Idee zu bewerten. Ashish, mit Illia, entworfen und implementiert die ersten Transformer-Modelle und Noam schlug vor, skalierte Punkt-Produkt-Aufmerksamkeit, Mehr-Kopf-Aufmerksamkeit und die parameterfreie Positionsdarstellung und wurde die andere Person, die in fast jedem Detail. Niki konzipiert, implementiert, abgestimmt und bewertet unzählige Modellvarianten in unserer originalen Codebase und tensor2tensor. Llion experimentierte auch mit neuartigen Modellvarianten, war verantwortlich für unsere erste Codebase, und effiziente Folgerungen und Visualisierungen. Lukasz und Aidan verbrachten unzählige lange Tage damit, verschiedene Teile von und Umsetzung von Tensor2tensor, Ersetzung unserer früheren Codebasis, deutliche Verbesserung der Ergebnisse und massive Beschleunigung Unsere Forschung.

Arbeit durchgeführt, während bei Google Brain.

Arbeit durchgeführt, während bei Google Research.

31. Konferenz über neurale Informationsverarbeitungssysteme (NIPS 2017), Long Beach, CA, USA.

1 . .Einleitung . . . . . .

Rezidivierende neuronale Netzwerke, langes Kurzzeitgedächtnis [13] und gated rezidivieren
insbesondere als Stand der Technik in der Sequenzmodellierung fest etabliert sind und
Transduktionsprobleme wie Sprachmodellierung und maschinelle Übersetzung [35, <a href
Bemühungen haben seitdem fortgesetzt, die Grenzen von wiederkehrenden Sprachmodellen und Encoder-Decoder zu verschie
Architekturen [38, 24, 15].

Recurrent-Modelle typischerweise Faktorberechnung entlang der Symbolpositionen der Ein- und Ausgabe
Sequenzen. Richten Sie die Positionen zu Schritten in Rechenzeit, erzeugen sie eine Sequenz von versteckten
Staaten , als Funktion des vorherigen versteckten Zustandes und der Eingang für Position . Dies ist inhärent
sequenzieller Charakter schließt Parallelisierung innerhalb von Ausbildungsbeispielen aus, die länger kritisch wird
Sequenzlängen, als Speicher-Beschränkungen begrenzen Batching über Beispiele. Jüngste Arbeit hat erreicht
signifikante Verbesserungen der Recheneffizienz durch Factorisierungstricks [21] und bedi
Berechnung [32], bei gleichzeitiger Verbesserung der Modellleistung bei letzterem.
Einschränkungen der sequentiellen Berechnung bleiben jedoch bestehen.

Die Aufmerksamkeitsmechanismen sind zu einem festen Bestandteil der zwingenden Sequenzmodellierung und Transduc-
modellen in verschiedenen Aufgaben, die die Modellierung von Abhängigkeiten ohne Rücksicht auf ihre Entfernung in
die Eingabe- oder Ausgabesequenzen [2, 19]. In allen bis
in Verbindung mit einem wiederkehrenden Netzwerk verwendet werden.

In dieser Arbeit schlagen wir den Transformer vor, eine Modellarchitektur, die ein Wiederaufleben verhindert und stattdessen
sich vollständig auf einen Aufmerksamkeitsmechanismus zu verlassen, um globale Abhängigkeiten zwischen Input und Outpu
Der Transformer ermöglicht eine deutlich mehr Parallelisierung und kann einen neuen Stand der Technik in erreichen
Übersetzungsqualität, nachdem sie auf acht P100 GPUs für nur zwölf Stunden trainiert wurde.

2 Hintergrund

Das Ziel, die sequenzielle Berechnung zu reduzieren, bildet auch die Grundlage der erweiterten Neural-GPU
[16], ByteNet [18] und ConvS2S <a href="attention.html
Block, Berechnung versteckter Darstellungen parallel für alle Eingangs- und Ausgangspositionen. In diesen Modellen,
die Anzahl der Operationen, die erforderlich sind, um Signale von zwei beliebigen Eingangs- oder Ausgangspositionen zu bez
im Abstand zwischen den Positionen, linear für ConvS2S und logarithmisch für ByteNet.
Es ist schwieriger, Abhängigkeiten zwischen entfernten Positionen zu lernen [12]. Im Trans
auf eine konstante Anzahl von Operationen reduziert, wenn auch auf Kosten einer reduzierten effektiven Abwicklung
um aufmerksamkeitsgewichtete Positionen zu durchschnittlich, ein Effekt, den wir mit Multi-Head Attention als
beschrieben in Abschnitt 3.2.

Selbstaufmerksamkeit, manchmal Intra-Aufmerksamkeit genannt, ist ein Aufmerksamkeitsmechanismus, der unterschiedliche
einer einzelnen Sequenz, um eine Darstellung der Sequenz zu berechnen.
erfolgreich in einer Vielzahl von Aufgaben eingesetzt werden, einschließlich Leseverständnis, abstrakte Zusammenfassung,
textual mitwirkende und lernende aufgabenunabhängige Satzdarstellungen [4, <a href="atte

End-to-End-Speichernetzwerke basieren auf einem wiederkehrenden Aufmerksamkeitsmechanismus anstelle von Sequenz-
ein abgestimmtes Rezidiv und haben gezeigt, dass gut auf einfach-sprachliche Frage beantworten und
Sprachmodellierungsaufgaben [34].

Nach bestem Wissen ist der Transformer jedoch das erste Transduktionsmodell, das sich
vollständig auf Selbstachtung, um Darstellungen seiner Ein- und Ausgabe zu berechnen, ohne Sequenz-
ausgerichtete RNNs oder Konvolution. In den folgenden Abschnitten beschreiben wir den Transformer, motivieren
Selbstachtung und diskutieren ihre Vorteile gegenüber Modellen wie [17, 18] und <a href="

3 Modellarchitektur

Die meisten wettbewerbsfähigen neuronalen Sequenztransduktionsmodelle haben eine Encoder-Decoder-Struktur <a href="att
Hier zeigt der Encoder eine Eingabefolge von Symboldarstellungen zu einer Sequenz
von kontinuierlichen Darstellungen . Gegeben , erzeugt der Decoder dann einen Ausgang
Reihenfolge von Symbolen ein Element zu einem Zeitpunkt. Bei jedem Schritt ist das Modell auto-regressiv
[10], die zuvor erzeugten Symbole als zusätzliche Eingabe bei der Erzeugung des nächsten.

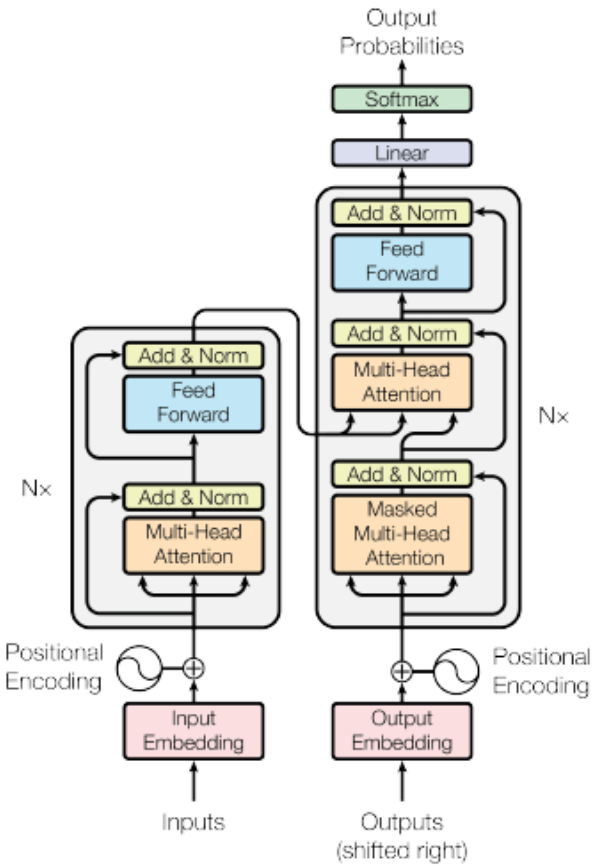


Abbildung 1: Der Transformer - Modellarchitektur.

Der Transformer folgt dieser Gesamtarchitektur mit gestapelter Selbstaufmerksamkeit und point-wise, voll verbundene Schichten für Encoder und Decoder, die in der linken und rechten Hälfte der Abbildung <a href="attention.html#3 entsprechend.

3.1. Encoder und Decoder Stapel

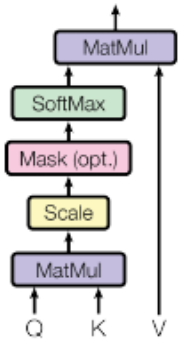
Encoder: Der Encoder besteht aus einem Stapel von identische Schichten. Jede Ebene hat zwei Unterlagen. Die erste ist ein Mehrkopf-Selbstaufmerksamkeitsmechanismus, und die zweite ist eine einfache, Position-weise voll angeschlossene Feed-Forward-Netzwerk. Wir verwenden eine Restverbindung <a href="attention.html#11">[11] <a die beiden Unterschichten, gefolgt von der Ebenennormalisierung [1]. Das heißt, die Ausgab , wobei ist die von der Unterschicht implementierte Funktion sich selbst. Um diese Restverbindungen zu erleichtern, sind alle Unterschichten im Modell, sowie die Einbettung Schichten, produzieren Ergebnisse der Dimension . . . . . .

Decoder: Der Decoder besteht auch aus einem Stapel von identische Schichten. Zusätzlich zu den beiden Unterlagen in jeder Encoder-Schicht, der Decoder fügt eine dritte Unterschicht, die Mehrkopf führt Aufmerksamkeit über die Ausgabe des Encoder-Stacks. Ähnlich wie der Encoder verwenden wir Restverbindungen um jede der Unterschichten, gefolgt von Schichtnormalisierung. Wir modifizieren auch die Selbstaufmerksamkeit Sub-Layer im Decoder-Stack, um zu verhindern, dass Positionen in Folgepositionen. Dies Maskierung, kombiniert mit der Tatsache, dass die Output-Embeddings durch eine Position kompensiert werden, sorgt dafür, Vorhersagen für Position kann nur von den bekannten Ausgängen an Positionen abhängen, die weniger als . . . . . .

3.2. Achtung

Eine Aufmerksamkeitsfunktion kann als Mapping einer Abfrage und einer Reihe von Schlüssel-Wert-Paaren zu einer Ausgabe wobei Abfrage, Schlüssel, Werte und Ausgabe alle Vektoren sind. Die Ausgabe wird als gewichtete Summe berechnet

Skalierte Dot-Produkt-Achtung



Mehrkopf-Achtung

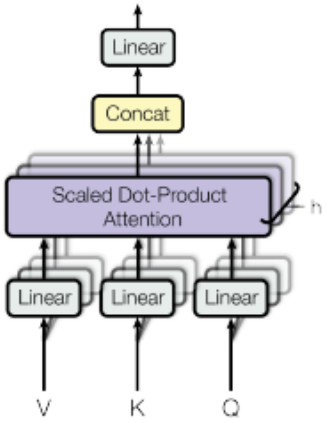


Abbildung 2: (links) Skalierte Dot-Produkt-Achtung. (rechts) Multi-Head-Achtung besteht aus mehreren Aufmerksamkeitsschichten, die parallel verlaufen.

der Werte, bei denen das jedem Wert zugewiesene Gewicht durch eine Kompatibilitätsfunktion der Abfrage mit dem entsprechenden Schlüssel.

3.2.1 Skalierte Dot-Produkt-Achtung

Wir nennen unsere besondere Aufmerksamkeit "Scaled Dot-Product Attention" (Abbildung <a href="attention.html#4">2). </
Abfragen und Schlüssel der Dimension , und Werte der Dimension . Wir berechnen die Punktprodukte der
Abfrage mit allen Tasten, teilen Sie jede durch , und wenden Sie eine Softmax-Funktion, um die Gewichte auf der
Werte.

In der Praxis berechnen wir die Aufmerksamkeitsfunktion auf eine Reihe von Abfragen gleichzeitig, zusammengepackt
in eine Matrix . Schlüssel und Werte sind auch in Matrizen verpackt und . Wir berechnen
die Matrix der Ausgänge wie folgt:

1)

Die beiden am häufigsten verwendeten Aufmerksamkeitsfunktionen sind additive Aufmerksamkeit <a href="attention.html#10
plikative) Aufmerksamkeit. Dot-Produkt Aufmerksamkeit ist identisch mit unserem Algorithmus, außer für den Skalierungsfa
von . Additive Aufmerksamkeit berechnet die Kompatibilitätsfunktion mit einem Feed-Forward-Netzwerk mit
eine einzelne verborgene Schicht. Während die beiden in theoretischer Komplexität ähnlich sind, ist die Aufmerksamkeit des
viel schneller und platzsparender in der Praxis, da es mit hochoptimierten
Matrix-Multiplikationscode.

Während für kleine Werte von Die beiden Mechanismen wirken ähnlich, additive Aufmerksamkeit übertrifft
dot Produkt Aufmerksamkeit ohne Skalierung für größere Werte von [3]. Wir vermuten, dass für gro
, die dot Produkte wachsen groß in der Größenordnung, schieben die Softmax-Funktion in Regionen, in denen es hat
extrem kleine Steigungen . Um diesem Effekt entgegenzuwirken, skalieren wir die Punktprodukte durch .

3.2.2 Mehrkopf-Achtung

Statt eine einzige Aufmerksamkeitsfunktion mit -dimensionale Schlüssel, Werte und Abfragen,
wir fanden es vorteilhaft, die Abfragen, Schlüssel und Werte linear zu projizieren mit unterschiedlichen, gelernten
lineare Projektionen auf . . . und . . . Auf jeder dieser projizierten Versionen von
Abfragen, Schlüssel und Werte führen wir dann die Aufmerksamkeitsfunktion parallel durch, -dimensional

Um zu veranschaulichen, warum die Dot-Produkte groß werden, gehen Sie davon aus, dass die Komponenten von sind unabhängige Zufallsvariablen mit Mittelwert und Varianz . Dann ihr Punkt Produkt, , hat Mittel und Varianz

Diese werden konkateniert und erneut projiziert, so dass die Endwerte wie
Abbildung 2.

Mehrkopf-Aufmerksamkeit ermöglicht es dem Modell, sich gemeinsam um Informationen aus unterschiedlichen Darstellunge
Subräume an unterschiedlichen Positionen. Mit einem einzigen Aufmerksamkeitskopf hemmt das durchschnittlich.

Dabei ist

Wenn die Projektionen Parameter Matrizen sind Die in dieser Verordnung vorgesehenen Maßnahmen entsprechen der Stellungnahme des Ständigen Ausschusses für Pflanzen, Tiere, Lebensmittel und Futtermittel
und

In dieser Arbeit beschäftigen wir parallele Aufmerksamkeitsschichten, oder Köpfe. Für jede von diesen verwenden wir
. Aufgrund der reduzierten Dimension jedes Kopfes, die gesamten Rechenkosten
ist ähnlich wie bei der Ein-Kopf-Aufmerksamkeit mit voller Dimensionalität.

3.2.3 Anwendungen der Aufmerksamkeit in unserem Modell

Der Transformer nutzt mehrköpfige Aufmerksamkeit auf drei verschiedene Arten:

- In "Encoder-Decoder-Aufmerksamkeit"-Schichten kommen die Abfragen aus der vorherigen Decoder-Schicht, und die Speichertasten und Werte kommen von der Ausgabe des Encoders. Dies ermöglicht jede Position im Decoder zu besuchen über alle Positionen in der Eingangssequenz. Dies imitiert die typische Encoder-Decoder-Aufmerksamkeitsmechanismen in Sequenz-zu-Sequenz-Modellen wie [38, 2, 9].
- Der Encoder enthält Selbstaufmerksamkeitsschichten. In einer Selbstaufmerksamkeitsschicht alle Tasten, Werte und Abfragen von der gleichen Stelle kommen, in diesem Fall die Ausgabe der vorherigen Ebene in der encoder. Jede Position im Encoder kann sich um alle Positionen in der vorherigen Ebene der Encoder.

- In ähnlicher Weise ermöglichen die Eigenabstimmungsschichten im Decoder jede Position im Decoder, sich um alle Positionen im Decoder bis einschließlich dieser Position. Wir müssen Links verhindern Informationsfluss im Decoder, um die auto-regressive Eigenschaft zu erhalten. innerhalb der skalierten Punkt-Produkt Aufmerksamkeit durch Ausblenden (Einstellen auf) alle Werte in der Eingabe der Softmax, die illegalen Verbindungen entsprechen. Siehe Abbildung 2.

3.3. Positionsweise Feed-Forward-Netzwerke

Zusätzlich zu den Aufmerksamkeitsunterlagen enthält jede der Schichten in unserem Encoder und Decoder eine vollständige connected feed-forward network, das auf jede Position separat und identisch angewendet wird.
besteht aus zwei linearen Transformationen mit einer ReLU-Aktivierung dazwischen.

2)

Während die linearen Transformationen über verschiedene Positionen hinweg gleich sind, verwenden sie unterschiedliche Para
von Layer zu Layer. Eine andere Art dies zu beschreiben ist als zwei Konvolutionen mit Kernelgröße 1.
Die Dimensionalität von Input und Output ist , und die innere Schicht hat Dimensionalität

3.4. Einbetten und Softmax

Ähnlich wie bei anderen Sequenztransduktionsmodellen verwenden wir gelernte Einbettungen, um die Eingabe zu konvertiere
Token und Ausgabetoken zu Vektoren der Dimension . Wir verwenden auch die üblichen gelernten linearen transfor-
mation und Softmax-Funktion, um den Decoder-Ausgang in vorhergesagte Next-Token-Wahrscheinlichkeiten umzuwandeln.
unser modell, teilen wir die gleiche gewichtsmatrix zwischen den beiden einbettenden schichten und der pre-softmax
lineare Transformation, ähnlich wie [30]. In den Einbettungsschichten multiplizieren wir die

Tabelle 1: Maximale Pfadlängen, pro Schicht Komplexität und minimale Anzahl von sequenziellen Operationen für verschiedene Schichttypen. ist die Sequenzlänge, ist die Darstellungsdimension, ist der Kernel Größe der Konvolutionen und die Größe der Nachbarschaft in eingeschränkter Selbstachtung.

Ebenentyp	Komplexität pro Schicht	Sequenzielle Vorhaben	Maximale Pfadlänge
Selbstaufmerksamkeit			
Wiederkehren			
Konvolutional			
Selbstaufmerksamkeit (eingeschränkt)			

3.5. Positionskodierung

Da unser Modell keine Wiederholung und keine Konvolution enthält, um das Modell von der Reihenfolge der Reihenfolge, müssen wir einige Informationen über die relative oder absolute Position der Tokens in der Sequenz. Zu diesem Zweck fügen wir den Input-Embeddings am Boden des Encoders und Decoders. Die Positionskodierungen haben die gleiche Dimension

als die Einbettungen, so dass die beiden zusammengefasst werden können. Es gibt viele Möglichkeiten der Positionskodierung gelernt und fixiert [9].

In dieser Arbeit verwenden wir Sinus- und Kosinusfunktionen unterschiedlicher Frequenzen:

Dabei ist ist die Position und ist die Dimension. Das heißt, jede Dimension der Positionskodierung entspricht einem Sinusoid. Die Wellenlängen bilden eine geometrische Progression zu aus Wir Wir haben diese Funktion gewählt, weil wir hypothesiert es würde dem Modell leicht lernen, durch relative Positionen, da für jeden festen Offset . . . . . kann als lineare Funktion von . . . . . .

Wir experimentierten auch mit der Verwendung von gelernten Positionseinbettungen [9] sta Wir haben die sinusförmige Version gewählt, die fast identische Ergebnisse liefert (siehe Tabelle <a href="attention.html#9"> weil es dem Modell erlauben kann, auf Sequenzlängen zu extrapolieren, die länger sind als die, denen es begegnet ist während der Ausbildung.

4 . . Warum Selbstachtung . . . . . .

In diesem Abschnitt vergleichen wir verschiedene Aspekte von Selbstachtungsebenen mit den wiederkehrenden und konvolutional layers, die häufig für die Abbildung einer variabel langen Sequenz von Symboldarstellungen verwendet werden

zu einer anderen Sequenz gleicher Länge , mit , wie z.B. eine versteckte Schicht in einem typischen Sequenztransduktions-Encoder oder Decoder. Motiviert unseren Einsatz von Selbst-Achtung wir Betrachten Sie drei desiderata.

Eine ist die Gesamtkomplexität der Berechnung pro Schicht. Eine andere ist die Menge der Berechnung, die parallelisiert werden, gemessen an der Mindestanzahl der erforderlichen sequenziellen Operationen.

Die dritte ist die Länge des Pfades zwischen den Längen-Abhängigkeiten im Netzwerk.
Abhängigkeiten ist eine zentrale Herausforderung in vielen Sequenztransduktionsaufgaben. Ein Schlüsselfaktor, der die Fähigkeit, solche Abhängigkeiten zu lernen ist die Länge der Wege vorwärts und rückwärts Signale müssen Traverse im Netzwerk. Je kürzer diese Pfade zwischen jeder Kombination von Positionen in der Eingabe und Ausgabesequenzen, desto einfacher ist es, Long-Range-Abhängigkeiten [12] zu lernen. die maximale Pfadlänge zwischen zwei Eingangs- und Ausgangspositionen in Netzen, die aus dem verschiedene Schichttypen.

Wie in Tabelle 1, angegeben, verbindet eine Selbstaufmerksamkeitsschicht alle Positionen m ausgeführte Operationen, während eine wiederkehrende Ebene erfordert sequenzielle Operationen. Berechnungskomplexität, Selbstaufmerksamkeit Schichten sind schneller als wiederkehrende Schichten, wenn die Reihenfolg

Länge ist kleiner als die Darstellungsdimensionalität , das ist am häufigsten der Fall mit
Satzdarstellungen, die von modernen Modellen in maschinellen Übersetzungen, wie z. B. Wort-Stück, verwendet werden
[38] und Byte-Paar [31] Repräsentationen.
sehr lange Sequenzen, Selbstachtung könnte auf nur eine Nachbarschaft von Größe beschränkt werden.
in der
die Eingabesequenz zentriert um die jeweilige Ausgangsposition. Dies würde die maximale
Pfadlänge bis . Wir planen, diesen Ansatz in Zukunft weiter zu untersuchen.

Eine einzige konvolutionäre Schicht mit Kernelbreite nicht alle Ein- und Ausgabepaare miteinander verbinden
Positionen. Dies erfordert einen Stapel von konvolutionäre Schichten bei zusammenhängenden Kernen,
oder bei ausgedehnten Konvolutionen [18], Verlängerung der Länge der längsten
zwischen zwei Positionen im Netzwerk. Konvolutionäre Schichten sind in der Regel teurer als
rezidivierende Schichten, um einen Faktor von . Separable Konvolutionen [6], aber die Komplexität verri
in erheblichem Umfang, um . Auch mit , jedoch die Komplexität einer trennbaren
Konvolution ist gleich der Kombination aus einer Selbstaufmerksamkeitsschicht und einer punktuellen Vorschubschicht,
die Herangehensweise, die wir in unserem Modell nehmen.

Als Nebennutzen könnte Selbstachtung zu interpretierbareren Modellen führen. Wir prüfen Aufmerksamkeitsverteilungen
aus unseren Modellen und präsentieren und diskutieren Beispiele im Anhang.
Köpfe deutlich lernen, verschiedene Aufgaben auszuführen, viele scheinen zu zeigen, Verhalten im Zusammenhang mit der sy
und semantische Struktur der Sätze.

5 . .Ausbildung . . . . . .

Dieser Abschnitt beschreibt das Trainingsregime für unsere Modelle.

5.1. Trainingsdaten und Batching

Wir trainierten auf dem Standard WMT 2014 Deutsch-Englisch Datensatz bestehend aus ca. 4,5 Millionen
Satzpaare. Sätze wurden mit der Byte-pair-Kodierung [3], kodiert, die eine gemeinsame Qu
Zielvokabular von ca. 37000 Token. Für Englisch-Französisch haben wir die deutlich größere WMT verwendet.
2014 Englisch-Französischer Datensatz bestehend aus 36M Sätzen und geteilten Tokens in ein 32000-Wortstück
Vokabular [38]. Sentencepaare wurden nach ungefährer Sequenzlänge zusammengestapelt.
batch enthielt einen Satz paare mit etwa 25000 Quelle Token und 25000
Zielmarken.

5.2. Hardware und Zeitplan

Wir trainierten unsere Modelle auf einer Maschine mit 8 NVIDIA P100 GPUs. Für unsere Basismodelle mit
Die im gesamten Papier beschriebenen Hyperparameter, jeder Trainingsschritt dauerte etwa 0,4 Sekunden.
die Basismodelle für insgesamt 100.000 Stufen oder 12 Stunden ausgebildet. Für unsere großen Modelle, (beschrieben auf der
untere Zeile der Tabelle 3), Schrittzeit betrug 1,0 Sekunden. Die großen Modelle wurden für
(3,5 Tage).

5.3 Optimierer

Wir verwendeten den Adam-Optimierer <a href="attention.html#11">[20] . Wir variierten das Lernen . . . . . .
die Rate während der Ausbildung, nach der Formel:

3)

Dies entspricht einer linearen Erhöhung der Lernrate für die erste Schulungsschritte,
und sie dann proportional zur inversen Quadratwurzel der Schrittzahl zu verringern.
. . . . . .

5.4 Regularisierung

Wir beschäftigen drei Arten von Regularisierung während der Ausbildung:

Tabelle 2: Der Transformer erzielt bessere BLEU-Scores als frühere Modelle auf dem neuesten Stand der Technik Englisch-Deutsch und Englisch-Französisch Newstest2014 Tests zu einem Bruchteil der Schulungskosten.

Modell	BLWU		Ausbildungskosten (FLOP)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18] 	23.75			
Deep Att + PosUnk [39]		39.2		
GNMT + RL [38] 	24.6	39.92		
ConvS2S [9] 	25.16	40.46		
MoE [32] 	26.03	40.56		
Deep Att + PosUnk Ensemble [39]		40,4		
GNMT + RL Ensemble [38] 	26.30	41.16		
ConvS2S Ensemble [9] 	26.36	**41.29**		
Transformator (Basismodell)	27.3	38.1		
Transformator (groß)	**28.4**	**41,8**		

Residual Dropout Wir wenden Dropout [33] auf die Ausgabe jeder Unterschicht an, bevo Sub-Layer-Eingabe und normalisiert. Darüber hinaus wenden wir Dropout auf die Summen der Einbettungen und die Positionale Kodierungen in den Encoder- und Decoder-Stacks. Für das Basismodell verwenden wir eine Rate von

. . . .

Glättung des Etiketts Während der Ausbildung setzten wir Labelglättung von Wert ein <a href="attention.html#12">[36]. </ schmerzt Verwirrung, da das Modell lernt, unsicherer zu sein, verbessert aber die Genauigkeit und den BLEU-Score.

6 Ergebnisse

6.1 Maschinelle Übersetzung

Auf der WMT 2014 Englisch-Deutsch-Übersetzungsaufgabe, dem großen Transformatorenmodell (Transformer (groß) In Tabelle <a href="attention.html#8">2) werden die besten bisher gemeldeten Modelle (einschließlich Ensembles) um mehr a BLWU, Einführung eines neuen aktuellen BLEU-Scores von . Die Konfiguration dieses Modells ist In der unteren Zeile von Tabelle <a href="attention.html#8">3 aufgeführt. Tage dauerte 8 P100 GPUs. Sogar unser Basismodell übertrifft alle bisher veröffentlichten Modelle und Ensembles, zu einem Bruchteil der Ausbildungskosten eines der die Wettbewerbsmodelle.

Auf der WMT 2014 Englisch-Französische Übersetzungsaufgabe erreicht unser großes Modell einen BLEU-Score von . . . . . übertrifft alle bisher veröffentlichten Einzelmodelle mit weniger als die Ausbildungskosten der Vorgängermodell auf dem neuesten Stand der Technik. Das Transformer-Modell (großes) für Englisch-Französisch ausgebilde Abbrecherquote , statt . . . .

Für die Basismodelle haben wir ein einziges Modell verwendet, das durch Mittelung der letzten 5 Kontrollpunkte, die wurden in 10-Minuten-Intervallen geschrieben. Für die großen Modelle durchschnittlich die letzten 20 Kontrollpunkte. verwendete Strahlsuche mit einer Strahlgröße von und Längenstrafe [38]. Diese Hyperpara wurden nach dem Experimentieren am Entwicklungsset ausgewählt. Wir setzen die maximale Ausgangslänge während Inferenz zur Eingangslänge +, aber vorzeitig beenden, wenn möglich [38].

Tabelle 2 Zusammenfassung unserer Ergebnisse und vergleicht unsere Übersetzungsqualität u Architekturen aus der Literatur. Wir schätzen die Anzahl der Floating-Point-Operationen verwendet, um eine Modell durch Multiplikation der Trainingszeit, der Anzahl der verwendeten GPUs und einer Schätzung der Einzelpräzisions-Schwingpunktkapazität jeder GPU *

6.2 Modellvariationen

Um die Bedeutung der verschiedenen Komponenten des Transformers zu bewerten, variierten wir unser Basismodell auf unterschiedliche Weise, Messung der Leistungsveränderung auf Englisch-Deutsch-Übersetzung auf dem

Wir verwendeten Werte von 2.8, 3.7, 6.0 und 9.5 TFLOPS für K80, K40, M40 und P100.

Tabelle 3: Variationen der Transformer-Architektur. Nicht aufgeführte Werte sind identisch mit denen der Basis Modell. Alle Metriken befinden sich auf dem Deutsch-Englisch Übersetzungs-Entwicklungs-Set, newstest2013. Verwirrtheiten sind pro-Wortstück, entsprechend unserer Byte-Pair-Kodierung, und sollten nicht mit Per-Wort-Perplexitäten.

										Zug Schritte	PPL (dev)	BLWU (dev)	Paramen
Basis	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 Uhr	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	0,0	0,0	5.77	24,6	0,0	0,0
							0,2			4.95	25.5		
								0,0	0,0	4.67	25,3	0,0	0,0
								0,2		5.47	25.7		
(E)	Positionale Einbettung anstelle von Sinusoiden									4.92	25.7		
groß	6	1024	4096	16			0,3		300K	**4.33**	**26.4**	213	

development set, newstest2013. Wir verwendeten Strahlsuche wie im vorherigen Abschnitt beschrieben, aber nein
Wir stellen diese Ergebnisse in Tabelle 3.

In Tabelle 3 rows (A) variieren wir die Anzahl der Aufmerksamkeitsköpfe und die Aufmerks
halten die Menge der Berechnung konstant, wie in Abschnitt 3.2.2. Während ein Kopf
Die Aufmerksamkeit ist 0,9 BLEU schlechter als die beste Einstellung, Qualität sinkt auch mit zu vielen Köpfen ab.

In Tabelle 3 rows (B) ist zu beobachten, dass die Reduzierung des Schlüsselgröße verringer
schlägt vor, dass die Bestimmung der Kompatibilität ist nicht einfach und dass eine anspruchsvollere Kompatibilität
Funktion als Punkt Produkt kann von Vorteil sein. Wir beobachten weiter in den Zeilen (C) und (D), dass, wie erwartet,
größere Modelle sind besser, und Dropout ist sehr hilfreich bei der Vermeidung von Überrüstung. In Reihe (E) ersetzen wir un
sinusförmige Positionskodierung mit erlernten positionalen Einbettungen [9], und beobachte
Ergebnisse zum Basismodell.

6.3 Englische Konstituenz Parsing

Um zu bewerten, ob der Transformer zu anderen Aufgaben generalisieren kann, führten wir Experimente auf Englisch durch
Diese Aufgabe stellt besondere Herausforderungen dar: Die Produktion ist stark strukturell bedingt.
schränkt und ist deutlich länger als die Eingabe. Darüber hinaus, RNN Sequenz-zu-Sequenz
Modelle konnten in kleinen Datensystemen keine aktuellen Ergebnisse erzielen [37].

Wir trainierten einen 4-Schicht-Transformator mit auf dem Wall Street Journal (WSJ) Teil des
Penn Treebank [25], ca. 40K Trainingssätze. Wir trainierten es auch in einer halbbeaufsicht
mit dem größeren Hochvertrauen und BerkleyParser corpora aus etwa 17M Sätzen
[37]. Wir verwendeten einen Vokabular von 16K Token für die WSJ nur Einstellung und ei
für die halbbeaufsichtigte Einstellung.

Wir haben nur eine kleine Anzahl von Experimenten durchgeführt, um den Dropout auszuwählen, sowohl Aufmerksamkeit als
(Abschnitt 5.4), Lernraten und Strahlgröße auf dem Abschnitt 22-Entwicklungssatz, alle ande
blieb unverändert vom deutsch-englischen Basis-Übersetzungsmodell.

Tabelle 4: Der Transformer verallgemeinert gut zum englischen Wahlkreis Parsing (Ergebnisse sind auf Abschnitt 23 von WSJ)

Parser	Ausbildung	WSJ 23 F1
Vinyals & Kaiser el al. (2014) <a href="attention.html#12">[37]	Nur WSJ, diskriminierend	88.3
Petrov et al. (2006) [29] 	Nur WSJ, diskriminierend	90,4
Zhu et al. (2013) [40] 	Nur WSJ, diskriminierend	90,4
Dyer et al. (2016) [8] 	Nur WSJ, diskriminierend	91.7
Transformator (4 Schichten)	Nur WSJ, diskriminierend	91.3
Zhu et al. (2013) [40]	halbbeaufsichtigt	91.3
Huang & Harper (2009) [14]	halbbeaufsichtigt	91.3
McClosky et al. (2006) [26] 	halbbeaufsichtigt	92.1
Vinyals & Kaiser el al. (2014) [37] 	halbbeaufsichtigt	92.1
Transformator (4 Schichten)	halbbeaufsichtigt	92.7
Luong et al. (2015) [23] 	Multi-Task	93,0
Dyer et al. (2016) [8]	generativ	93.3

erhöhte die maximale Ausgangslänge auf Eingangslänge +. Wir verwendeten eine Strahlgröße von sowohl für WSJ als auch für die halbbeaufsichtigte Einstellung.

Unsere Ergebnisse in Tabelle 4 zeigen, dass trotz fehlender aufgabenspezifischer Abstimmu In den letzten Jahren hat sich die Zahl der Beschäftigten, die in der Industrie tätig waren, in den letzten zehn Jahren erhöht, un Recurrent Neural Network Grammatik [8].

Im Gegensatz zu RNN Sequenz-zu-Sequenz-Modellen <a href="attention.html#12">[37] übertrifft der Transformer den Berke Parser [29] auch wenn man nur auf dem WSJ-Trainingssatz von 40K Sätzen trainiert.

7 . .Schlußfolgerung . . . . . .

In dieser Arbeit präsentierten wir den Transformer, das erste Sequenztransduktionsmodell, das ausschließlich auf Aufmerksamkeit, ersetzt die rezidivierenden Schichten, die am häufigsten in Encoder-Decoder-Architekturen verwendet werd Mehrköpfige Selbstachtung.

Für Übersetzungsaufgaben kann der Transformer deutlich schneller trainiert werden als Architekturen basierend auf rezidivierenden oder konvolutionären Schichten. Auf beiden WMT 2014 Englisch-Deutsch und WMT 2014 Englisch-Französische Übersetzungsaufgaben, erreichen wir einen neuen Stand der Technik. In der früheren Aufgabe unser B Modell übertrifft selbst alle bisher gemeldeten Ensembles.

Wir freuen uns über die Zukunft aufmerksamkeitsbasierter Modelle und planen, sie auf andere Aufgaben anzuwenden. Plan zur Erweiterung des Transformers auf Probleme mit anderen Input- und Output-Modalitäten als Text und Untersuchung lokaler, eingeschränkter Aufmerksamkeitsmechanismen für den effizienten Umgang mit großen Inputs und Out wie Bilder, Audio und Video. Die Erzeugung weniger sequentiell zu machen, ist ein weiteres Forschungsziel von uns.

Der Code, den wir verwendet haben, um unsere Modelle zu trainieren und zu bewerten, ist verfügbar unter == Weblinks == == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==* http://www.github.com/te

Danksagung Wir danken Nal Kalchbrenner und Stephan Gouws für ihre fruchtbare Kommentare, Korrekturen und Inspirationen.

Literaturverzeichnis

[1] Jimmy Lei Ba, Jamie Ryan Kiros und Geoffrey E Hinton. Schichtnormalisierung. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[2] Dzmitry Bahdanau, Kyunghyun Cho und Yoshua Bengio. Neural maschinelle Übersetzung von gemeinsam Lernen, sich auszurichten und zu übersetzen. abs/1409.0473, 2014.

[3] Denny Britz, Anna Goldie, Minh-Thang Luong und Quoc V. Le. Massive Erforschung neuraler maschinelle Übersetzungsarchitekturen. , abs/1703.03906, 2017.

[4] Jianpeng Cheng, Li Dong und Mirella Lapata. Lange Kurzzeitspeicher-Netzwerke für die Maschine Lesen. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, und Yoshua Bengio. Lernen Phrasen Darstellungen mit rnn Encoder-Decoder für statistische maschinelle Übersetzung. , abs/1406.1078, 2014.

[6] Francois Chollet. Xception: Tiefes Lernen mit tief trennbaren Konvolutionen. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[7] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho und Yoshua Bengio. Empirische Auswertung von gated rezidivierenden neuronalen Netzwerken auf Sequenzmodellierung. abs/1412.3555, 2014.

[8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros und Noah A. Smith. Netzwerk-Grammatiken. In , 2016.

[9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats und Yann N. Dauphin. In den meisten Fällen ist dies jedoch nicht der Fall. == Einzelnachweise ==

[10] Alex Graves. Erzeugen von Sequenzen mit wiederkehrenden neuronalen Netzwerken. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren und Jian Sun. Tiefes Restlernen für Im- Alterszugehörigkeit. , Seiten 770–778, 2016.

[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi und Jürgen Schmidhuber. wiederkehrende Netze: die Schwierigkeit, langfristige Abhängigkeiten zu lernen, 2001.

[13] Sepp Hochreiter und Jürgen Schmidhuber. Langes Kurzzeitgedächtnis. . . . . == Einzelnachweise ==

[14] Zhongqiang Huang und Mary Harper. Selbst-Training PCFG Grammatiken mit latenten Anmerkungen in allen Sprachen. In , Seiten 832–841. ACL, August 2009.

[15] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer und Yonghui Wu. die Grenzen der Sprachmodellierung. == Weblinks ==* Offizielle Website (englisch)== Ein

[16] Łukasz Kaiser und Samy Bengio. Kann aktives Gedächtnis die Aufmerksamkeit ersetzen? , 2016.

[17] Łukasz Kaiser und Ilya Suzkever. Neurale GPUs lernen Algorithmen. In , 2016.

[18] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves und Koray Kavukcuoglu. Neural maschinelle Übersetzung in linearer Zeit. . . . . Im Jahr 2017.

[19] Yoon Kim, Carl Denton, Luong Hoang und Alexander M. Rush. Strukturierte Aufmerksamkeitsnetzwerke. Im == Einzelnachweise ==

[20] Diederik Kingma und Jimmy Ba. Adam: Eine Methode zur stochastischen Optimierung.== Einzelnachweise ==

[21] Oleksii Kuchaiev und Boris Ginsburg. Factorisierungstricks für LSTM-Netzwerke. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[22] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou und Yoshua Bengio. Ein strukturierter selbstaufmerksamer Satz, der einbettet. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[23] Minh-Thang Luong, Quoc V. Le, Ilya Suzkever, Oriol Vinyals und Lukasz Kaiser. Sequenz zum Sequenz-Lernen. == Weblinks ==* Offizielle Website (englisch)== Einze

[24] Minh-Thang Luong, Hieu Pham und Christopher D Manning. Effektive Ansätze zur Aufmerksamkeit- neuronale maschinelle Übersetzung. == Weblinks ==* Offizielle Website (englisch)== Ei

[25] Mitchell P Marcus, Mary Ann Marcinkiewicz und Beatrice Santorini. Bau einer großen kommentiert corpus of english: Die penn treebank. == Einzelnachweise ==

[26] David McClosky, Eugene Charniak, und Mark Johnson. Effektive Selbst-Training für Parsing. In . . . . . Seite 152–159. ACL, Juni 2006.

[27] Ankur Parikh, Oscar Täckström, Dipanjan Das und Jakob Uszkoreit. Modell. In , 2016.

[28] Romain Paulus, Caiming Xiong und Richard Socher. Ein tief verstärktes Modell für abstrakte Zusammenfassung. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[29] Slav Petrov, Leon Barrett, Romain Thibaux und Dan Klein. und interpretierbare Baum-Annotation. In , Seiten 433–440. ACL, Juli 2006.

[30] Ofir Press und Lior Wolf. Verwendung der Ausgabe Einbettung, um Sprachmodelle zu verbessern. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[31] Rico Sennrich, Barry Haddow und Alexandra Birch. Neural maschinelle Übersetzung von seltenen Wörtern mit Unterwort-Einheiten. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, und Jeff Dean. Ausserordentlich große neuronale Netzwerke: Die spärlich-gated Mischung-von-Experten Schicht. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[33] Nitische Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Suzkever und Ruslan Salakhutdinov. Dropout: ein einfacher Weg, um zu verhindern, dass neuronale Netzwerke überlagert werden. == Einzelnachweise ==

[34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston und Rob Fergus. End-to-End-Speicher In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama und R. Garnett, Herausgeber, , Seiten 2440–2448. Curran Associates, == Einzelnachweise ==

[35] Ilya Suzkever, Oriol Vinyals und Quoc VV Le. Sequenz zum Sequenzlernen mit neuronalen . . . . . ,. Seiten 3104–3112, 2014. . . . . .

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens und Zbigniew Wojna. Überdenken der Inception-Architektur für Computer-Vision. , abs/1512.00567, 2015.

[37] Vinyals & Kaiser, Koo, Petrov, Suzkever und Hinton. Grammatik als Fremdsprache. In == Einzelnachweise ==

[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-Neuralmaschine Übersetzungssystem: Die Lücke zwischen Mensch und Maschinelle Übersetzung überbrücken. == Weblinks ==* Offizielle Website (englisch)== Einzelnachweise ==

[39] Jie Zhou, Ying Cao, Xugugang Wang, Peng Li und Wei Xu. Tief wiederkehrende Modelle mit Fast-Forward-Verbindungen für neuronale maschinelle Übersetzung. , abs/1606.04199, 2016.

[40] Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang und Jingbo Zhu. Schnell und genau Schicht-Reduce konstituierende Parsierung. , Seiten 434–443. ACL, August 2013.

Aufmerksamkeitsvisualisierungen

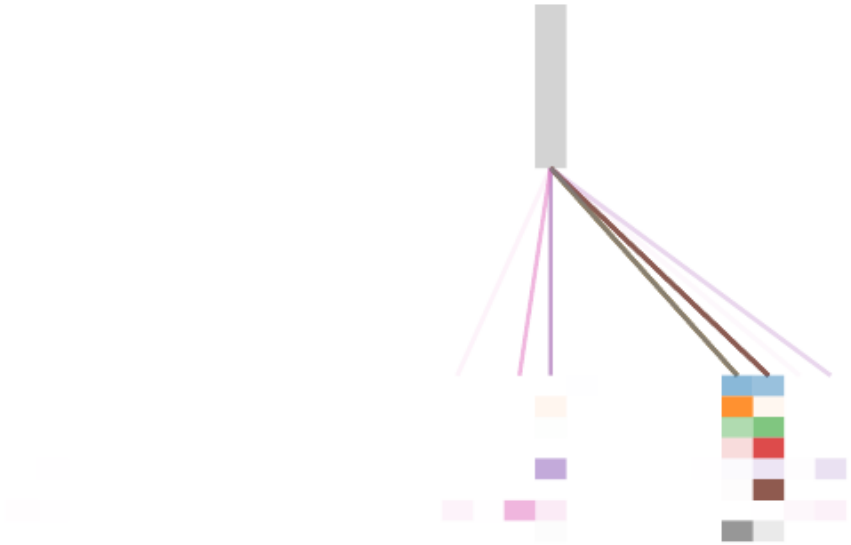


Abbildung 3: Ein Beispiel für den Aufmerksamkeitsmechanismus nach Fernabhängigkeiten in der encoder Selbstachtung in Schicht 5 von 6. Viele der Aufmerksamkeitsköpfe kümmern sich um eine entfernte Abhängigkeit vo das Verb ‘making-, die Ergänzung der Phrase ‘making...schwieriger . Aufmerksamkeiten hier nur für gezeigt das Wort „making". Verschiedene Farben repräsentieren verschiedene Köpfe. Am besten in Farbe betrachtet.

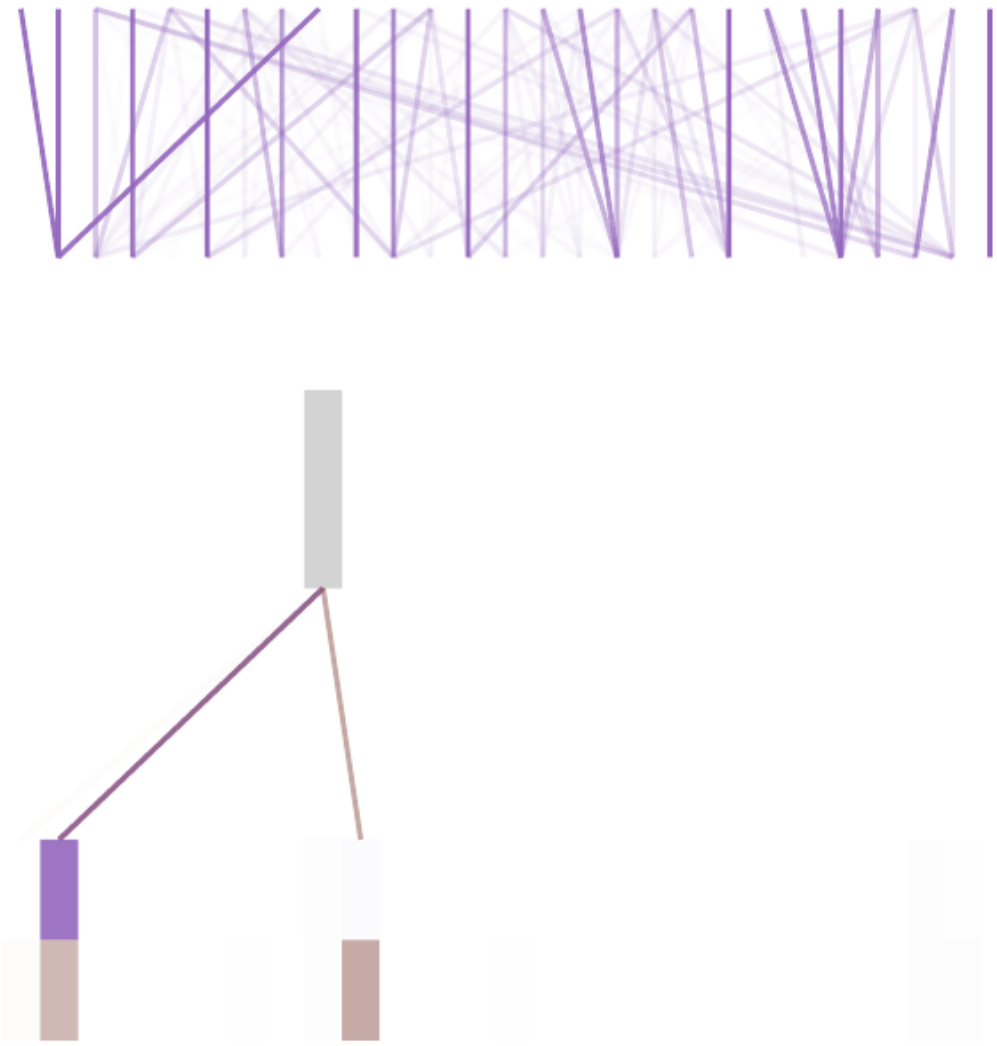


Abbildung 4: Zwei Aufmerksamkeitsköpfe, auch in Schicht 5 von 6, offenbar an der Auflösung der Anaphora beteiligt. Volle Aufmerksamkeit für Kopf 5. Unten: isolierte Aufmerksamkeiten von nur dem Wort ‘sits' für Aufmerksamkeit Köpfe 5 und 6. Beachten Sie, dass die Aufmerksamkeit für dieses Wort sehr scharf ist.

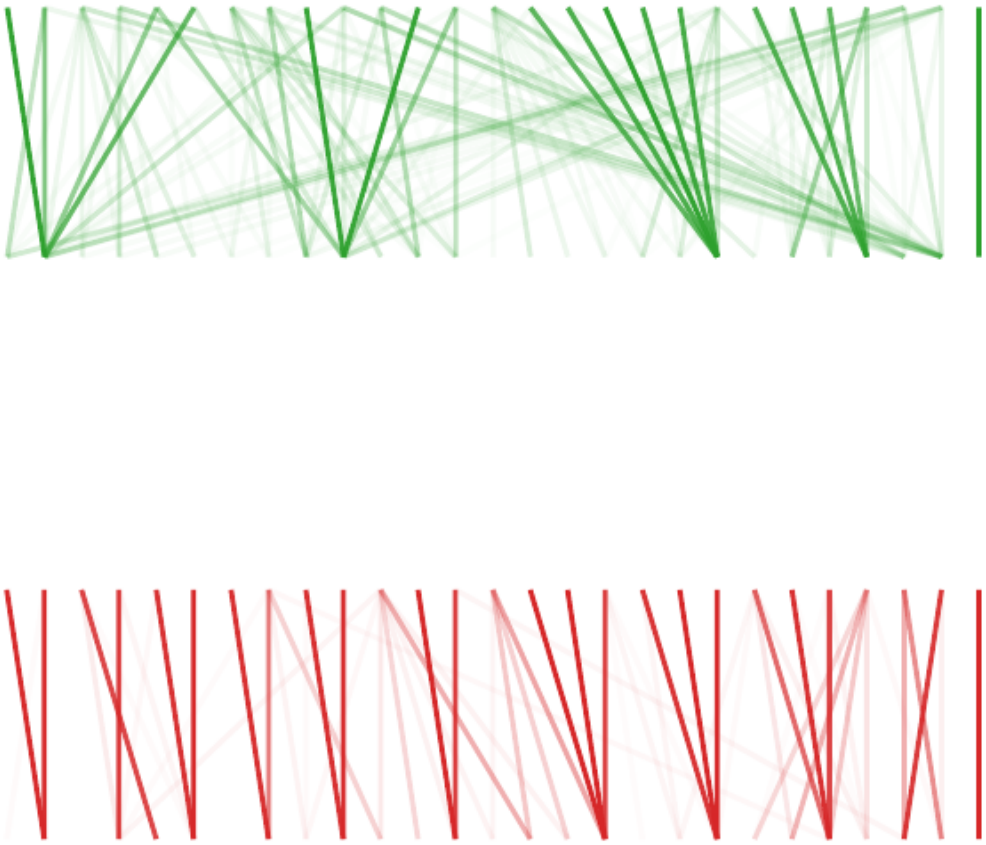


Abbildung 5: Viele der Aufmerksamkeitsköpfe weisen ein Verhalten auf, das mit der Struktur des Satz. Wir geben zwei solche Beispiele oben, aus zwei verschiedenen Köpfen von der Encoder Selbstaufmerksamkeit bei Schicht 5 von 6. Die Köpfe haben klar gelernt, verschiedene Aufgaben zu erfüllen.